ADOBE® DREAMWEAVER® CS3

DREAMWEAVER API リファレンス

Adobe

Adobe® Dreamweaver® API リファレンス Windows® / Macintosh® 版

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

目次

第 10 章 : ソースコントロール統合 API

第 11 章 : アプリケーション

第 12 章 : ワークスペース

第 13 章 : サイト

第 14 章 : ドキュメント

第 15 章 : ページコンテンツ

第 16 章 : 動的ドキュメント

第 17 章 : デザイン

第 18 章 : コード

第 19 章 : イネーブラ

第 1 章 : はじめに

『Adobe Dreamweaver CS3 API リファレンス』では、Adobe® Dreamweaver® CS3 の拡張機能を開発するときや Dreamweaver の Web ページにプログラムコードを追加するときに、さまざまなサポートタスクを実行するために使用する API (アプリケーションプログラミングインターフェイス) について説明します。これらの API には、メニューから実行可能なほぼすべての機能、およびそれ以外の機能も含めた、Dreamweaver の中心となる機能の大部分に対するアクセスを提供する、主要な JavaScript API が含まれます。また、ファイルの読み取りや書き込み、HTTP を使用した情報の転送、Fireworks や Flash との通信などの一般的なタスクを実行するための、さまざまなユーティリティ API も含まれています。

ユーティリティ API は、特定の種類のタスクを実行するために使用する関連機能のサブセットから構成されています。ユーティリティ API に含まれる API は以下のとおりです。

- ファイル I/O API。ローカルファイルシステム上で、ファイルの読み取りおよび書き込みを行います。
- HTTP API。Web サーバーの情報を送受信します。
- デザインノート API。Dreamweaver ドキュメントに関するノートの保存および取得を行います。
- Fireworks 統合 API。Adobe Fireworks と通信します。
- Flash との統合。Dreamweaver ユーザーインターフェイス (UI) への Flash エレメントの追加に関する情報と、Adobe Flash コンテンツを作成するオブジェクトを構築するために必要な Flash オブジェクト API の詳細が含まれています。
- データベース API。データベースに格納されている情報にアクセスし、データベース接続を管理します。
- データベース接続 API。新規または既存のサーバーモデルに対して、新しい接続タイプと、それに対応するダイアログボックスを作成します。
- JavaBeans API。定義した JavaBeans のクラス名、メソッド、プロパティ、イベントを取得します。
- ソースコントロール統合 API。共有ライブラリへの書き込みにより、Dreamweaver のチェックイン / チェックアウト機能を拡張します。

JavaScript API は、デベロッパーがさまざまな細かいタスクを実行するために使用する広範な API です。それらのタスクの多くは、ユーザーが Dreamweaver ドキュメントを作成または編集するときに実行します。これらの API 関数は、その関数が影響を与える Dreamweaver UI の部分ごとにグループ化されています。たとえば、JavaScript API にはワークスペース関数、ドキュメント関数、デザイン関数などがあります。これらの関数を使用すると、新しいドキュメントを開く、フォントサイズを設定および取得する、HTML コード内で検索文字列を検索する、ツールバーを表示するというような数多くのタスクを実行できます。

背景

ここでは、Dreamweaver、HTML、XML、JavaScript のプログラミングに関する知識をお持ちの方、場合によってはさらに C 言語のプログラミングに関する知識をお持ちの方を対象としています。Web アプリケーションを構築するための拡張機能を記述する場合は、Active Server Page (ASP)、ASP.net、PHP: Hypertext Preprocessor (PHP)、ColdFusion、Java Server Pages (JSP) など、少なくとも 1 つのプラットフォームでのサーバーサイドスクリプトの記述に精通している必要があります。

Dreamweaver 拡張ガイド

Dreamweaver のフレームワークと、Dreamweaver の拡張機能を構築する際に使用する API の詳細については、『Dreamweaver 拡張ガイド』を参照してください。『Dreamweaver 拡張ガイド』では、Dreamweaver のさまざまな機能を構成するオブジェクト、メニュー、フローティングパネル、サーバービヘイビアなどを実装するために Dreamweaver が呼び出す API 関数について説明しています。ここで説明されている API を使用して、オブジェクト、メニュー、フローティングパネル、その他の機能を製品に追加することができます。『Dreamweaver 拡張ガイド』では、さまざまな HTML ファイルや XML ファイルを編集したりタグを追加してメニュー項目やドキュメントタイプなどを追加することにより、Dreamweaver をカスタマイズする方法についても説明しています。

拡張機能を開発する際に使用できるその他のリソース

Dreamweaver のオンラインフォーラムに登録し、拡張機能を開発している他のデベロッパーと情報を交換することができます。このオンラインフォーラムには、www.adobe.com/support/dreamweaver/extend/form/ からアクセスできます。

Dreamweaver CS3 の新しい関数

Dreamweaver CS3 の JavaScript API には、次の新しい関数が追加されています。見出しは、新しい関数が含まれる章や項を示しています。

アプリケーション

第 11 章「アプリケーション」には、次の関数が追加されています。

外部アプリケーション関数

- 98 ページの dom.insertFiles()
- 98 ページの dreamweaver.activateApp()
- 99 ページの dreamweaver.printDocument()
- 99 ページの dreamweaver.revealDocument()

一般的なアプリケーション関数

- 103 ページの dw.registerIdleHandler()
- 103 ページの dw.revokeIdleHandler()

Bridge 通信

- 104 ページの BridgeTalk.bringToFront()
- 104 ページの Bridgetalk.send()
- 105 ページの BridgeTalk.suppressStartupScreen()
- 105 ページの dw.browseInBridge()

ワークスペース

106 ページのワークスペースには、次の新しいアクティブコンテンツ関数およびコードビュー関数が追加されています。

アクティブコンテンツ

- 115 ページの dom.convertNextActiveContent()
- 115 ページの dom.convertActiveContent()

コードビュー

- 174 ページの dom.source.refreshVariableCodeHints()

サイト

177 ページのサイトには、次の新しいサイト関数が追加されています。

- 188 ページの site.displaySyncInfoForFile()
- 435 ページの site.canDisplaySyncInfoForFile()

ドキュメント

210 ページのドキュメントには、次の新しい XML データセット関数が追加されています。

- 248 ページの MMXSLT.getXML()

ページコンテンツ

250 ページのページコンテンツには、次の新しい関数が追加されています。これらの関数では、Spry XML データセットの作成機能、Spry および他の Widget に対する強化された編集機能、Spry Widget の挿入機能、およびブラウザ互換性のチェック機能が提供されています。

Spry Widget の編集

- 280 ページの element.removeTranslatedAttribute()
- 281 ページの element.setTranslatedAttribute()
- 281 ページの element.translatedClassName
- 281 ページの element.translatedStyle

Spry Widget の挿入

- 282 ページの dom.addJavaScript()
- 282 ページの dom.copyAssets()
- 284 ページの dom.getDefaultAssetFolder()

ブラウザ互換性の問題のチェック

- 284 ページの elem.getComputedStyleProp()
- 285 ページの window.getDeclaredStyle()
- 286 ページの dom.getMinDisplayWidth()
- 286 ページの dom.getBlockElements() elem.getBlockElements()
- 287 ページの dom.getInlineElements() elem.getInlineElements()
- 288 ページの dom.getHeaderElements() elem.getHeaderElements()
- 288 ページの dom.getListElements() elem.getListElements()
- 289 ページの elem.isBlockElement()
- 289 ページの elem.isInlineElement()
- 290 ページの elem.isHeaderElement()
- 291 ページの elem.isListElement()

動的ドキュメント

292 ページの動的ドキュメントには、次の新しいデータソース関数が追加されています。

- 293 ページの dw.dbi.setExpanded()

デザイン

308 ページのデザインには、次の新しい CSS レイアウト関数が追加されています。

CSS レイアウト

- 308 ページの dom.applyLayout()
- 309 ページの dom.canApplyLayout()
- 310 ページの dw.getLayoutNames()
- 310 ページの dw.getLayoutDescriptions()
- 309 ページの dw.getFilesForLayout()

削除された関数

次の関数は、対応する機能が製品から削除されたため、Dreamweaver CS3 API から削除されました。

- 218 ページの dreamweaver.exportCSS() (非推奨)
- 418 ページの dreamweaver.canExportCSS() (非推奨)

お断り

既に確認されている問題点のリストは、Dreamweaver サポートセンターの「拡張機能」セクション (www.adobe.com/support/dreamweaver/extend/extending_dwmx_errata) で入手できます。

本マニュアルの表記上の規則

表記規則

本マニュアルでは、次のような表記規則を使用しています。

- コードの書式が設定されているフォントは、コードの一部および API リテラルを示します。このフォントで表記される項目には、クラス名、メソッド名、関数名、タイプ名、スクリプト、SQL ステートメント、HTML タグ、XML タグ、および属性名などがあります。
- イタリックコードの書式が設定されているフォントは、コード内の置き換え可能な項目を示します。
- 継続記号 (¬) は、1 行の長いコードが 2 行以上にまたがっていることを示します。本マニュアルの書式に基づくマージン制限により、本来は 1 行として入力する必要のあるコード行を分割して表記しています。コード行をコピーするときは、継続記号を削除して 1 行として入力してください。
- 関数の引数が波カッコ ({ }) で囲まれている場合は、その引数が省略可能であることを示しています。
- 接頭辞が付く `dreamweaver.`*`funcname`* という関数名は、コードでは省略して `dw.`*`funcname`* と記述することができます。本マニュアルでは、関数定義と索引では完全な `dreamweaver.` 接頭辞を使用しています。ただし、多くの例では `dw.` 接頭辞を使用しています。

用語

本マニュアルでは、以下の語句を使用します。

- デベロッパー - 拡張機能を作成するデベロッパー
- ユーザー - Dreamweaver の使用者

第 2 章 : ファイル I/O API

Adobe® Dreamweaver® CS3 には、DWfile と呼ばれる C 共有ライブラリが付属しています。このライブラリを使用すると、オブジェクト、コマンド、ビヘイビア、データトランスレータ、フローティングパネル、およびプロパティインスペクタの作成者は、ローカルファイルシステム上でファイルの読み取りおよび書き込みを行うことができます。本章では、ファイル I/O API とその使用方法について説明します。

Dreamweaver での C ライブラリと JavaScript インタープリタ間の対話に関する一般情報については、『Dreamweaver 拡張ガイド』の「C レベル拡張機能」を参照してください。

設定フォルダへのアクセス

Microsoft Windows 2000、Windows XP、および Mac OS X の各プラットフォームでは、ユーザーごとに専用の設定ファイルがあります。Dreamweaver によって設定ファイルへの書き込みが行われるときは、ユーザーの "Configuration" フォルダのファイルにも同じ内容が書き込まれます。同様に、Dreamweaver によって設定ファイルが読み取られる際は、まずユーザーの "Configuration" フォルダが検索され、次に Dreamweaver の "Configuration" フォルダが検索されます。DWfile 関数でも同じメカニズムが使用されます。つまり、拡張機能で Dreamweaver の "Configuration" フォルダにあるファイルの読み取りや書き込みを行う場合、ユーザーの "Configuration" フォルダもアクセスされます。マルチユーザープラットフォームにおける設定フォルダの詳細については、『Dreamweaver 拡張ガイド』を参照してください。

ファイル I/O API

ファイル I/O API の関数はすべて `DWfile` オブジェクトのメソッドです。

DWfile.copy()

対応バージョン

Dreamweaver 3

説明

この関数は、指定されたファイルを新しい場所にコピーします。

引数

originalURL、*copyURL*

- *originalURL* 引数には、コピーするファイルを file:// URL 形式で指定します。
- *copyURL* 引数には、コピーしたファイルを保存する場所を file:// URL 形式で指定します。

戻り値

コピーが成功した場合は `true` を、失敗した場合は `false` を示すブール値。

例

以下のコードは、"myconfig.cfg" というファイルを "myconfig_backup.cfg" にコピーします。

```
var fileURL = "file:///c|/Config/myconfig.cfg";
var newURL ="file:///c|/Config/myconfig_backup.cfg";
DWfile.copy(fileURL, newURL);
```

DWfile.createFolder()

対応バージョン
Dreamweaver 2

説明
この関数は、指定された場所にフォルダを作成します。

引数
folderURL

- *folderURL* 引数には、フォルダを作成する場所を file:// URL 形式で指定します。

戻り値
フォルダの作成に成功した場合は `true` を、失敗した場合は `false` を示すブール値。

例
以下のコードでは、"tempFolder" というフォルダを C ドライブの最上位レベルに作成し、操作が成功したかどうかを知らせるメッセージを表示します。

```
var folderURL = "file:///c|/tempFolder";
if (DWfile.createFolder(folderURL)){
    alert("Created " + folderURL);
}else{
    alert("Unable to create " + folderURL);
}
```

DWfile.exists()

対応バージョン
Dreamweaver 2

説明
この関数は、指定されたファイルが存在するかどうかをテストします。

引数
fileURL

- *fileURL* 引数には、目的のファイルを file:// URL 形式で指定します。

戻り値
ファイルが存在する場合は `true` を、存在しない場合は `false` を示すブール値。

例
以下のコードは、"mydata.txt" ファイルの存在を調べ、ファイルが存在するかどうかを知らせるメッセージを表示します。

```
var fileURL = "file:///c|/temp/mydata.txt";
if (DWfile.exists(fileURL)){
    alert(fileURL + " exists!");
}else{
    alert(fileURL + " does not exist.");
}
```

DWfile.getAttributes()

対応バージョン

Dreamweaver 2

説明

この関数は、指定されたファイルまたはフォルダの属性を取得します。

引数

fileURL

- *fileURL* 引数には、属性を取得するファイルまたはフォルダを file:// URL 形式で指定します。

戻り値

指定されたファイルまたはフォルダの属性を表すストリング。ファイルまたはフォルダが存在しない場合は、null 値が返されます。ストリング内の文字が表す属性は、次のとおりです。

- R - 読み取り専用
- D - フォルダ
- H - 隠しファイルまたは隠しフォルダ
- S - システムファイルまたはシステムフォルダ

例

以下のコードでは、"mydata.txt" ファイルの属性を取得し、ファイルが読み取り専用である場合はメッセージを表示します。

```
var fileURL = "file:///c|/temp/mydata.txt";
var str = DWfile.getAttributes(fileURL);
if (str && (str.indexOf("R") != -1)){
    alert(fileURL + " is read only!");
}
```

DWfile.getModificationDate()

対応バージョン

Dreamweaver 2

説明

この関数は、ファイルが最後に修正された日時を取得します。

引数

fileURL

- *fileURL* 引数には、最後に修正された日時をチェックするファイルを file:// URL 形式で指定します。

戻り値

基準時間から経過した時間単位数を表す 16 進数を含むストリング。基準時間と時間単位の正確な意味は、プラットフォームによって異なります。たとえば Windows では、単位は 100ns で、基準時間は 1600 年 1 月 1 日です。

例

この関数によって返される値は、プラットフォームによって異なり、また識別可能な日時ではないため、この関数を 2 回呼び出して、それぞれの戻り値を比較する方法が便利です。以下のコードでは、"file1.txt" と "file2.txt" の変更日を取得し、どちらのファイルが新しいかを知らせるメッセージを表示します。

```
var file1 = "file:///c|/temp/file1.txt";
var file2 = "file:///c|/temp/file2.txt";
var time1 = DWfile.getModificationDate(file1);
var time2 = DWfile.getModificationDate(file2);
if (time1 == time2){
    alert("file1 and file2 were saved at the same time");
}else if (time1 < time2){
    alert("file1 older that file2");
}else{
    alert("file1 is newer than file2");
}
```

DWfile.getCreationDate()

対応バージョン

Dreamweaver 4

説明

この関数は、ファイルが作成された日時を取得します。

引数

fileURL

- *fileURL* 引数には、作成された日時をチェックするファイルを file:// URL 形式で指定します。

戻り値

基準時間から経過した時間単位数を表す 16 進数を含むストリング。基準時間と時間単位の正確な意味は、プラットフォームによって異なります。たとえば Windows では、単位は 100ns で、基準時間は 1600 年 1 月 1 日です。

例

ファイルに対して、この関数および `DWfile.getModificationDate()` 関数を呼び出すことで、変更日と作成日を比較できます。

```
var file1 = "file:///c|/temp/file1.txt";
var time1 = DWfile.getCreationDate(file1);
var time2 = DWfile.getModificationDate(file1);
if (time1 == time2){
    alert("file1 has not been modified since it was created");
}else if (time1 < time2){
    alert("file1 was last modified on " + time2);
}
```

DWfile.getCreationDateObj()

対応バージョン

Dreamweaver MX

説明

この関数は、ファイルが作成された日時を表す JavaScript オブジェクトを取得します。

引数

fileURL

- *fileURL* 引数には、作成された日時をチェックするファイルを file:// URL 形式で指定します。

戻り値

指定したファイルが作成された日時を表す JavaScript `Date` オブジェクト。

DWfile.getModificationDateObj()

対応バージョン
Dreamweaver MX

説明
この関数は、ファイルが最後に修正された日時を表す JavaScript `Date` オブジェクトを取得します。

引数
`fileURL`

- *`fileURL`* 引数には、最後に修正された日時をチェックするファイルを file:// URL 形式で指定します。

戻り値
指定したファイルが最後に修正された日時を表す JavaScript `Date` オブジェクト。

DWfile.getSize()

対応バージョン
Dreamweaver MX

説明
この関数は、指定されたファイルのサイズを取得します。

引数
`fileURL`

- *`fileURL`* 引数には、サイズをチェックするファイルを file:// URL 形式で指定します。

戻り値
指定されたファイルの実際のサイズをバイト数で表す整数。

DWfile.listFolder()

対応バージョン
Dreamweaver 2

説明
この関数は、指定されたフォルダの内容のリストを取得します。

引数
`folderURL`、*`{constraint}`*

- *`folderURL`* 引数には、内容のリストを取得するフォルダを file:// URL 形式で指定し、オプションとしてワイルドカードのファイルマスクを追加指定します。有効なワイルドカードは、1 文字以上に相当するアスタリスク (*) と、1 文字に相当する疑問符 (?) です。
- *`constraint`* 引数 (オプション) は、`"files"` (ファイルのみを返す) または `"directories"` (フォルダのみを返す) のいずれかを指定します。この引数を省略した場合は、ファイルとフォルダが返されます。

戻り値
フォルダの内容を表すストリングの配列。

例

以下のコードは、"C:¥temp" フォルダ内のすべてのテキスト (TXT) ファイルのリストを取得し、それをメッセージとして表示します。

```
var folderURL = "file:///c|/temp";
var fileMask = "*.txt";
var list = DWfile.listFolder(folderURL + "/" + fileMask, "files");
if (list){
    alert(folderURL + " contains: " + list.join("\n"));
}
```

DWfile.read()

対応バージョン

Dreamweaver 2

説明

この関数は、指定されたファイルの内容をストリングとして読み取ります。

引数

fileURL

- *fileURL* 引数には、読み取るファイルを file:// URL 形式で指定します。

戻り値

ファイルの内容を含むストリング。読み取りが失敗した場合は、`null` 値が返されます。

例

以下のコードは "mydata.txt" ファイルを読み取り、成功した場合はファイルの内容をメッセージに表示します。

```
var fileURL = "file:///c|/temp/mydata.txt";
var str = DWfile.read(fileURL);
if (str){
    alert(fileURL + " contains: " + str);
}
```

DWfile.remove()

対応バージョン

Dreamweaver 3

説明

この関数は、指定されたファイルを削除します。

引数

fileURL

- *fileURL* 引数には、削除するファイルを file:// URL 形式で指定します。

戻り値

操作が成功した場合は `true` を、失敗した場合は `false` を示すブール値。

例

以下の例では、DWfile.getAttributes() 関数を使用してファイルが読み取り専用かどうかを判別し、confirm() 関数を使用して判別結果を示すダイアログボックスを表示します。

```
function deleteFile(){
    var delAnyway = false;
    var selIndex = document.theForm.menu.selectedIndex;

    var selFile = document.theForm.menu.options[selIndex].value;
    if (DWfile.getAttributes(selFile).indexOf('R') != -1){
        delAnyway = confirm('This file is read-only.Delete anyway?');
        if (delAnyway){
            DWfile.remove(selFile);
        }
    }
}
```

DWfile.setAttributes()

対応バージョン

Dreamweaver MX

説明

この関数は、特定のファイルにシステムレベルの属性を設定します。

引数

fileURL、*strAttrs*

- *fileURL* 引数には、属性を設定するファイルを file:// URL 形式で指定します。
- *strAttrs* 引数には、*fileURL* 引数で指定したファイルに設定するシステムレベルの属性を指定します。次の表は、有効な属性値とその意味を示しています。

属性値	説明
R	読み取り専用
W	書き込み可能 (R より優先されます)
H	非表示
V	表示 (H より優先されます)

strAttrs ストリングで使用できる値は、R、W、H、V、RH、RV、WH、または WV です。

R と W は互いに排他的なので、組み合わせて使用しないでください。この 2 つの値を組み合わせると、R は無効になり、ファイルは書き込み可能 (W) に設定されます。また、H と V も互いに排他的なので、組み合わせて使用しないでください。この 2 つの値を組み合わせると、H は無効になり、ファイルは表示 (V) に設定されます。

読み取り / 書き込み属性の R または W を指定せずに H または V を指定すると、ファイルの既存の読み取り / 書き込み属性は変更されません。同様に、表示属性の H または V を指定せずに R または W を指定すると、ファイルの既存の表示属性は変更されません。

戻り値

なし

DWfile.write()

対応バージョン

Dreamweaver 2

説明

この関数は、指定されたストリングを指定されたファイルに書き込みます。指定されたファイルが存在しない場合は、そのファイルを作成します。

引数

fileURL、*text*、*{mode}*

- *fileURL* 引数には、書き込み先のファイルを file:// URL 形式で指定します。
- *text* 引数には、書き込むストリングを指定します。
- *mode* 引数 (オプション) は、指定する場合は常に `"append"` (追加) である必要があります。この引数を省略すると、ファイルの内容が指定されたストリングによって上書きされます。

戻り値

ファイルへのストリングの書き込みが成功した場合は `true` を、失敗した場合は `false` を示すブール値。

例

以下のコードでは、"mydata.txt" ファイルに `"xxx"` というストリングを書き込み、書き込み処理が成功した場合はメッセージを表示します。次に、`"aaa"` というストリングをこのファイルに追加し、その書き込みが成功した場合は 2 番目のメッセージを表示します。このスクリプトを実行した後は、"mydata.txt" ファイルに `xxxaaa` というテキストのみが含まれることになります。

```
var fileURL = "file:///c|/temp/mydata.txt";
if (DWfile.write(fileURL, "xxx")){
    alert("Wrote xxx to " + fileURL);
}
if (DWfile.write(fileURL, "aaa", "append")){
    alert("Appended aaa to " + fileURL);
}
```

第 3 章 : HTTP API

拡張機能の対象は、ローカルファイルシステム内での操作に限りません。Adobe® Dreamweaver® CS3 では、HTTP (Hypertext Transfer Protocol) を使用して、Web サーバーから情報を取得したり、Web サーバーに情報を送信するメカニズムが提供されています。本章では、HTTP API とその使用方法について説明します。

HTTP API の動作

HTTP API の関数は、すべて `MMHttp` オブジェクトのメソッドです。これらの関数の大部分では、URL が引数として使用され、1 つのオブジェクトが返されます。URL 引数のデフォルトのポートは 80 です。80 以外のポートを指定するには、次の例に示すように、URL の後にコロンとポート番号を追加します。

```
MMHttp.getText("http://www.myserver.com:8025");
```

関数がオブジェクトを返す場合、そのオブジェクトには `statusCode` と `data` の 2 つのプロパティがあります。

`statusCode` プロパティでは、操作のステータスが示されます。以下に、このプロパティの値の一部を示します。

- 200: 正常ステータス。
- 400: 要求を判別できません。
- 404: 要求された URL が見つかりません。
- 405: サーバーが要求されたメソッドをサポートしません。
- 500: 不明なサーバーエラーです。
- 503: サーバーの処理能力の限界に達しました。

使用するサーバーのすべてのステータスコードについては、ご利用のインターネットサービスプロバイダまたはシステム管理者までお問い合わせください。

`data` プロパティの値は関数によって異なります。このプロパティに使用できる値は、各関数のセクションで指定されています。

オブジェクトを返す関数には、コールバックバージョンもあります。コールバック関数を使用すると、Web サーバーが HTTP 要求を処理する間に、他の関数を実行できます。この機能は、Dreamweaver から複数の HTTP 要求を送信する場合に便利です。コールバックバージョンの関数により、最初の引数として指定された関数に、その ID と戻り値が直接渡されます。

HTTP API

この項では、`MMHttp` オブジェクトのメソッドである関数について説明します。

MMHttp.clearServerScriptsFolder()

対応バージョン

Dreamweaver MX

説明

現在のサイト (ローカルまたはリモート) で、ルートフォルダの下にある "_mmServerScripts" フォルダとそのファイルをすべて削除します。"_mmServerScripts" フォルダは、"Configuration/Connections/Scripts/*server-model*/_mmDBScripts" フォルダにあります。

引数

serverScriptsfolder

- *serverScriptsfolder* 引数は、サーバースクリプトを取得およびクリアするときの対象になるフォルダの名前を、アプリケーションサーバー上の "Configuration" フォルダを基準に指定するストリングです。

戻り値

サーバーからの応答を表すオブジェクト。このオブジェクトの `data` プロパティは、削除したスクリプトの内容を含むストリングです。エラーが発生した場合は、返されるオブジェクトの `statusCode` プロパティにエラーが設定されます。

例

次のコードを "Configuration/Menus" フォルダ内のメニューコマンドファイルに挿入すると、メニューから呼び出したときに、"_mmServerScripts" フォルダからすべてのファイルが削除されます。

```
<!-- MENU-LOCATION=NONE -->
<html>
<head>
<TITLE>Clear Server Scripts</TITLE>
<SCRIPT SRC="ClearServerScripts.js"></SCRIPT>
<SCRIPT LANGUAGE="javascript">
</SCRIPT>
<body onLoad="MMHttp.clearServerScriptsFolder()">
</body>
</html>
```

MMHttp.clearTemp()

説明

この関数は、Dreamweaver アプリケーションフォルダ内の "Configuration/Temp" フォルダから、すべてのファイルを削除します。

引数

なし

戻り値

なし

例

以下のコードを "Configuration/Shutdown" フォルダ内のファイルに保存すると、Dreamweaver を終了する際に、"Configuration/Temp" フォルダからすべてのファイルが削除されます。

```
<html>
<head>
<title>Clean Up Temp Files on Shutdown</title>
</head>
<body onLoad="MMHttp.clearTemp()">
</body>
</html>
```

MMHttp.getFile()

説明

この関数は、指定された URL にあるファイルを取得し、Dreamweaver アプリケーションフォルダ内の "Configuration/Temp" フォルダに保存します。Dreamweaver では、サーバーのフォルダ構造に合わせたサブフォルダが自動的に作成されます。たとえば、指定されたファイルが "www.dreamcentral.com/people/index.html" である場合、"www.dreamcentral.com" フォルダ内の "People" フォルダに "index.html" ファイルが保存されます。

引数

URL、*{prompt}*、*{saveURL}*、*{titleBarLabel}*

- *URL* 引数には、Web サーバー上の絶対 URL を指定します。URL の "http://" を省略すると、HTTP プロトコルが指定されたと見なされます。
- *prompt* 引数 (オプション) は、ファイルの保存をユーザーに指示するかどうかを指定するブール値です。*saveURL* が "Configuration/Temp" フォルダ以外にある場合は、*prompt* に `false` を指定しても、セキュリティ上の理由から無視されます。
- *saveURL* 引数 (オプション) には、ファイルの保存先となるユーザーのハードディスク上の場所を file:// URL 形式で指定します。prompt を `true` 値に設定した場合、または *saveURL* が "Configuration/Temp" フォルダ以外にある場合、ユーザーは [保存] ダイアログボックスで *saveURL* を上書きすることができます。
- *titleBarLabel* 引数 (オプション) には、[保存] ダイアログボックスのタイトルバーに表示されるラベルを指定します。

戻り値

サーバーからの応答を表すオブジェクト。このオブジェクトの `data` プロパティは、ファイルが保存される場所を file:// URL 形式で表したストリングです。通常、このオブジェクトの `statusCode` プロパティには、サーバーから受信したステータスコードが格納されます。ただし、ローカルドライブ上にファイルを保存する動作中にディスクエラーが発生して操作が失敗すると、`statusCode` プロパティには以下のいずれかのエラーコードを表す整数が格納されます。

- 1: 指定されていないエラーです。
- 2: ファイルが見つかりません。
- 3: 無効なパスです。
- 4: 開くことのできるファイル数の上限に達しました。
- 5: アクセスが拒否されました。
- 6: 無効なファイルハンドルです。
- 7: 現在の作業フォルダを削除できません。
- 8: これ以上のフォルダエントリはありません。
- 9: ファイルポインタの設定エラーです。
- 10: ハードウェアエラーです。
- 11: 共有違反です。
- 12: ロック違反です。
- 13: ディスクがいっぱいです。
- 14: ファイルの終わりに達しました。

例

以下のコードは、HTML ファイルを取得し、"Configuration/Temp" フォルダ内のすべてのファイルを保存してから、取得した HTML ファイルのローカルコピーをブラウザで開きます。

```
var httpReply = MMHttp.getFile("http://www.dreamcentral.com/people/profiles/scott.html",
false);
if (Boolean == 200){
    var saveLoc = httpReply.data;
    dw.browseDocument(saveLoc);
}
```

MMHttp.getFileCallback()

説明

この関数は、指定された URL にあるファイルを取得し、Dreamweaver アプリケーションフォルダ内の "Configuration/Temp" フォルダに保存します。次に、要求 ID と応答結果を指定して、指定された関数を呼び出します。Dreamweaver では、ファイルがローカルに保存される際に、サーバーのフォルダ構造に合わせて、サブフォルダが自動的に作成されます。たとえば、指定されたファイルが "www.dreamcentral.com/people/index.html" である場合は、"www.dreamcentral.com" フォルダ内の "People" フォルダに、"index.html" ファイルが保存されます。

引数

callbackFunction、*URL*、*{prompt}*、*{saveURL}*、*{titleBarLabel}*

- *callbackFunction* 引数は、HTTP 要求が完了したときに呼び出される JavaScript 関数の名前です。
- *URL* 引数には、Web サーバー上の絶対 URL を指定します。URL の "http://" を省略すると、HTTP プロトコルが指定されたと見なされます。
- *prompt* 引数 (オプション) は、ファイルの保存をユーザーに指示するかどうかを指定するブール値です。*saveURL* 引数に "Configuration/Temp" フォルダ以外を指定した場合は、*prompt* に `false` を指定しても、セキュリティ上の理由から無視されます。
- *saveURL* 引数 (オプション) には、ファイルの保存先となるユーザーのハードディスク上の場所を file:// URL 形式で指定します。prompt を `true` 値に設定した場合、または *saveURL* が "Configuration/Temp" フォルダ以外にある場合、ユーザーは [保存] ダイアログボックスで *saveURL* を上書きすることができます。
- *titleBarLabel* 引数 (オプション) には、[保存] ダイアログボックスのタイトルバーに表示されるラベルを指定します。

戻り値

サーバーからの応答を表すオブジェクト。このオブジェクトの `data` プロパティは、ファイルが保存された場所を file:// URL 形式で表したストリングです。通常、このオブジェクトの `statusCode` プロパティには、サーバーから受信したステータスコードが格納されます。ただし、ローカルドライブにファイルを保存する動作中にディスクエラーが発生した場合は、`statusCode` プロパティに以下のいずれかのエラーコードを表す整数が格納されます。これらのエラーコードのリストについては 15 ページの MMHttp.getFile() を参照してください。

MMHttp.getText()

対応バージョン

Dreamweaver UltraDeveloper 4、Dreamweaver MX で機能強化。

説明

指定された URL にあるドキュメントの内容を取得します。

引数

URL、*{serverScriptsFolder}*

- *URL* 引数には、Web サーバー上の絶対 URL を指定します。URL の "http://" を省略した場合、Dreamweaver では、それが HTTP プロトコルと見なされます。
- *serverScriptsFolder* 引数は、サーバースクリプトを取得する元のフォルダの名前を、アプリケーションサーバー上の "Configuration" フォルダを基準に指定するオプションのストリングです。スクリプトの取得には、FTP、WebDAV、リモートファイルシステムなどの適切な転送プロトコルが使用されます。これらのファイルは、現在のサイトのルートフォルダにある "_mmServerScripts" サブフォルダにコピーされます。

エラーが発生した場合は、返されるオブジェクトの `statusCode` プロパティにエラーが設定されます。

MMHttp.getTextCallback()

対応バージョン

Dreamweaver UltraDeveloper 4、Dreamweaver MX で機能強化。

説明

指定された URL にあるドキュメントの内容を取得し、指定された関数に渡します。

引数

callbackFunc、*URL*、*{serverScriptsFolder}*

- *callbackFunc* 引数は、HTTP 要求が完了したときに呼び出される JavaScript 関数です。
- *URL* 引数には、Web サーバー上の絶対 URL を指定します。URL の "http://" を省略すると、HTTP プロトコルが指定されたと見なされます。
- *serverScriptsFolder* 引数は、サーバースクリプトを取得する元のフォルダの名前を、アプリケーションサーバー上の "Configuration" フォルダを基準に指定するオプションのストリングです。スクリプトの取得には、FTP、WebDAV、リモートファイルシステムなどの適切な転送プロトコルが使用されます。これらのファイルが取得され、*callbackFunc* に指定した関数に渡されます。

エラーが発生した場合は、返されるオブジェクトの `statusCode` プロパティにエラーが設定されます。

MMHttp.postText()

対応バージョン

Dreamweaver UltraDeveloper 4、Dreamweaver MX で機能強化。

説明

指定された URL に対して、指定されたデータの HTTP post を実行します。通常、post 操作に関連付けられているデータは、フォームエンコードされたテキストですが、サーバーが受信できるすべてのタイプのデータを使用できます。

引数

URL、*dataToPost*、*{contentType}*、*{serverScriptsFolder}*

- *URL* 引数には、Web サーバー上の絶対 URL を指定します。URL の "http://" を省略すると、HTTP プロトコルが指定されたと見なされます。
- *dataToPost* 引数には、post するデータを指定します。3 番目の引数が `"application/x-www-form-urlencoded"` であるか、または省略された場合は、RFC 1866 のセクション 8.2.1 の規定に従って、*dataToPost* をフォームエンコードする必要があります。RFC 1866 は、www.faqs.org/rfcs/rfc1866.html から入手できます。
- *contentType* 引数 (オプション) には、post するデータのコンテンツタイプを指定します。この引数を省略した場合は、デフォルトで `"application/x-www-form-urlencoded"` に設定されます。
- *serverScriptsFolder* 引数は、データの post 先とするフォルダの名前を、アプリケーションサーバー上の "Configuration" フォルダを基準に指定するオプションのストリングです。データの post には、FTP、WebDAV、リモートファイルシステムなどの適切な転送プロトコルが使用されます。

エラーが発生した場合は、返されるオブジェクトの `statusCode` プロパティにエラーが設定されます。

例

次の `MMHttp.postText()` 関数呼び出しの例では、ローカルコンピュータの "Configuration" フォルダにある "DeployScripts" というフォルダに、"myScripts.cfm" ファイルが格納されていることを前提としています。

```
MMHttp.postText(
    "http://ultraqa8/DeployScripts/myScripts.cfm",
    "arg1=Foo",
    "application/x-www-form-urlencoded",
    "Configuration/DeployScripts/"
)
```

Dreamweaver でこの関数呼び出しが実行されると、次のことが起きます。

1 ローカルコンピュータの "Configuration/DeployScripts" フォルダにある "myScripts.cfm" ファイルが、"DeployScripts" という別のフォルダにコピーされます。"DeployScripts" フォルダは、ultraqa8 サイト上のルートフォルダのサブフォルダです。ファイルの配置には、サイト構成プロパティで指定したプロトコルが使用されます。

2 Dreamweaver が HTTP プロトコルを使って、`arg1=Foo` データを Web サーバーに post します。

3 post 要求を受けた ultraqa8 上の Web サーバーが、`arg1` のデータを使って myScripts.cfm スクリプトを実行します。

MMHttp.postTextCallback()

対応バージョン

Dreamweaver UltraDeveloper 4、Dreamweaver MX で機能強化。

説明

指定された URL に対してテキストの HTTP post を実行し、サーバーからの応答を指定された関数に渡します。通常、post 操作に関連付けられているデータは、フォームエンコードされたテキストですが、サーバーが受信できるすべてのタイプのデータを使用できます。

引数

`callbackFunc`、*`URL`*、*`dataToPost`*、*`{contentType}`*、*`{serverScriptsFolder}`*

- *`callbackFunc`* 引数には、HTTP 要求が完了したときに呼び出される JavaScript 関数の名前を指定します。
- *`URL`* 引数には、Web サーバー上の絶対 URL を指定します。URL の "http://" を省略すると、HTTP プロトコルが指定されたと見なされます。
- *`dataToPost`* 引数には、post するデータを指定します。3 番目の引数が `"application/x-www-form-urlencoded"` であるか、または省略された場合は、RFC 1866 のセクション 8.2.1 の規定に従って、*`data`* をフォームエンコードする必要があります。RFC 1866 は、www.faqs.org/rfcs/rfc1866.html から入手できます。
- *`contentType`* 引数 (オプション) には、post するデータのコンテンツタイプを指定します。この引数を省略した場合は、デフォルトで `"application/x-www-form-urlencoded"` に設定されます。
- *`serverScriptsFolder`* 引数は、オプションのストリングです。データの post 先とするフォルダの名前を、アプリケーションサーバー上の "Configuration" フォルダを基準に指定します。データの post には、FTP、WebDAV、リモートファイルシステムなどの適切な転送プロトコルが使用されます。これらのデータが取得され、*`callbackFunc`* に指定した関数に渡されます。

エラーが発生した場合は、返されるオブジェクトの `statusCode` プロパティにエラーが設定されます。

第 4 章 : デザインノート API

Web デザイナーやデベロッパーは、Adobe® Dreamweaver® CS3、Fireworks、および Flash を使用して、ドキュメントに関する追加情報 (レビューコメント、変更メモ、GIF や JPEG のソースファイルなど) をデザインノートというファイルに保存したり、その情報を取り出すことができます。

MMNotes は、拡張機能ファイルの作成者によるデザインノートファイルの読み取りや書き込みを可能にする C 共有ライブラリです。DWfile 共有ライブラリと同様に、MMNotes には、ライブラリ内の関数をオブジェクト、コマンド、ビヘイビア、フローティングパネル、プロパティインスペクタ、およびデータトランスレータから呼び出すための JavaScript API があります。

また、MMNotes には、他のアプリケーションによるデザインノートファイルの読み取りや書き込みを可能にする C API も含まれています。MMNotes 共有ライブラリは、Dreamweaver がインストールされていなくても単独で使用することができます。

Dreamweaver におけるデザインノート機能の詳細については、『Dreamweaver ユーザーガイド』を参照してください。

デザインノートの動作

各デザインノートファイルには、単一のドキュメントに関する情報が保存されます。フォルダ内のドキュメントにデザインノートファイルが関連付けられている場合は、Dreamweaver によって "_notes" サブフォルダが作成され、そこにデザインノートファイルが保存されます。"_notes" フォルダとその中のデザインノートファイルは、[サイト] パネルには表示されませんが、Finder (Macintosh) や Windows エクスプローラには表示されます。デザインノートのファイル名は、基本となるファイル名と拡張子 .mno で構成されます。たとえば "avocado8.gif" に関連付けられているデザインノートファイルは、"avocado8.gif.mno" という名前になります。

デザインノートファイルは、一連のキー / 値ペアとして情報を保存する XML ファイルです。キーは保存される情報の種類を表し、値は情報を表します。キーの長さは 64 文字に制限されています。

次の例は、"foghorn.gif.mno" のデザインノートファイルを示しています。

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<info>
    <infoitem key="FW_source" value="file:///C|sites/dreamcentral/images/sourceFiles/¬
        foghorn.png" />
    <infoitem key="Author" value="Heidi B." />
    <infoitem key="Status" value="Final draft, approved by Jay L." />
</info>
```

デザインノート JavaScript API

デザインノート JavaScript API のすべての関数は、`MMNotes` オブジェクトのメソッドです。

MMNotes.close()

説明

指定されたデザインノートファイルを閉じ、加えられた変更内容を保存します。キーと値のペアがすべて削除されると、デザインノートファイルは削除されます。削除されたファイルが "_notes" フォルダの最後のデザインノートファイルだった場合、"_notes" フォルダも削除されます。

注意 : デザインノートを終了するときは、常に *`MMNotes.close()`* 関数を呼び出します。これにより、*Dreamweaver* によるファイルへの書き込みが実行されます。

引数

fileHandle

- *fileHandle* 引数には、MMNotes.open() 関数から返されるファイルハンドルを指定します。

戻り値

なし

例

詳細については、23 ページの MMNotes.set() を参照してください。

MMNotes.filePathToLocalURL()

説明

指定されたローカルドライブパスを file:// URL 形式に変換します。

引数

drivePath

- *drivePath* 引数には、完全なドライブパスを含むストリングを指定します。

戻り値

指定されたファイルの file:// URL を含むストリング。

例

MMNotes.filePathToLocalURL('C:¥sites¥webdev¥index.htm') を呼び出すと、
"file:///c|/sites/webdev/index.htm" が返されます。

MMNotes.get()

説明

指定されたデザインノートファイルに含まれている指定されたキーの値を取得します。

引数

fileHandle、*keyName*

- *fileHandle* 引数には、MMNotes.open() から返されるファイルハンドルを指定します。
- *keyName* 引数には、キーの名前を含むストリングを指定します。

戻り値

キーの値を含むストリング。

例

詳細については、21 ページの MMNotes.getKeys() を参照してください。

MMNotes.getKeyCount()

説明

指定されたデザインノートファイルに含まれているキー / 値ペアの数を取得します。

引数

fileHandle

- *fileHandle* 引数には、MMNotes.open() 関数から返されるファイルハンドルを指定します。

戻り値

デザインノートファイルに含まれるキー / 値ペアの数を表す整数。

MMNotes.getKeys()

説明

デザインノートファイルに含まれるすべてのキーのリストを取得します。

引数

fileHandle

- *fileHandle* 引数には、MMNotes.open() 関数から返されるファイルハンドルを指定します。

戻り値

キーの名前を含むストリングの配列。

例

次のコードは、カスタムフローティングパネルでアクティブなドキュメントのデザインノート情報を表示するために使用できます。

```
var noteHandle = MMNotes.open(dw.getDocumentDOM().URL);
var theKeys = MMNotes.getKeys(noteHandle);
var noteString = "";
var theValue = "";
for (var i=0; i < theKeys.length; i++){
    theValue = MMNotes.get(noteHandle,theKeys[i]);
    noteString +=0theKeys[i] + " = " theValue + "\n";
}
document.theForm.bigTextField.value = noteString;
// 必ず noteHandle を閉じます。
MMNotes.close(noteHandle);
```

MMNotes.getSiteRootForFile()

説明

指定されたデザインノートファイルのサイトルートを確認します。

引数

fileURL

- *fileURL* 引数には、ローカルファイルへのパスを file:// URL 形式で指定します。

戻り値

サイトのローカルルートフォルダへのパス (file:// URL 形式) を含むストリング。または、Dreamweaver がインストールされていない場合やデザインノートファイルが Dreamweaver で定義されたサイトの外部にある場合は、空のストリング。この関数は、Dreamweaver に定義されているすべてのサイトを検索します。

MMNotes.getVersionName()

説明

MMNotes 共有ライブラリのバージョン名を取得します。このバージョン名は、共有ライブラリを実装したアプリケーションを示します。

引数

なし

戻り値

MMNotes 共有ライブラリを実装したアプリケーション名を含むストリング。

例

Dreamweaver のコマンド、オブジェクト、ビヘイビア、プロパティインスペクタ、フローティングパネル、またはデータトランスレータから `MMNotes.getVersionName()` 関数を呼び出すと、`"Dreamweaver"` が返されます。同様に、Fireworks から `MMNotes.getVersionName()` 関数を呼び出した場合も、`"Dreamweaver"` が返されます。これは、Fireworks が Dreamweaver のエンジニアリングチームによって作成された同じバージョンのライブラリを使用しているためです。

MMNotes.getVersionNum()

説明

MMNotes 共有ライブラリのバージョン番号を取得します。

引数

なし

戻り値

バージョン番号を含むストリング。

MMNotes.localURLToFilePath()

説明

指定された file:// URL をローカルドライブパスに変換します。

引数

`fileURL`

- *`fileURL`* 引数には、ローカルファイルへのパスを file:// URL 形式で指定します。

戻り値

指定されたファイルへのローカルドライブパスを含むストリング。

例

`MMNotes.localURLToFilePath('file:///MacintoshHD/images/moon.gif')` を呼び出すと、`"MacintoshHD:images:moon.gif"` が返されます。

MMNotes.open()

説明

指定されたファイルに関連付けられているデザインノートファイルを開きます。デザインノートファイルがない場合は、作成します。

引数

`filePath`、*`{bForceCreate}`*

- *`filePath`* 引数には、デザインノートファイルが関連付けられているメインファイルへのパスを file:// URL 形式で指定します。
- *`bForceCreate`* 引数には、サイトでデザインノートが使用不可になっている場合、または *`filePath`* 引数がどのサイトにも関連付けられていない場合にも、デザインノートファイルを作成するかどうかを示すブール値を指定します。

戻り値

デザインノートファイルのファイルハンドル。ファイルが開かれなかった場合や作成されなかった場合はゼロ。

例

詳細については、23 ページの MMNotes.set() を参照してください。

MMNotes.remove()

説明

指定されたデザインノートファイルから指定されたキーとその値を削除します。

引数

fileHandle、*keyName*

- *fileHandle* 引数には、MMNotes.open() 関数から返されるファイルハンドルを指定します。
- *keyName* 引数には、削除するキーの名前を含むストリングを指定します。

戻り値

ブール値。true は操作が成功したことを示し、false は失敗したことを示します。

MMNotes.set()

説明

デザインノートファイル内で 1 つのキー / 値ペアを作成または更新します。

引数

fileHandle、*keyName*、*valueString*

- *fileHandle* 引数には、MMNotes.open() 関数から返されるファイルハンドルを指定します。
- *keyName* 引数には、キーの名前を含むストリングを指定します。
- *valueString* 引数には、値を含むストリングを指定します。

戻り値

ブール値。true は操作が成功したことを示し、false は失敗したことを示します。

例

以下の例では、dreamcentral サイト内の "peakhike99/index.html" というファイルに関連付けられているデザインノートファイルを開き、新しいキー / 値のペアを追加すると共に既存キーの値を変更して、デザインノートファイルを閉じます。

```
var noteHandle = MMNotes.open('file:///c|/sites/dreamcentral/peakhike99/
    index.html',true);
if(noteHandle > 0){
    MMNotes.set(noteHandle,"Author","M. G. Miller");
    MMNotes.set(noteHandle,"Last Changed","August 28, 1999");
    MMNotes.close(noteHandle);
}
```

デザインノート C API

MMNotes 共有ライブラリには JavaScript API 以外に、他のアプリケーションによるデザインノートファイルの作成を可能にする C API も含まれています。Dreamweaver で MMNotes 共有ライブラリを使っているときは、これらの C 関数を直接呼び出す必要はありません。各関数の JavaScript バージョンによって呼び出されるためです。

この項では、関数とその引数および戻り値について説明します。これらの関数とデータタイプは、Dreamweaver アプリケーションフォルダ内の "Extending/c_files" フォルダに入っている "MMInfo.h" ファイルに定義されています。

void CloseNotesFile()

説明

指定されたデザインノートファイルを閉じ、加えられた変更内容を保存します。キーと値のペアがデザインノートファイルからすべて削除されると、デザインノートファイルは削除されます。最後のデザインノートファイルが削除されると、"_notes" フォルダも削除されます。

引数

noteHandle

- *noteHandle* 引数には、OpenNotesFile() 関数から返されるファイルハンドルを指定します。

戻り値

なし

BOOL FilePathToLocalURL()

説明

指定されたローカルドライブパスを file:// URL 形式に変換します。

引数

const char* *drivePath*、char* *localURLBuf*、int *localURLMaxLen*

- *drivePath* 引数には、完全なドライブパスを含むストリングを指定します。
- *localURLBuf* 引数には、file:// URL を保存するバッファを指定します。
- *localURLMaxLen* 引数には、*localURLBuf* の最大サイズを指定します。

戻り値

ブール値。true は操作が成功したことを示し、false は失敗したことを示します。*localURLBuf* 引数が示すバッファに、file:// URL の値が設定されます。

BOOL GetNote()

説明

指定されたデザインノートファイルに含まれている指定されたキーの値を取得します。

引数

FileHandle *noteHandle*、const char *keyName[64]*、char* *valueBuf*、int *valueBufLength*

- *noteHandle* 引数には、OpenNotesFile() 関数から返されるファイルハンドルを指定します。
- *keyName[64]* 引数には、キーの名前を含むストリングを指定します。
- *valueBuf* 引数には、値を保存するバッファを指定します。
- *valueBufLength* 引数には、GetNoteLength(*noteHandle*, *keyName*) から返される整数を指定します。これは、値バッファの最大長を示します。

戻り値

ブール値。`true` は操作が成功したことを示し、`false` は失敗したことを示します。*`valueBuf`* 引数が示すバッファに、キーの値が設定されます。

例

次のコードは "welcome.html" ファイルに関連付けられているデザインノートファイル内の `comments` キーの値を取得します。

```
FileHandle noteHandle = OpenNotesFile("file:///c|/sites/avocado8/iwjs/welcome.html");
if(noteHandle > 0){
    int valueLength = GetNoteLength( noteHandle, "comments");
    char* valueBuffer = new char[valueLength + 1];
    GetNote(noteHandle, "comments", valueBuffer, valueLength + 1);
    printf("Comments: %s",valueBuffer);
    CloseNotesFile(noteHandle);
}
```

int GetNoteLength()

説明

指定されたキーに関連付けられている値の長さを取得します。

引数

FileHandle *`noteHandle`*、const char *`keyName[64]`*

- *`noteHandle`* 引数には、`OpenNotesFile()` 関数から返されるファイルハンドルを指定します。
- *`keyName[64]`* 引数には、キーの名前を含むストリングを指定します。

戻り値

値の長さを示す整数。

例

詳細については、24 ページの BOOL GetNote() を参照してください。

int GetNotesKeyCount()

説明

指定されたデザインノートファイルに含まれているキー / 値ペアの数を取得します。

引数

FileHandle *`noteHandle`*

- *`noteHandle`* 引数には、`OpenNotesFile()` 関数から返されるファイルハンドルを指定します。

戻り値

デザインノートファイルに含まれるキー / 値ペアの数を表す整数。

BOOL GetNotesKeys()

説明

デザインノートファイルに含まれるすべてのキーのリストを取得します。

引数

FileHandle *noteHandle*、char* *keyBufArray[64]*、int *keyArrayMaxLen*

- *noteHandle* 引数には OpenNotesFile() から返されるファイルハンドルを指定します。
- *keyBufArray[64]* 引数には、キーを保存するバッファ配列を指定します。
- *keyArrayMaxLen* 引数には、GetNotesKeyCount(*noteHandle*) から返される整数を指定します。これは、キーバッファ配列内の最大アイテム数を示します。

戻り値

ブール値。true は操作が成功したことを示し、false は失敗したことを示します。*keyBufArray* 引数が示すバッファ配列に、キーの名前が設定されます。

例

次のコードは "welcome.html" ファイルに関連付けられているデザインノートファイル内のすべてのキーの名前と値を印刷します。

```
typedef char[64] InfoKey;
FileHandle noteHandle = OpenNotesFile("file:///c|/sites/avocado8/iwjs/welcome.html");
if(noteHandle > 0){
    int keyCount = GetNotesKeyCount(noteHandle);
    if (keyCount <= 0)
        return;
    InfoKey* keys = new InfoKey[keyCount];
    BOOL succeeded = GetNotesKeys(noteHandle, keys, keyCount);

    if (succeeded){
        for (int i=0; i < keyCount; i++){
            printf("Key is: %s\n", keys[i]);
            printf("Value is: %s\n\n", GetNote(noteHandle, keys[i]);
        }
    }
    delete []keys;
}
CloseNotesFile(noteHandle);
```

BOOL GetSiteRootForFile()

説明

指定されたデザインノートファイルのサイトルートを確認します。

引数

const char* *filePath*、char* *siteRootBuf*、int *siteRootBufMaxLen*、{InfoPrefs* *infoPrefs*}

- *filePath* 引数には、サイトルートを探すファイルの file:// URL を指定します。
- *siteRootBuf* 引数には、サイトルートを保存するバッファを指定します。
- *siteRootBufMaxLen* 引数には、*siteRootBuf* が示すバッファの最大サイズを指定します。
- *infoPrefs* 引数 (オプション) には、サイトの環境設定を保存する struct への参照を指定します。

戻り値

ブール値。true は操作が成功したことを示し、false は失敗したことを示します。*siteRootBuf* 引数には、サイトルートを保存したバッファのアドレスが設定されます。*infoPrefs* 引数を指定した場合、この関数はサイトのデザインノート環境設定も返します。InfoPrefs 構造体には、bUseDesignNotes と bUploadDesignNotes という 2 つの変数があります。どちらの変数もデータタイプは BOOL です。

BOOL GetVersionName()

説明

MMNotes 共有ライブラリのバージョン名を取得します。このバージョン名は、共有ライブラリを実装したアプリケーションを示します。

引数

`char* versionNameBuf`、`int versionNameBufMaxLen`

- *`versionNameBuf`* 引数には、バージョン名を保存するバッファを指定します。
- *`versionNameBufMaxLen`* 引数には、*`versionNameBuf`* 引数が示すバッファの最大サイズを指定します。

戻り値

ブール値。`true` は操作が成功したことを示し、`false` は失敗したことを示します。*`versionNameBuf`* 引数には、`"Dreamweaver"` が保存されます。

BOOL GetVersionNum()

説明

MMNotes 共有ライブラリのバージョン番号を取得します。これにより、特定の関数が使用可能かどうかを確認できます。

引数

`char* versionNumBuf`、`int versionNumBufMaxLen`

- *`versionNumBuf`* 引数には、バージョン番号を保存するバッファを指定します。
- *`versionNumBufMaxLen`* 引数には、*`versionNumBuf`* が示すバッファの最大サイズを指定します。

戻り値

ブール値。`true` は操作が成功したことを示し、`false` は失敗したことを示します。*`versionNumBuf`* 引数が示すバッファに、バージョン番号が設定されます。

BOOL LocalURLToFilePath()

説明

指定された file:// URL をローカルドライブパスに変換します。

引数

const char* *`localURL`*、char* *`drivePathBuf`*、int *`drivePathMaxLen`*

- *`localURL`* 引数には、ローカルファイルへのパスを file:// URL 形式で指定します。
- *`drivePathBuf`* 引数には、ローカルドライブパスを保存するバッファを指定します。
- *`drivePathMaxLen`* 引数には、*`drivePathBuf`* 引数が示すバッファの最大サイズを指定します。

戻り値

操作が成功した場合は `true` を、失敗した場合は `false` を示すブール値。*`drivePathBuf`* 引数が示すバッファに、ローカルドライブパスが設定されます。

FileHandle OpenNotesFile()

説明

指定されたファイルに関連付けられているデザインノートファイルを開きます。デザインノートファイルがない場合は、作成します。

引数

`const char* localFileURL`、{`BOOL bForceCreate`}

- `localFileURL` 引数は、デザインノートファイルが関連付けられているメインファイルへのパスを含む file:// URL 形式のストリングです。
- `bForceCreate` 引数には、サイトでデザインノートが使用不可になっている場合、または `localFileURL` 引数に指定したパスがどのサイトにも関連付けられていない場合にも、デザインノートファイルを作成するかどうかを示すブール値を指定します。

FileHandle OpenNotesFilewithOpenFlags()

説明

指定されたファイルに関連付けられているデザインノートファイルを開きます。デザインノートファイルがない場合は、作成します。このファイルは、読み取り専用モードで開くことができます。

引数

`const char* localFileURL`、{`BOOL bForceCreate`}、{`BOOL bReadOnly`}

- `localFileURL` 引数は、デザインノートファイルが関連付けられているメインファイルへのパスを含む file:// URL 形式のストリングです。
- `bForceCreate` 引数には、サイトでデザインノートが使用不可になっている場合、またはパスがどのサイトにも関連付けられていない場合にも、デザインノートファイルを作成するかどうかを示すブール値を指定します。デフォルト値は `false` です。この引数はオプションですが、3 番目の引数を指定する場合は必ず指定します。
- `bReadOnly` 引数 (オプション) には、ファイルを読み取り専用モードで開くかどうかを示すブール値を指定します。デフォルト値は `false` です。`bReadOnly` 引数は、バージョン 2 以降の "MMNotes.dll" ファイルで指定できます。

BOOL RemoveNote()

説明

指定されたデザインノートファイルから指定されたキーとその値を削除します。

引数

`FileHandle noteHandle`、`const char keyName[64]`

- `noteHandle` 引数には、`OpenNotesFile()` 関数から返されるファイルハンドルを指定します。
- `keyName[64]` 引数には、削除するキーの名前を含むストリングを指定します。

戻り値

ブール値。`true` は操作が成功したことを示し、`false` は失敗したことを示します。

BOOL SetNote()

説明

デザインノートファイル内で 1 つのキー / 値ペアを作成または更新します。

引数

FileHandle noteHandle、const char *keyName[64]*、const char* *value*

- *noteHandle* 引数には、OpenNotesFile() 関数から返されるファイルハンドルを指定します。
- *keyName[64]* 引数には、キーの名前を含むストリングを指定します。
- *value* 引数には、値を含むストリングを指定します。

戻り値

ブール値。true は操作が成功したことを示し、false は失敗したことを示します。

第 5 章 : Fireworks との統合

FWLaunch は、オブジェクト、コマンド、ビヘイビア、およびプロパティインスペクタの作成者と Adobe® Fireworks® との間の通信を可能にする C 共有ライブラリです。FWLaunch を使用すると、Fireworks のユーザーインターフェイス (UI) を開き、独自の JavaScript API を通じて Fireworks にコマンドを渡す JavaScript を記述できます。この JavaScript API については、『Fireworks 拡張ガイド』に記載されています。Adobe® Dreamweaver® CS3 における C ライブラリと JavaScript インタープリタとの対話に関する一般情報については、『Dreamweaver 拡張ガイド』の「C レベル拡張機能」を参照してください。

FWLaunch API

FWLaunch オブジェクトを使用すると、拡張機能から Fireworks を開き、Fireworks JavaScript API を使用して Fireworks の操作を実行し、Dreamweaver に値を返すことができます。本章では、FWLaunch 通信 API とその使用方法について説明します。

FWLaunch.bringDWToFront()

対応バージョン

Dreamweaver 3、Fireworks 3

説明

Dreamweaver を前面に移動します。

引数

なし

戻り値

なし

FWLaunch.bringFWToFront()

対応バージョン

Dreamweaver 3、Fireworks 3

説明

Fireworks が実行中であれば、それを前面に移動します。

引数

なし

戻り値

なし

FWLaunch.execJsInFireworks()

対応バージョン

Dreamweaver 3、Fireworks 3

説明

指定した JavaScript または JavaScript ファイルへの参照を Fireworks に渡して実行します。

引数

`javascriptOrFileURL`

- `javascriptOrFileURL` 引数には、リテラル JavaScript のストリングまたは JavaScript ファイルへのパスを file:// URL 形式で指定します。

戻り値

JavaScript が渡された場合は cookie オブジェクト、そうでない場合は、以下のエラーが発生したことを示す 0 以外のエラーコード。

- 無効な使い方。`javascriptOrFileURL` 引数が `null` 値または空白のストリングとして指定されているか、JS ファイルまたは JSF ファイルへのパスが無効であることを示します。
- ファイル I/O エラー。ディスクに空き領域がないため、Fireworks が応答ファイルを作成できないことを示します。
- ユーザーが実行している Dreamweaver が有効なバージョン (3 以降のバージョン) でないことを Dreamweaver に知らせるエラー。
- Fireworks プロセス開始時のエラー。関数が開いている Fireworks が有効なバージョン (3 以降のバージョン) でないことを示します。
- 操作がキャンセルされました。

FWLaunch.getJsResponse()

対応バージョン

Dreamweaver 3、Fireworks 3

説明

この関数は、Fireworks が `FWLaunch.execJsInFireworks()` 関数によって渡された JavaScript を実行中か、スクリプトが正常に終了したか、またはエラーが発生したかを判断します。

引数

`progressTrackerCookie`

- `progressTrackerCookie` 引数には、`FWLaunch.execJsInFireworks()` 関数から返される cookie オブジェクトを指定します。

戻り値

操作が正常に完了した場合は、`FWLaunch.execJsInFireworks()` 関数に渡されたスクリプトの結果を含むストリング、Fireworks が JavaScript を実行中の場合は `null` 値、そうでない場合は以下のエラーが発生したことを示す 0 以外のエラーコード。

- 無効な使い方。Fireworks がスクリプトを実行中に JavaScript エラーが発生したことを示します。
- ファイル I/O エラー。ディスクに空き領域がないため、Fireworks が応答ファイルを作成できないことを示します。
- ユーザーが実行している Dreamweaver が有効なバージョン (3 以降のバージョン) でないことを Dreamweaver に知らせるエラー。
- Fireworks プロセス開始時のエラー。関数が開いている Fireworks が有効なバージョン (3 以降のバージョン) でないことを示します。
- 操作がキャンセルされました。

例

以下のコードを実行すると、ストリング "prompt('Please enter your name:')" が FWLaunch.execJsInFireworks() 関数に渡され、結果がチェックされます。

```
var progressCookie = FWLaunch.execJsInFireworks("prompt('Please enter your name:')");
var doneFlag = false;
while (!doneFlag){
    // 0.5 秒ごとに完了を確認します。
    setTimeout('checkForCompletion()',500);
}

function checkForCompletion(){
    if (progressCookie != null) {
        var response = FWLaunch.getJsResponse(progressCookie);
        if (response != null) {
            if (typeof(response) == "number") {
                // エラーまたはユーザーによるキャンセルが発生。ウィンドウを閉じます。
                // エラーが発生したことをユーザーに通知します。
                window.close();
                alert("An error occurred.");
            }else{
                // 有効な応答を受け取りました。
                alert("Nice to meet you, " + response);
                window.close();
            }
                doneFlag = true;
        }
    }
}
```

FWLaunch.mayLaunchFireworks()

対応バージョン

Dreamweaver 2、Fireworks 2

説明

Fireworks 最適化セッションを開くことができるかどうかを判断します。

引数

なし

戻り値

プラットフォームが Windows と Macintosh のどちらであるかを示すブール値。Macintosh の場合、この値は別の Fireworks 最適化セッションが既に実行中であるかどうかを示します。

FWLaunch.optimizeInFireworks()

対応バージョン

Dreamweaver 2、Fireworks 2

説明

指定されたイメージに対して Fireworks 最適化セッションを開きます。

引数

docURL、*imageURL*、*{targetWidth}*、*{targetHeight}*

- *docURL* 引数には、アクティブなドキュメントへのパスを file:// URL 形式で指定します。
- *imageURL* 引数には、選択されたイメージへのパスを指定します。相対パスを使用する場合は、*docURL* 引数で指定したパスを基準に指定します。
- *targetWidth* 引数 (オプション) には、サイズ変更するイメージの幅を定義します。
- *targetHeight* 引数 (オプション) には、サイズ変更するイメージの高さを定義します。

戻り値

指定されたイメージ用の Fireworks 最適化セッションが正常に開始された場合は 0、そうでない場合は、以下のエラーが発生したことを示す 0 以外のエラーコード。

- 無効な使い方。*docURL* 引数と *imageURL* 引数のいずれかまたは両方に、null 値または空白のストリングが指定されたことを示します。
- ファイル I/O エラー。ディスクに空き領域がないため、Fireworks が応答ファイルを作成できないことを示します。
- ユーザーが実行している Dreamweaver が有効なバージョン (2 以降のバージョン) でないことを Dreamweaver に知らせるエラー。
- Fireworks プロセス開始時のエラー。関数が開いている Fireworks が有効なバージョン (2 以降のバージョン) でないことを示します。
- 操作がキャンセルされました。

FWLaunch.validateFireworks()

対応バージョン

Dreamweaver 2、Fireworks 2

説明

この関数は、Fireworks の指定されたバージョンをユーザーのハードディスク上で検索します。

引数

{versionNumber}

- *versionNumber* 引数 (オプション) には、2 以上の浮動小数点値を指定します。この値は、必要な Fireworks のバージョンを表します。この引数を指定しない場合のデフォルト値は、2 です。

戻り値

Fireworks の指定されたバージョンが見つかったかどうかを示すブール値。

例

次のコードは、Fireworks がインストール済みであるかどうかを調べます。

```
if (FWLaunch.validateFireworks(6.0)){
    alert("Fireworks 6.0 or later is installed.");
}else{
    alert("Fireworks 6.0 is not installed.");
}
```

FWLaunch 通信 API の簡単な使用例

次のコマンドにより、Fireworks からユーザー名を入力するよう要求され、その名前が Dreamweaver に返されます。

```
<html>
<head>
<title>Prompt in Fireworks</title>
<meta http-equiv="Content-Type" content="text/html; ¬
charset=iso-8859-1">
<script>

function commandButtons(){
    return new Array("Prompt", "promptInFireworks()", "Cancel", ¬
    "readyToCancel()", "Close","window.close()");
}

var gCancelClicked = false;
var gProgressTrackerCookie = null;

function readyToCancel() {
    gCancelClicked = true;
}

function promptInFireworks() {
    var isFireworks3 = FWLaunch.validateFireworks(3.0);
    if (!isFireworks3) {
        alert("You must have Fireworks 3.0 or later to use this ¬
        command");
    return;
}

    // Fireworks に、prompt() メソッドの実行を指示します。
    gProgressTrackerCookie = FWLaunch.execJsInFireworks¬
    ("prompt('Please enter your name:')");

    // null は起動されていないことを、数値はエラーコードを示します。
    if (gProgressTrackerCookie == null || ¬
    typeof(gProgressTrackerCookie) == "number") {
        window.close();
        alert("an error occurred");
        gProgressTrackerCookie = null;
    } else {
        // Fireworks を前面に移動します。
        FWLaunch.bringFWToFront();
        // Fireworks での処理が完了しているか、チェックを開始します。
        checkOneMoreTime();
    }
}

function checkOneMoreTime() {
    // checkJsResponse() を 0.5 秒ごとに呼び出して、Fireworks での
    // 処理が完了しているかどうかチェックします。
    window.setTimeout("checkJsResponse();", 500);
}

function checkJsResponse() {
    var response = null;

    // ユーザーが [キャンセル] ボタンをクリックした場合、ウィンドウを閉じます。
    if (gCancelClicked) {
        window.close();
```

```
        alert("cancel clicked");
    } else {
        // 処理を続行中の場合は、Fireworks に進捗状況を確認します。
        if (gProgressTrackerCookie != null)
            response = FWLaunch.getJsResponse(gProgressTrackerCookie);

        if (response == null) {
            // まだ応答を待っている場合、0.5 秒後に再度
            // 呼び出します。
            checkOneMoreTime();
        } else if (typeof(response) == "number") {
            // 応答が数字の場合、エラーが発生したことを示します。
            // Fireworks でユーザーによるキャンセルが行われました。
            window.close();
            alert("an error occurred.");

        } else {
            // 有効な応答を受け取りました。戻り値は、常に有効で
            // あるとは限りません。これは、Fireworks の関数がすべてストリングを
            // 戻すとは限らないためです。この関数はストリングを戻すことが
            // わかっているため、受け取った応答をユーザーに表示できます。
            window.close();
            FWLaunch.bringDWToFront();// Dreamweaver を前面に移動します。
            alert("Nice to meet you, " + response + "!");
        }
    }
}

</script>
</head>
<body>
<form>
<table width="313" nowrap>
<tr>
<td>このコマンドは、Fireworks に prompt() 関数の実行を ¬
指示します。[Prompt] をクリックすると、Fireworks は前面に表示され、¬
ダイアログボックスへの値の入力が要求されます。次に、入力した値が ¬
Dreamweaver に返され、警告メッセージに表示されます。</td>
</tr>
</table>
</form>
</body>
</html>
```

第 6 章 : Flash との統合

Adobe® Dreamweaver® CS3 は、Flash Generator テンプレートファイルを利用して新しい Flash オブジェクトを作成する Flash オブジェクト API を引き続きサポートすると共に、Flash エレメントもサポートするようになりました。本章では、Flash エレメント (SWC ファイル) の操作方法を説明し、さらに、Flash Generator テンプレート (SWT ファイル) からの Flash オブジェクト (SWF ファイル) の作成について詳しく説明します。

Dreamweaver のオブジェクトまたはコマンドに Flash のコンテンツをそのまま追加する方法の詳細については、『Dreamweaver 拡張ガイド』を参照してください。

Flash エレメントの動作

Flash エレメントは、SWC ファイルとしてパッケージ化されます。SWC ファイルは、コンパイルされたムービークリップのコンポーネントで、Adobe およびサードパーティの製品で使用するために Flash で生成されます。Dreamweaver では、[挿入] バー、[挿入] メニュー、またはツールバーを介してユーザーにこれらのコンポーネントを提供することができます。Flash エレメントは Flash オーサリングツールを使用してデベロッパーが作成しますが、Dreamweaver は Flash エレメントのプロパティを解析し、`param` タグ (`object` タグの子タグ) によってその内容を表現できます。ユーザーは、公開時に `param` タグの属性を編集して、エレメントのプロパティを変更できます (Dreamweaver におけるコンポーネントプロパティの操作の詳細については、『Dreamweaver ユーザーガイド』を参照)。

Flash エレメントの挿入

Flash エレメントは、Extension Manager を使用してインストールされます。Dreamweaver は、[挿入] バーや [挿入] メニューで使用できるオブジェクトと同様の方法で、Flash エレメントをドキュメントに追加します (Dreamweaver オブジェクトの操作の詳細については、『Dreamweaver 拡張ガイド』の「[挿入] バーオブジェクト」を参照)。ユーザーは、[挿入] バーでオブジェクトをクリックするか、[挿入] メニューでメニューオプションを選択することにより、コードストリングをドキュメントに追加できます。Flash エレメントは、[挿入] バーまたは [挿入] メニューを通じてユーザーに提供されます。つまり、デベロッパーは、"Configuration/Objects/FlashElements" フォルダまたはそのいずれかのサブフォルダに既にインストールされている有効な Flash エレメントファイルを、[挿入] バーや [挿入] メニューに追加することができます。拡張機能のデベロッパーは、オブジェクト定義ファイルで JavaScript 関数 113 ページの dom.insertFlashElement() を使用して、使用可能な Flash エレメントをドキュメントに追加することができます。ユーザーが Flash エレメントオブジェクトを選択すると、Dreamweaver は SWC ファイルをアンパックします。SWC ファイルには、Flash コンテンツ (SWF ファイル)、およびユーザーが編集できるパラメータを詳述するファイルが含まれています。Dreamweaver はこの SWF ファイルをユーザーのドキュメントに挿入します。

[挿入] バーへの Flash エレメントの追加

他のオブジェクトと同様に、Flash エレメントを [挿入] バーに追加するには `button` タグを使用します。ただし、Flash エレメントの `button` タグには、`file` 属性と `command` 属性が必要です。これらの属性がない Flash エレメントは、ドキュメントに正常に追加されません (button タグの詳細については、『Dreamweaver 拡張ガイド』の「[挿入] バーオブジェクト」を参照)。`file` 属性を使用して、エレメントのソースファイルがある場所を "Objects" フォルダを基準に設定します。次に、`command` 属性を使用して、ユーザーが [挿入] バーでそのボタンをクリックしたときに Dreamweaver が `dom.insertFlashElement()` 関数を実行するように指示します。

次の例は、"inserbar.xml" ファイルに配置するコードを示しています。このコードは、Flash エレメントボタンを表示する場所に応じて、該当する `category` タグまたは `menubutton` タグの子タグとして配置します。

```
<button id="FlashElement_Nav"
name="Navigation"
file="FlashElements¥nav.swc"
command="dw.getDocumentDOM().insertFlashElement('nav.swc')" />
```

注意 : [挿入] バーに表示される *Flash* エレメントのイメージは、*SWC* ファイル内で決定されます。また、*Flash* エレメントオブジェクトの *`button`* タグには、*file* 属性を定義する必要があります。

メニューへの Flash エレメントの追加

Dreamweaver では、[挿入] メニューや他のメニューにも Flash エレメントを配置できます。Flash エレメントのメニュー項目の場所を指定するには、"menus.xml" ファイル形式で JavaScript 関数 113 ページの dom.insertFlashElement() を使用します (『Dreamweaver 拡張ガイド』の「メニューおよびメニューコマンド」を参照)。次の例では、[挿入]-[Flash エレメント] メニューで "Navigation" というエレメントを使用できるように、"menus.xml" ファイル内のコードで指定しています。

```
<menuitem name="Navigation"
key=""command="dw.getDocumentDOM().insertFlashElement('nav.swc')"
enabled="(dw.getFocus() != 'browser') && (dw.getDocumentDOM() != null && ¬
    dw.getDocumentDOM().getParseMode() == 'html')"
id="DWMenu_Insert_FlashElement_Nav" />
```

Flash オブジェクト API

拡張機能のデベロッパーは、Flash Generator を使って簡単な Flash コンテンツを作成するオブジェクトを、Flash オブジェクト API を使用して構築できます。この API では、Flash Generator テンプレートにパラメータを設定して、SWF ファイルまたはイメージファイルを出力できます。また、Flash オブジェクトの新規作成、読み取り、および操作を行うことができます。[Flash] ボタンおよび Flash テキストの各機能は、この API を使って構築されます。

SWT ファイルは、Flash Generator のテンプレートファイルで、Flash オブジェクトファイルの構築に必要なすべての情報を含んでいます。この API 関数を使用して、SWT ファイルのパラメータを実際の値に置換し、SWT ファイルから SWF ファイル (またはイメージファイル) を新規作成することができます。Flash の詳細については、Flash のマニュアルを参照してください。`SWFFile` オブジェクトのメソッドには、以下の関数があります。

SWFFile.createFile()

説明

指定されたテンプレートおよびパラメータの配列を使って、新しい Flash オブジェクトファイルを生成します。GIF、PNG、JPEG、および MOV フォーマットのファイル名が指定された場合は、それぞれのフォーマットバージョンのタイトルも作成します。

使用しないオプションのパラメータの後にオプションのパラメータを指定する場合は、使用しないパラメータに空白のストリングを指定します。たとえば、GIF ファイルを指定しないで PNG ファイルを指定する場合は、PNG ファイル名の前に空白のストリングを指定します。

引数

`templateFile`、*`templateParams`*、*`swfFileName`*、*`{gifFileName}`*、*`{pngFileName}`*、*`{jpgFileName}`*、*`{movFileName}`*、*`{generatorParams}`*

- *`templateFile`* 引数には、テンプレートファイルへのパスを file:// URL 形式で指定します。このファイルには SWT ファイルを指定できます。
- *`templateParams`* 引数には、名前 / 値のペア、つまり、SWT ファイル内のパラメータとその値の組み合わせのリストを指定します。Dreamweaver で SWF ファイルが Flash オブジェクトとして認識されるようにするには、最初のパラメータに対して `"dwType"` を設定します。このパラメータの値には、`"Flash Text"` などのオブジェクトタイプの名前を表すストリングを設定します。
- *`swfFileName`* 引数には、SWF ファイルの出力ファイル名を file:// URL 形式で指定します。この引数を無視する場合は、空白のストリングを指定します。
- *`gifFileName`* 引数には、GIF ファイルの出力ファイル名を file:// URL 形式で指定します。この引数はオプションです。
- *`pngFileName`* 引数には、PNG ファイルの出力ファイル名を file:// URL 形式で指定します。この引数はオプションです。
- *`jpgFileName`* 引数には、JPEG ファイルの出力ファイル名を file:// URL 形式で指定します。この引数はオプションです。

- *movFileName* 引数には、QuickTime ファイルの出力ファイル名を file:// URL 形式で指定します。この引数はオプションです。
- *generatorParams* 引数には、オプションの Generator コマンドラインフラグを表すストリングの配列を指定します。この引数はオプションです。配列内では、各フラグの後にそのデータアイテムを指定する必要があります。よく使用されるフラグの一部を次の表に示します。

オプションフラグ	データ	説明	例
-defaultsize	幅、高さ	出力イメージのサイズを、指定された幅と高さに設定します。	"-defaultsize", "640", "480"
-exactFit	なし	出力イメージの内容を、指定された出力サイズにちょうど収まるように拡大します。	"-exactFit"

戻り値

次の値のいずれかを含むストリング。

- "noError" は、呼び出しが正常に終了したことを意味します。
- "invalidTemplateFile" は、指定されたテンプレートファイルが無効であるか、見つからなかったことを意味します。
- "invalidOutputFile" は、指定された出力ファイル名の少なくとも 1 つが無効であることを意味します。
- "invalidData" は、名前 / 値のペア *templateParams* の 1 つまたは複数が無効であることを意味します。
- "initGeneratorFailed" は、Generator を初期化できなかったことを意味します。
- "outOfMemory" は、メモリが不足しているため、操作を完了できないことを意味します。
- "unknownError" は、未知のエラーが発生したことを意味します。

例

次の JavaScript は、"myType" タイプの Flash オブジェクトファイルを作成します。その際に、テンプレートファイル内のすべての "text" ストリングを "Hello World" ストリングに置き換えます。このスクリプトでは、SWF ファイルと共に GIF ファイルを作成します。

```
var params = new Array;
params[0] = "dwType";
params[1] = "myType";
params[2] = "text";
params[3] = "Hello World";
errorString = SWFFile.createFile( "file:///MyMac/test.swt", ¬
params, "file:///MyMac/test.swf", "file:///MyMac/test.gif");
```

SWFFile.getNaturalSize()

説明

非圧縮 Flash コンテンツの本来のサイズを返します。

引数

fileName

- *fileName* 引数には、Flash コンテンツへのパスを file:// URL 形式で指定します。

戻り値

非圧縮 SWF ファイルの幅と高さを表す 2 つの要素を含む配列。ファイルが非圧縮 SWF ファイルでない場合は、null 値が返されます。

SWFFile.getObjectType()

説明

Flash オブジェクトのタイプを返します。これは、SWFFile.createFile() 関数がファイルを作成したときに dwType パラメータに渡された値です。

引数

fileName

- *fileName* 引数には、Flash オブジェクトファイルへのパスを file:// URL 形式で指定します。通常、このファイルは SWF ファイルです。

戻り値

オブジェクトタイプを表すストリング。ファイルが Flash オブジェクトファイルではない場合、またはファイルが見つからなかった場合は、null が返されます。

例

次のコードは、"test.swf" ファイルがタイプ myType の Flash オブジェクトであるかどうかをチェックします。

```
if ( SWFFile.getObjectType("file:///MyMac/test.swf") == "myType" ){
    alert ("This is a myType object.");
}else{
    alert ("This is not a myType object.");
}
```

SWFFile.readFile()

説明

Flash オブジェクトファイルを読み取ります。

引数

fileName

- *fileName* 引数には、Flash オブジェクトファイルへのパスを file:// URL 形式で指定します。

戻り値

ストリングの配列。この配列の最初のエレメントは、テンプレート SWT ファイルへの完全なパスです。それに続くストリングは、オブジェクトのパラメータ (名前 / 値のペア) です。配列内では、それぞれの名前の後にその値が続きます。最初の名前 / 値ペアは、"dwType" とその値です。ファイルが見つからなかった場合、またはファイルが Flash オブジェクトファイルでない場合、関数は null 値を返します。

例

var params = SWFFile.readFile("file:///MyMac/test.swf") を呼び出すと、パラメータの配列に次の値が返されます。

```
"file:///MyMac/test.swt"      // この .swf ファイルの作成に必要なテンプレートファイル
"dwType"                      // 最初のパラメータ
"myType"                      // 最初のパラメータの値
"text"                        // 2 番目のパラメータ
"Hello World"                 // 2 番目のパラメータの値
```

第 7 章 : データベース API

データベース API の関数を使用すると、データベース接続とデータベースに格納されたアクセス情報を管理できます。データベース API には、データベース接続の管理とアクセスという 2 つの目的があります。

データベース接続の管理では、データベースへの接続に必要なユーザー名とパスワードを取得したり、データベース接続のダイアログボックスを表示します。

データベース情報へのアクセスでは、たとえば、データベースのスキーマまたは構造を記述するメタデータを取得します。このメタデータには、テーブル、列、ストアードプロシージャ、およびビューの名前などの情報が含まれます。データベースクエリーやストアードプロシージャの実行結果を表示することもできます。この API を介してデータベースにアクセスするときは、SQL (Structured Query Language) ステートメントを使用します。

データベース API 関数は、Web アプリケーションが配置されるランタイムではなく、デザイン時、つまりユーザーが Web アプリケーションを構築する際に使用されます。

これらの関数は、あらゆる拡張機能で使用できます。Adobe® Dreamweaver® CS3 のサーバービヘイビア、データフォーマット、およびデータソースの各 API ではいずれも、これらのデータベース関数が使用されています。

本章では関数をグループ分けして、それぞれを以下の項で説明します。

データベース API 関数の動作

次の例は、サーバービヘイビア関数 `getDynamicBindings()` を "Recordset.js" 用に定義するところを示しています。この例では、`MMDB.getColumnAndTypeList()` 関数を使用しています。

```
function getDynamicBindings(ss)
{
    var serverModel = dw.getDocumentDOM().serverModel.getServerName();
    var bindingsAndTypeArray = new Array();
    var connName = ss.connectionName;
    var statement = ss.source;
    var rsName = ss.rsName;

    // SQL コメントを削除します。
    statement = statement.replace(/\/\*[\S\s]*?\*\//g, " ");
    var bIsSimple = ParseSimpleSQL(statement);
    statement = stripCFIFSimple(statement);

    if (bIsSimple) {
    statement = RemoveWhereClause(statement,false);
    } else {
    var pa = new Array();

        if (ss.ParamArray != null) {
        for (var i = 0; i < ss.ParamArray.length; i++) {
            pa[i] = new Array();
            pa[i][0] = ss.ParamArray[i].name;
            pa[i][1] = ss.ParamArray[i].value;
        }
        }

    var statement = replaceParamsWithVals(statement, pa, serverModel);
}
    bindingsAndTypeArray = MMDB.getColumnAndTypeList(connName, statement);
    return bindingsAndTypeArray;
}
```

データベース接続関数

データベース接続関数を使用すると、Dreamweaver に用意されている ADO、ColdFusion、および JDBC 接続などのあらゆる接続を確立および管理できます。これらの関数は、[接続] ダイアログボックスとだけ接続し、データベースにはアクセスしません。データベースにアクセスする関数の詳細については、52 ページのデータベースアクセス関数を参照してください。

MMDB.deleteConnection()

対応バージョン

Dreamweaver MX

説明

この関数は、指定されたデータベース接続を削除します。

引数

connName

- *connName* 引数は、[接続] ダイアログボックスで指定されたデータベース接続の名前です。この引数により、削除対象のデータベース接続が名前で識別されます。

戻り値

なし

例

次の例では、1 つのデータベース接続を削除します。

```
function clickedDelete()
{
    var selectedObj = dw.serverComponents.getSelectedNode();
    if (selectedObj && selectedObj.objectType=="Connection")
    {
        var connRec = MMDB.getConnection(selectedObj.name);
        if (connRec)
        {
            MMDB.deleteConnection(selectedObj.name);
            dw.serverComponents.refresh();
        }
    }
}
```

MMDB.getColdFusionDsnList()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

この関数は、`getRDSUserName()` および `getRDSPassword()` 関数を使用して、サイトサーバーから ColdFusion データソース名 (DSN) を取得します。

引数

なし

戻り値

現在のサイトに対してサーバーで定義されている ColdFusion DSN を含む配列。

MMDB.getConnection()

対応バージョン

Dreamweaver UltraDeveloper 4、Dreamweaver MX で機能強化。

説明

指定された接続オブジェクトを取得します。

引数

name

- *name* 引数は、参照する接続の名前を指定するストリング変数です。

戻り値

指定された接続オブジェクトへの参照。接続オブジェクトには、以下のプロパティがあります。

プロパティ	説明
name	接続名
type	useHTTP が false の場合に、ランタイムにデータベースに接続するために使用する DLL を示します
string	ランタイム ADO 接続ストリングまたは JDBC URL
dsn	ColdFusion の DSN
driver	ランタイム JDBC ドライバ
username	ランタイムユーザー名
password	ランタイムパスワード
useHTTP	true または false のいずれかの値を含むストリング。デザイン時にリモートドライバ (HTTP 接続) を使用するか、ローカルドライバ (DLL) を使用するかを指定します
includePattern	[ライブデータ] および [ブラウザでプレビュー] で、ページ上のファイルインクルードステートメントを検索するために使用される正規表現
variables	[ライブデータ] および [ブラウザでプレビュー] で使用されるページ変数名とそれに対応する値の配列
catalog	表示されるメタデータを制限するために使用されます (詳細については 55 ページの MMDB.getProcedures() を参照)
schema	表示されるメタデータを制限するために使用されます (詳細については 55 ページの MMDB.getProcedures() を参照)
filename	接続を作成するために使用されたダイアログボックスのファイル名

注意 : これらのプロパティは、*Dreamweaver* が実装する標準プロパティです。デベロッパーは、独自の接続タイプを定義し、新しいプロパティをこの標準セットに追加するか、別のセットのプロパティを指定できます。

MMDB.getConnectionList()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

この関数は、[接続] ダイアログボックスに定義されているすべての接続ストリングのリストを取得します。

引数

なし

戻り値

ストリングの配列。各ストリングは [接続] ダイアログボックスに表示される接続名です。

例

`MMDB.getConnectionList()` を呼び出すと、`["EmpDB", "Test", TestEmp"]` のようなストリングが返されます。

MMDB.getConnectionName()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

指定された接続ストリングに対応する接続名を取得します。この関数は、ユーザーインターフェイス (UI) でページ上のデータから接続名を再選択する場合に便利です。

接続ストリングが 2 つのドライバを参照している場合は、接続ストリングと共に、取得する接続名に対応するドライバを指定することができます。たとえば、2 つの接続があるとします。

- 接続 1 のプロパティは次のとおりです。

```
ConnectionString="jdbc:inetdae:velcro-qa-5:1433?database=pubs"

DriverName="com.inet.tds.TdsDriver"
```

- 接続 2 のプロパティは次のとおりです。

```
ConnectionString="jdbc:inetdae:velcro-qa-5:1433?database=pubs"

DriverName="com.inet.tds.TdsDriver2"
```

接続 1 と接続 2 の接続ストリングは同じです。接続 2 は、`TdsDriver` ドライバのより新しいバージョンに接続しています。取得する接続名を完全に限定するには、この関数にドライバ名を渡す必要があります。

引数

`connString`、*`{driverName}`*

- *`connString`* 引数は、接続名を取得する接続ストリングです。
- オプションの *`driverName`* 引数により、*`connString`* 引数をさらに限定します。

戻り値

接続ストリングに対応する接続名ストリング。

例

以下のコードを実行すると、ストリング `"EmpDB"` が返されます。

```
var connectionName = MMDB.getConnectionName ¬
("dsn=EmpDB;uid=;pwd=");
```

MMDB.getConnectionString()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

この関数は、指定された接続名に関連付けられている接続ストリングを取得します。

引数

connName

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。

戻り値

指定した接続に対応する接続ストリング。

例

コード `var connectionString = MMDB.getConnectionString ("EmpDB")` を実行すると、ADO 接続と JDBC 接続とで異なるストリングが返されます。

- ADO 接続の場合は、次のようなストリングが返されます。

```
"dsn=EmpDB;uid=;pwd=";
```

- JDBC 接続の場合は、次のようなストリングが返されます。

```
"jdbc:inetdae:192.168.64.49:1433?database=pubs&user=JoeUser&¬
password=joesSecret"
```

MMDB.getDriverName()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

指定された接続に関連付けられているドライバ名を取得します。JDBC 接続だけがドライバ名を使用します。

引数

connName

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。

戻り値

ドライバ名を含むストリング。

例

ステートメント `MMDB.getDriverName ("EmpDB");` を実行すると、次のようなストリングが返されます。

```
"jdbc/oracle/driver/JdbcOracle"
```

MMDB.getDriverUrlTemplateList() (非推奨)

対応バージョン

Dreamweaver UltraDev 4 (英語版)、Dreamweaver MX では非推奨

注意 : *Dreamweaver UltraDev 4* の場合、*JDBC* ドライバのリストは "*Configuration/Connections*" フォルダにある "*connections.xml*" ファイルに格納されます。各ドライバには、関連付けられた *URL* テンプレートがあります。この関数は、*JDBC* ドライバのリストを返します。

Dreamweaver MX 以降のバージョンの場合、これらのドライバと URL テンプレートは、JDBC ダイアログボックスにハードコーディングされています。さらに、この関数は未定義の関数エラーを排除するための空の関数定義です。次の例は、JDBC ドライバと URL テンプレートのハードコーディング方法を示しています。

```
var DEFAULT_DRIVER = "COM.ibm.db2.jdbc.app.DB2Driver";
var DEFAULT_TEMPLATE = "jdbc:db2:[database name]";
```

Dreamweaver には、ドライバと URL テンプレートの各ペアに 1 つずつダイアログボックスがあります。

つまり、Dreamweaver UltraDev 4 のデベロッパーは、新しいエントリを XML に追加する必要があり、Dreamweaver MX 以降のバージョンのデベロッパーは、新しいダイアログボックスを実装する必要があります。

説明

JDBC ドライバと、対応する URL テンプレートを取得します。

引数

なし

戻り値

ユーザーのシステム上で検出された JDBC ドライバ、およびそれぞれに URL テンプレートが指定されている場合はそれを含む配列。配列には、`Driver1`、`UrlTemplate1`、`Driver2`、`UrlTemplate2` などの偶数個のエレメントが含まれています。

MMDB.getLocalDsnList()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

ユーザーのシステム上で定義された ODBC DSN を取得します。

引数

なし

戻り値

ユーザーのシステムで定義されている ODBC DSN を含む配列。

MMDB.getPassword()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

指定された接続に使用されているパスワードを取得します。

引数

connName

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。

戻り値

接続名に関連付けられているパスワードストリング。

例

ステートメント `MMDB.getPassword ("EmpDB");` を実行すると、`"joessecret"` などのストリングが返されます。

MMDB.getRDSPassword()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

RDS (Remote Development Services: リモート開発サービス) パスワードを取得します。RDS パスワードは ColdFusion 接続に使用されます。

引数

なし

戻り値

RDS パスワードを含むストリング。

MMDB.getRDSUserName()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

RDS ユーザー名を取得します。RDS ユーザー名は ColdFusion 接続に使用されます。

引数

なし

戻り値

RDS ユーザー名を含むストリング。

MMDB.getRemoteDsnList()

対応バージョン

Dreamweaver UltraDeveloper 4、Dreamweaver MX で機能強化。

説明

サイトサーバーから ODBC DSN を取得します。現在のサイトのサーバーモデルが ColdFusion の場合は、`getRDSUserName()` および `getRDSPassword()` 関数が使用されます。この関数は、`MMDB.getRemoteDsnList()` が生成するリモート接続 URL に追加される URL パラメータストリングをデベロッパーが指定するためのオプションを提供します。デベロッパーがパラメータストリングを指定すると、この関数によってそれが HTTP 接続スクリプトに渡されます。

引数

`{urlParams}`

- *`urlParams`* 引数 (オプション) は、アンパサンド (&) 文字で区切られた *`name=value`* 式のリストを含むストリングです。値を引用符で囲まないでください。値 "`Hello World`" におけるスペースなど、いくつかの文字はエンコードする必要があります。次の例は、`MMDB.getRemoteDsnList()` に渡すことができる有効なサンプル引数です。
  a=1&b=Hello%20World

戻り値

現在のサイトに対してサーバーで定義されている ODBC DSN を含む配列。

MMDB.getRuntimeConnectionType()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

指定された接続名のランタイムの接続タイプを返します。

引数

connName

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。

戻り値

接続タイプに対応するストリング。この関数から返される値は、"ADO"、"ADODSN"、"JDBC"、または "CFDSN" のいずれかです。

例

次のコードを実行すると、ADO 接続の場合はストリング "ADO" が返されます。

```
var connectionType = MMDB.getRuntimeConnectionType ("EmpDB")
```

MMDB.getUserName()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

指定された接続のユーザー名を取得します。

引数

connName

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。

戻り値

接続名に関連付けられているユーザー名ストリング。

例

ステートメント `MMDB.getUserName ("EmpDB");` を実行すると、"amit" などのストリングが返されます。

MMDB.hasConnectionWithName()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

指定された名前の接続が存在しているかどうかを判別します。

引数

name

- *name* 引数は接続名です。

戻り値

指定した名前の接続が存在することを示す true、または存在しないことを示す false のいずれかのブール値を返します。

MMDB.needToPromptForRdsInfo()

対応バージョン

Dreamweaver MX

説明

Dreamweaver で [RDS ログイン情報] ダイアログボックスを表示するかどうかを決定します。

引数

bForce

- *bForce* 引数はブール値です。true は、前に [RDS ログイン] ダイアログボックスをキャンセルしたユーザーにも RDS ログイン情報の入力を求めることを示します。

戻り値

ユーザーが RDS ログイン情報の入力を求められることを示す true、または求められないことを示す false のいずれかのブール値。

MMDB.needToRefreshColdFusionDsnList()

対応バージョン

Dreamweaver MX

説明

[接続] ダイアログボックスに対して、キャッシュを空にし、ユーザーが次にリストを要求したときにアプリケーションサーバーから ColdFusion データソースリストを取得するように指示します。

引数

なし

戻り値

なし

MMDB.popupConnection()

対応バージョン

Dreamweaver MX

説明

接続ダイアログボックスを表示します。この関数には、以下の 3 つのシグネチャがあります。

- 引数リストが *dialogFileName* (ストリング) のみで構成されている場合は、popupConnection() 関数によって Dreamweaver に [接続] ダイアログボックスが開き、新しい接続を定義できます。

- 引数リストが *connRec* (接続参照) のみで構成されている場合は、popupConnection() 関数によって Dreamweaver に [接続] ダイアログボックスが編集モードで表示され、指定された接続を編集できます。このモードでは、名前のテキストフィールドは淡色表示されます。
- 引数リストが *connRec* とブール値の *bDuplicate* で構成されている場合は、popupConnection() 関数によって Dreamweaver に [接続] ダイアログボックスが複製モードで表示されます。このモードでは、名前のテキストフィールドが空白になり、その他のプロパティは接続を複製するためにコピーされます。

引数

dialogFileName
または
connRec
または
connrec、*bDuplicate*

- *dialogFileName* 引数は、"Configuration/Connections/*server-model*" フォルダにある HTML ファイルの名前を含むストリングです。この HTML ファイルは、接続を作成するダイアログボックスを定義します。このファイルでは、findConnection()、inspectConnection()、applyConnection() という 3 つの JavaScript API 関数を実装する必要があります。通常、これらの関数を実装する JavaScript ファイルを作成し、そのファイルを HTML ファイルにインクルードします。接続の作成の詳細については、63 ページのデータベース接続 API を参照してください。
- *connRec* 引数は、既存の接続オブジェクトへの参照です。
- *bDuplicate* 引数はブール値です。

戻り値

なし定義されたダイアログボックスが表示されます。

MMDB.setRDSPassword()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

RDS パスワードを設定します。

引数

password

- *password* 引数は、RDS パスワードを含むストリングです。

戻り値

なし

MMDB.setRDSUserName()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

RDS ユーザー名を設定します。

引数

username

- *username* 引数は、有効な RDS ユーザー名です。

戻り値
なし

MMDB.showColdFusionAdmin()

対応バージョン
Dreamweaver MX

説明
[ColdFusion Administrator] ダイアログボックスを表示します。

引数
なし

戻り値
なし [ColdFusion Administrator] ダイアログボックスが表示されます。

MMDB.showConnectionMgrDialog()

対応バージョン
Dreamweaver UltraDev 1 (英語版)

説明
[接続] ダイアログボックスを表示します。

引数
なし

戻り値
なし [接続] ダイアログボックスが表示されます。

MMDB.showOdbcDialog()

対応バージョン
Dreamweaver UltraDev 4 (英語版。Windows のみ)

説明
[システム ODBC アドミニストレーション] ダイアログボックスまたは [ODBC データソースアドミニストレータ] ダイアログボックスを表示します。

引数
なし

戻り値
なし [システム ODBC アドミニストレーション] ダイアログボックスまたは [ODBC データソースアドミニストレータ] ダイアログボックスが表示されます。

MMDB.showRdsUserDialog()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

RDS ユーザー名とパスワードのダイアログボックスを表示します。

引数

username、*password*

- *username* 引数は初期ユーザー名です。
- *password* 引数は、初期パスワードです。

戻り値

username および password プロパティの新しい値を含むオブジェクト。いずれのプロパティも定義されていない場合は、ユーザーがダイアログボックスをキャンセルしたことを示します。

MMDB.showRestrictDialog()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

[制限] ダイアログボックスを表示します。

引数

catalog、*schema*

- *catalog* 引数は初期カタログ値です。
- *schema* は初期スキーマ値です。

戻り値

catalog および schema プロパティの新しい値を含むオブジェクト。いずれのプロパティも定義されていない場合は、ユーザーがダイアログボックスをキャンセルしたことを示します。

MMDB.testConnection()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

接続設定をテストします。結果を記述するモーダルダイアログボックスが表示されます。

引数

serverPropertiesArray

この関数では、1 つの引数が使用されます。これは、現在のサーバーモデルに適切な以下のリストの値を含む配列オブジェクトです。テストする接続に該当しないプロパティは、空白 ("") に設定してください。

- *type* 引数は、*useHTTP* が `false` 値の場合に、デザイン時に接続設定をテストするためにデータベースへの接続に使用する DLL を示します。
- *string* 引数は、ADO 接続ストリングまたは JDBC URL です。
- *dsn* 引数はデータソース名です。
- *driver* 引数は JDBC ドライバです。
- *username* 引数はユーザー名です。
- *password* 引数はパスワードです。
- *useHTTP* 引数はブール値です。`true` は、Dreamweaver がデザイン時に HTTP 接続を使用することを指定します。それ以外の場合は Dreamweaver は DLL を使用します。

戻り値

接続テストが正常に終了した場合は `true` を、それ以外の場合は `false` を示すブール値。

データベースアクセス関数

データベースアクセス関数を使用すると、データベースに対してクエリーを実行できます。データベース接続を管理する関数については、41 ページのデータベース接続関数を参照してください。

以下では、これらの関数で共通して使用される引数の一部を説明します。

- ほとんどのデータベースアクセス関数は、引数として接続名を使用します。有効な接続名のリストは、[接続] ダイアログボックスで参照できます。または、`MMDB.getConnectionList()` 関数を使ってすべての接続名のリストをプログラム的に取得することもできます。
- ストアードプロシージャでは、多くの場合パラメータを必要とします。データベース関数にパラメータ値を指定する方法は 2 種類あります。その 1 つは、パラメータ値の配列 (*paramValuesArray*) を指定する方法です。パラメータ値のみを指定する場合は、ストアードプロシージャがパラメータを必要とする順序で値を指定する必要があります。もう 1 つは、パラメータ名の配列 (*paramNameArray*) を提供してパラメータ値を指定する方法です。ストアードプロシージャのパラメータは、`MMDB.getSPParamsAsString()` 関数で取得できます。パラメータ名を指定する場合、*paramValuesArray* に指定する値の順序は、*paramNameArray* に指定するパラメータ名の順序に合わせる必要があります。

MMDB.getColumnAndTypeList()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

実行された SQL `SELECT` ステートメントから列とそのデータタイプのリストを取得します。

引数

connName、*statement*

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。
- *statement* 引数は、実行する SQL `SELECT` ステートメントです。

戻り値

`SELECT` ステートメントと一致した列とそのデータタイプのリストを表すストリングの配列。SQL ステートメントが無効であるか、または接続を確立できなかった場合はエラーが返されます。

例

コード `var columnArray = MMDB.getColumnAndTypeList("EmpDB","Select * from Employees")` を実行すると、次のストリングの配列が返されます。

```
columnArray[0] = "EmpName"
columnArray[1] = "varchar"
columnArray[2] = "EmpFirstName"
columnArray[3] = "varchar"
columnArray[4] = "Age"
columnArray[5] = "integer"
```

MMDB.getColumnList()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

実行された SQL `SELECT` ステートメントから列のリストを取得します。

引数

connName、*statement*

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。
- *statement* 引数は、実行する SQL `SELECT` ステートメントです。

戻り値

`SELECT` ステートメントと一致した列のリストを表すストリングの配列。SQL ステートメントが無効であるか、または接続を確立できなかった場合はエラーが返されます。

例

コード `var columnArray = MMDB.getColumnList("EmpDB","Select * from Employees")` を実行すると、次のストリングの配列が返されます。

```
columnArray[0] = "EmpName"
columnArray[1] = "EmpFirstName"
columnArray[2] = "Age"
```

MMDB.getColumns()

対応バージョン

Dreamweaver MX、Dreamweaver MX 2004 (Dreamweaver MX 2004 では引数を更新)

説明

指定されたテーブル内の列を記述するオブジェクトの配列を返します。

引数

connName、*tableName*

- *connName* 引数は接続名です。この値は、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用されるストリングを含む接続を識別します。
- *tableName* 引数には、クエリーを実行するテーブルを指定します。

戻り値

オブジェクトの配列 (各列に 1 オブジェクト)。各オブジェクトは、関連付けられた列に対して以下の 3 つのプロパティを定義します。

プロパティ名	説明
name	列の名前 (たとえば price)
datatype	列のデータタイプ (たとえば small money)
definedsize	列の定義したサイズ (たとえば 8)
nullable	列に null 値を格納できるかどうかの指定

例

次の例では、MMDB.getColumns() を使用してツールヒントテキストの値を設定します。

```
var columnNameObjs = MMDB.getColumns(connName,tableName);
var databaseType = MMDB.getDatabaseType(connName);

    for (i = 0; i < columnNameObjs.length; i++)
    {
        var columnObj = columnNameObjs[i];
        var columnName = columnObj.name;
        var typename = columnObj.datatype;
        if (dwscripts.isNumber(typename))
        {
            // typename が既に数値である場合
            typename = dwscripts.getDBColumnTypeAsString(typename, databaseType);
        }

        var tooltiptext = typename;
    }
```

MMDB.getColumnsOfTable()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

この関数は、指定されたテーブル内のすべての列のリストを取得します。

引数

connName、*tableName*

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。
- *tableName* 引数は *connName* 引数で指定されたデータベースのテーブルの名前です。

戻り値

ストリングの配列。各ストリングはテーブル内の列名です。

例

ステートメント MMDB.getColumnsOfTable ("EmpDB","Employees"); を実行すると、次のストリングが返されます。

```
["EmpID", "FirstName", "LastName"]
```

MMDB.getPrimaryKeys()

対応バージョン

Dreamweaver MX

説明

指定されたテーブルのプライマリキーを構成する列の名前を返します。プライマリキーは、データベース行の固有の識別子として機能し、少なくとも 1 つの列で構成されます。

引数

connName、*tableName*

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。
- *tableName* 引数は、テーブルのプライマリキーを構成する列のセットを取得するテーブルの名前です。

戻り値

ストリングの配列。配列には、プライマリキーを構成する列ごとに 1 つのストリングが含まれます。

例

次の例では、指定されたテーブルのプライマリキーを返します。

```
var connName      = componentRec.parent.parent.parent.name;
var tableName     = componentRec.name;
var primaryKeys   = MMDB.getPrimaryKeys(connName,tableName);
```

MMDB.getProcedures()

対応バージョン

Dreamweaver MX

説明

指定された接続と関連付けられているプロシージャオブジェクトの配列を返します。

引数

connName

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。

戻り値

プロシージャオブジェクトの配列。各プロシージャオブジェクトには以下の 3 つのプロパティのセットがあります。

プロパティ名	説明
schema[a]	オブジェクトに関連付けられているスキーマの名前。 このプロパティは、getProcedures() 関数がアクセスする SQL データベースのストアードプロシージャと関連付けられたユーザーを識別します。この関数がアクセスするデータベースは、接続の種類によって異なります。 • ODBC 接続の場合は、ODBC データソースがデータベースを定義します。DSN は、getProcedures() 関数に渡した接続オブジェクト (*connName*) の dsn プロパティによって指定されます。 • OLE DB 接続の場合は、接続ストリングがデータベースを指定します。
catalog	オブジェクトに関連付けられているカタログの名前 (所有者修飾子)。 catalog プロパティの値は、OLE DB ドライバの属性によって定義されます。このドライバの属性は、OLE DB 接続ストリングがデータベースを指定していない場合に使用するデフォルトの user.database プロパティを定義します。
procedure	プロシージャの名前。

a. レコードセットが変更されるたびに、Dreamweaver はデータベースに接続し、そのすべてのテーブルを取得します。データベースに多数のテーブルが含まれている場合、システムによっては、すべてのテーブルを取得するまでに長い時間がかかる場合があります。データベース内にスキーマやカタログがある場合は、それらを使用し、Dreamweaver がデザイン時に取得するデータベース項目の数を制限できます。Dreamweaver でスキーマやカタログを適用するには、データベースアプリケーション内で事前にスキーマまたはカタログを作成しておく必要があります。詳細については、データベースのマニュアルを参照するか、システム管理者に問い合わせてください。

例

以下のコードでは、プロシージャのリストを取得します。

```
var procObjects = MMDB.getProcedures(connectionName);
for (i = 0; i < procObjects.length; i++)
{
    var thisProcedure = procObjects[i]
    thisSchema =Trim(thisProcedure.schema)
    if (thisSchema.length == 0)
    {
    thisSchema = Trim(thisProcedure.catalog)
    }
    if (thisSchema.length > 0)
    {
    thisSchema += "."
    }

    var procName = String(thisSchema + thisProcedure.procedure);
 }
```

MMDB.getSPColumnList()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

この関数は、指定されたストアードプロシージャを呼び出すことで生成される結果セット列のリストを取得します。

引数

connName、*statement*、*paramValuesArray*

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。
- *statement* 引数は、実行後に結果セットを返すストアードプロシージャの名前です。

- *paramValuesArray* 引数は、デザイン時のテストパラメータ値のリストを含む配列です。これらのパラメータ値は、指定したストアードプロシージャで必要とされる順序で指定します。ストアードプロシージャに対するパラメータは、MMDB.getSPParamsAsString() 関数で取得できます。

戻り値

列のリストを表すストリングの配列。SQL ステートメントまたは接続ストリングが無効である場合は、エラーが返されます。

例

以下のコードでは、ストアードプロシージャ getNewEmployeesMakingAtLeast の実行によって生成された結果セット列のリストを返すことができます。

```
var paramValueArray = new Array("2/1/2000", "50000")
var columnArray = MMDB.getSPColumnList("EmpDB", ¬
"getNewEmployeesMakingAtLeast", paramValueArray)
```

以下の値が返されます。

```
columnArray[0] = "EmpID", columnArray[1] = "LastName", ¬
columnArray[2] = "startDate", columnArray[3] = "salary"
```

MMDB.getSPColumnListNamedParams()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

この関数は、指定されたストアードプロシージャを呼び出すことで生成される結果セット列のリストを取得します。

引数

connName、*statement*、*paramNameArray*、*paramValuesArray*

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。
- *statement* 引数は、実行後に結果セットを返すストアードプロシージャの名前です。
- *paramNameArray* 引数は、パラメータ名のリストを含む配列です。ストアードプロシージャのパラメータは、MMDB.getSPParamsAsString() 関数で取得できます。
- *paramValuesArray* 引数は、デザイン時のテストパラメータ値のリストを含む配列です。プロシージャが実行時にパラメータを必要とする場合に指定できます。*paramNameArray* にパラメータ名を指定した場合は、対応するパラメータ値を *paramNameArray* 内のパラメータ名と同じ順序で指定します。*paramNameArray* を指定しなかった場合は、ストアードプロシージャが必要とする順序で値を指定します。

戻り値

列のリストを表すストリングの配列。SQL ステートメントまたは接続ストリングが無効である場合は、エラーが返されます。

例

以下のコードでは、ストアードプロシージャ getNewEmployeesMakingAtLeast の実行によって生成された結果セット列のリストを返すことができます。

```
var paramNameArray = new Array("startDate", "salary")
var paramValueArray = new Array("2/1/2000", "50000")
var columnArray = MMDB.getSPColumnListNamedParams("EmpDB", ¬
"getNewEmployeesMakingAtLeast", paramNameArray, paramValueArray)
```

以下の値が返されます。

```
columnArray[0] = "EmpID", columnArray[1] = "LastName",¬
columnArray[2] = "startDate", columnArray[3] = "salary"
```

MMDB.getSPParameters()

対応バージョン

Dreamweaver MX

説明

指定されたプロシージャのパラメータオブジェクトの配列を返します。

引数

connName、*procName*

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。
- *procName* 引数は、プロシージャの名前です。

戻り値

パラメータオブジェクトの配列。各オブジェクトが以下のプロパティのセットを指定します。

プロパティ名	説明
name	パラメータの名前 (たとえば `@@lolimit`)
datatype	パラメータのデータタイプ (たとえば `smallmoney`)
direction	パラメータの方向。 `1` - このパラメータは入力にのみ使用されます。 `2` - このパラメータは出力にのみ使用されます。この場合、パラメータは参照を使用して渡し、値はメソッドによって指定されます。メソッドから返された後、その値を使用できます。 `3` - このパラメータは入力と出力の両方に使用されます。 `4` - このパラメータは戻り値を保持します。

例

次の例では、指定されたプロシージャへのパラメータオブジェクトを取得し、オブジェクトのプロパティを使用して各オブジェクトに対するツールヒントを作成します。

```
var paramNameObjs = MMDB.getSPParameters(connName,procName);
for (i = 0; i < paramNameObjs.length; i++)
{
    var paramObj = paramNameObjs[i];
    var tooltiptext = paramObj.datatype;
    tooltiptext+=" ";
    tooltiptext+=GetDirString(paramObj.directiontype);
}
```

MMDB.getSPParamsAsString()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

ストアードプロシージャが受け取るパラメータのリストを、カンマで区切ったストリングとして取得します。

引数

connName、*procName*

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。
- *procName* 引数は、ストアードプロシージャの名前です。

戻り値

ストアードプロシージャに必要なパラメータをカンマで区切ったリストのストリング。内容は、セミコロン (;) で区切られたパラメータの名前、方向、およびデータタイプです。

例

コード `MMDB.getSPParamsAsString ("EmpDB","getNewEmployeesMakingAtLeast")` は、たとえば "`startDate;direction:in;datatype:date, salary;direction:in;datatype:integer`" という形式のストリングを返します。

この例では、ストアードプロシージャ `getNewEmployeesMakingAtLeast` に `startDate` および `Salary` の 2 つのパラメータがあります。`startDate` の方向は `in`、データタイプは `date` です。`salary` の方向は `入力`、データタイプは `date` です。

MMDB.getTables()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

指定されたデータベースに定義されているすべてのテーブルのリストを取得します。各テーブルオブジェクトには、`table`、`schema`、および `catalog` の 3 つのプロパティがあります。

引数

connName

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。

戻り値

オブジェクトの配列。各オブジェクトには、`table`、`schema`、および `catalog` の 3 つのプロパティがあります。`Table` はテーブルの名前です。`Schema` はテーブルを含むスキーマの名前です。`Catalog` はテーブルを含むカタログです。

例

ステートメント `MMDB.getTables ("EmpDB");` を実行すると、2 つのオブジェクトから構成される配列が生成されます。最初のオブジェクトのプロパティは、次の例のようになります。

```
object1[table:"Employees", schema:"personnel", catalog:"syscat"]
```

2 番目のオブジェクトのプロパティは、次の例のようになります。

```
object2[table:"Departments", schema:"demo", catalog:"syscat2"]
```

MMDB.getViews()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

この関数は、指定されたデータベースに定義されているすべてのビューのリストを取得します。各ビューオブジェクトには、catalog、schema、および view のプロパティがあります。

引数

connName

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。

戻り値

ビューオブジェクトの配列。各オブジェクトには、catalog、schema、および view の 3 つのプロパティがあります。catalog または schema は、接続情報の一部として定義されている個々のスキーマ名またはカタログ名に関係するビューの数を制限またはフィルタリングするために使用されます。

例

次の例では、指定された接続値 CONN_LIST.getValue() のビューが返されます。

```
var viewObjects = MMDB.getViews(CONN_LIST.getValue())
for (i = 0; i < viewObjects.length; i++)
{
    thisView = viewObjects[i]
    thisSchema = Trim(thisView.schema)
    if (thisSchema.length == 0)
    {
        thisSchema = Trim(thisView.catalog)
    }
    if (thisSchema.length > 0)
    {
        thisSchema += "."
    }
    views.push(String(thisSchema + thisView.view))
}
```

MMDB.showResultset()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

この関数は、指定された SQL ステートメントの実行結果を示すダイアログボックスを表示します。このダイアログボックスには表形式のグリッドが表示され、表のヘッダーには結果セットについて説明する列情報が表示されます。接続ストリングまたは SQL ステートメントが無効である場合は、エラーが表示されます。この関数は、SQL ステートメントを検証します。

引数

connName、*SQLstatement*

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。
- *SQLstatement* 引数には、SQL SELECT ステートメントを指定します。

戻り値

なし SQL ステートメントまたは接続ストリングが無効である場合は、エラーが返されます。

例

次のコードは、実行された SQL ステートメントの結果を表示します。

```
MMDB.showResultset("EmpDB","Select EmpName,EmpFirstName,Age ¬
from Employees")
```

MMDB.showSPResultset()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

この関数は、指定されたストアードプロシージャの実行結果を示すダイアログボックスを表示します。このダイアログボックスには表形式のグリッドが表示され、表のヘッダーには結果セットについて説明する列情報が表示されます。接続ストリングまたはストアードプロシージャが無効である場合は、エラーが表示されます。この関数は、ストアードプロシージャを検証します。

引数

connName、*procName*、*paramValuesArray*

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。
- *procName* 引数は、実行するストアードプロシージャの名前です。
- *paramValuesArrayargument* は、デザイン時のテストパラメータ値のリストを含む配列です。これらのパラメータ値は、指定したストアードプロシージャで必要とされる順序で指定します。ストアードプロシージャのパラメータは、`MMDB.getSPParamsAsString()` 関数で取得できます。

戻り値

SQL ステートメントまたは接続ストリングが無効である場合は、エラーが返されます。それ以外の場合、戻り値はありません。

例

次のコードは、実行されたストアードプロシージャの結果を表示します。

```
var paramValueArray = new Array("2/1/2000", "50000")
MMDB.showSPResultset("EmpDB", "getNewEmployeesMakingAtLeast", ¬
paramValueArray)
```

MMDB.showSPResultsetNamedParams()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

この関数は、指定されたストアードプロシージャの結果セットを示すダイアログボックスを表示します。このダイアログボックスには表形式のグリッドが表示され、表のヘッダーには結果セットについて説明する列情報が表示されます。接続ストリングまたはストアードプロシージャが無効である場合は、エラーが表示されます。この関数は、ストアードプロシージャを検証します。この関数は、`MMDB.showSPResultset()` 関数と違って、ストアードプロシージャが必要とする順序ではなく、パラメータ値を名前で指定できます。

引数

connName、*procName*、*paramNameArray*、*paramValuesArray*

- *connName* 引数は、[接続] ダイアログボックスで指定された接続名です。これは、ライブデータソースとのデータベース接続を確立するために Dreamweaver によって使用される接続ストリングを識別します。
- *procName* 引数は、実行後に結果セットを返すストアードプロシージャの名前です。
- *paramNameArray* 引数は、パラメータ名のリストを含む配列です。ストアードプロシージャのパラメータは、MMDB.getSPParamsAsString() 関数で取得できます。
- *paramValuesArray* 引数は、デザイン時のテストパラメータ値のリストを含む配列です。

戻り値

SQL ステートメントまたは接続ストリングが無効である場合は、エラーが返されます。それ以外の場合、戻り値はありません。

例

次のコードは、実行されたストアードプロシージャの結果を表示します。

```
var paramNameArray = new Array("startDate", "salary")
var paramValueArray = new Array("2/1/2000", "50000")
MMDB.showSPResultsetNamedParams("EmpDB","getNewEmployees¬
MakingAtLeast", paramNameArray, paramValueArray)
```

第 8 章 : データベース接続 API

デベロッパーは、データベース接続 API を使用して、Adobe® Dreamweaver® CS3 の新規または既存のサーバーモデルに対して、新しい接続タイプと、それに対応するダイアログボックスを作成できます。これにより、ユーザーがサイトを設定してページを作成するときに、デベロッパーが作成した接続タイプを選択して、ユーザー側で新しい接続オブジェクトを作成できます。

ユーザーは、デベロッパーが作成した新しい接続タイプを以下の方法で選択できます。

- [アプリケーション] パネルで、[+] ボタンをクリックして [レコードセット] を選択します。[レコードセット] ダイアログボックスで [接続] リストボックスを展開します。
- [データベース] パネルの [データベース] タブで、[+] ボタンをクリックして [データソース名] を選択します。

新しい接続タイプの開発方法

新しい接続タイプを作成する手順の概略は、以下のとおりです。

1 接続ダイアログボックスのレイアウトを作成します。

接続ダイアログボックスのユーザーインターフェイス (UI) をレイアウトする HTML ファイルを作成します。接続の名前をこのファイルに付けます (myConnection.htm など)。ダイアログボックス作成の詳細については、『Dreamweaver ファーストステップガイド』を参照してください。

この HTML ファイルでは、手順 2 (64 ページの少なくとも以下のエレメントを実装する JavaScript ファイルを作成します。) で定義する JavaScript 実装ファイルを次の例のようにインクルードします。

```
<head>
    <script SRC="../myConnectionImpl.js"></script>
</head>
```

接続ダイアログボックスを定義するこの HTML ファイルを "Configuration/Connections/*server-model*/*platform*" フォルダに格納します (*platform* は Windows または Macintosh)。

たとえば、Windows プラットフォーム上の ASP JavaScript ドキュメントのデフォルトの ADO 接続ダイアログボックスは "ASP_Js/Win" フォルダに格納し、"Connection_ado_conn_string.htm" という名前を付けます。

注意 : ランタイムに、"*ASP_Js/Win*" フォルダにあるダイアログボックスの集合からユーザーが使用可能な接続タイプのリストが動的に構築されます。

"Configuration/ServerModels" フォルダには、それぞれのサーバーモデルを定義する HTML ファイルがあります。それぞれの HTML ファイル内には、サーバーモデルに関連するフォルダの名前を返す `getServerModelFolderName()` 関数があります。次の例は、ASP JavaScript ドキュメントタイプの関数を示しています。

```
function getServerModelFolderName()
{
    return "ASP_JS";
}
```

また、"Configuration/DocumentTypes" フォルダにある "MMDocumentTypes.xml" ファイルで、サーバーモデルとドキュメントタイプの間のマッピングを決定することもできます。

2 少なくとも以下のエレメントを実装する JavaScript ファイルを作成します。

エレメント	説明	例
変数のセット	それぞれの変数が特定の接続プロパティを定義します。	接続タイプ、データソース名など
ボタンのセット	それぞれのボタンが接続ダイアログボックスに表示されます。	[テスト]、[ヘルプ] など ([OK] および [キャンセル] は自動的に組み込まれます)

接続関数	これらの関数を一緒に使用して接続 API を定義します。	`findConnection()`
		`applyConnection()`
		`inspectConnection()`

この実装ファイルにはどのような名前も選択できますが、拡張子として .js を付ける必要があります ("myConnectionImpl.js" など)。この実装ファイルは、ローカルコンピュータ上にもリモートコンピュータ上にも保存できます。実装ファイルは、"Configuration/Connections" フォルダ内の適切なサブフォルダに保存することもできます。

注意 : 手順 *1* (63 ページの接続ダイアログボックスのレイアウトを作成します。) で定義した *HTML* ファイルには、この接続タイプの実装ファイルをインクルードする必要があります。

標準的な "connection_includefile.edml" ファイルで指定されていない接続パラメータを定義する必要がなければ、新しい接続ダイアログボックスを作成するために最低限必要な手順はこの 2 つです。

注意 : ユーザー向けに表示されるダイアログボックスのタイトルは、*HTML* ドキュメントで指定する `title` タグ内に指定します。

次の項に示す関数を使用して、接続ダイアログボックスを作成できます。ユーザー向けのインクルードファイルを生成する呼び出しを実装すると同時に、接続 XML ファイルのサーバーモデルセクション内で接続タイプを登録できます。

新しい接続の作成に関連するデータベース接続 API の詳細については、41 ページのデータベース接続関数を参照してください。

接続 API

新しい接続タイプ (ユーザーとの対話形式のダイアログボックスを含む) を作成するには、`findConnection()`、`inspectConnection()`、および `applyConnection()` の 3 つの関数を実装する必要があります。これらの 3 つの関数を作成したら、新しい接続タイプと関連付けられた JavaScript 実装ファイルにインクルードする必要があります。手順 2 (64 ページの少なくとも以下のエレメントを実装する JavaScript ファイルを作成します。) を参照してください。

`applyConnection()` 関数は、インクルードファイル内の HTML ソースを返します。HTML ソースの例については、67 ページの生成されたインクルードファイルを参照してください。`findConnection()` 関数は、この HTML ソースを取り込み、そのプロパティを抽出します。`findConnection()` を実装し、XML ファイルの検索パターンを使用すると、`applyConnection()` から返される情報を抽出できます。このような実装の例については、以下の 2 つの JavaScript ファイルを参照してください。

- "`connection_ado_conn_string.js`" は "Configuration/Connections/ASP_Js" フォルダにあります。
- "`connection_common.js`" は "Configuration/Connections/Shared" フォルダにあります。

ユーザーがサイトを開くと、Dreamweaver によって、"Connections" フォルダの各ファイルが調べられて開かれ、そのコンテンツが `findConnection()` に渡されます。ファイルのコンテンツが有効な接続の基準に一致すると、`findConnection()` は接続オブジェクトを返します。その後、Dreamweaver によって [データベース] パネルにすべての接続オブジェクトが表示されます。

ユーザーが接続ダイアログボックスを開き、新しい接続を作成するか、既存の接続を複製または編集することを選択すると、Dreamweaver によって `inspectConnection()` 関数が呼び出され、`findConnection()` で作成されたオブジェクトと同じ接続オブジェクトが返されます。この処理によって、接続情報がダイアログボックスに読み込まれます。

ユーザーが接続ダイアログボックスで [OK] をクリックすると、Dreamweaver によって applyConnection() 関数が呼び出されて HTML が構築されます。この HTML は、"Configuration/Connections" フォルダにある接続インクルードファイルに配置されます。applyConnection() 関数は、フィールドの 1 つにエラーがあり、ダイアログボックスを閉じてはいけないことを示す空白のストリングを返します。このインクルードファイルのデフォルトのファイル拡張子は、現在のサーバーモデルに対応するタイプです。

ユーザーが、レコードセットやストアードプロシージャなど、接続を使用するサーバービヘイビアをページに追加すると、接続インクルードファイルをインクルードするステートメントが Dreamweaver によってページに追加されます。

findConnection()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

この関数は、指定した HTML ソースで接続を検出し、接続パラメータを解析するために呼び出されます。このソースファイルのコンテンツが有効な接続の基準に一致する場合、findConnection() は接続オブジェクトを返します。それ以外の場合は、null 値を返します。

引数

htmlSource

htmlSource 引数は、接続用の HTML ソースです。

戻り値

以下の表に示すプロパティの特定の組み合わせに対して値を提供する接続オブジェクト。この関数が値を返すプロパティは、ドキュメントタイプによって異なります。

プロパティ	説明
name	接続の名前
type	useHTTP が false の場合に、ランタイムにデータベースに接続するために使用する DLL
string	ランタイム接続ストリング。ADO の場合は接続パラメータのストリング。JDBC の場合は接続 URL
dsn	ODBC または Cold Fusion ランタイム接続に使用されるデータソース名
driver	ランタイムに使用される JDBC ドライバの名前
username	ランタイム接続に使用されるユーザー名
password	ランタイム接続に使用されるパスワード
designtimeString	デザイン時接続ストリング (string を参照)
designtimeDsn	デザイン時データソース名 (dsn を参照)
designtimeDriver	デザイン時に使用される JDBC ドライバの名前
designtimeUsername	デザイン時接続に使用されるユーザー名
designtimePassword	デザイン時接続に使用されるパスワード
designtimeType	デザイン時接続タイプ
usesDesigntimeInfo	false の場合はデザイン時にランタイムプロパティを使用し、それ以外の場合はデザイン時プロパティを使用
useHTTP	true または false を含むストリング。true の場合はデザイン時に HTTP 接続を使用し、false は DLL を使用

プロパティ	説明
includePattern	[ライブデータ] および [ブラウザでプレビュー] で、ページ上のファイルインクルードステートメントを検索するために使用される正規表現
variables	対応する値に設定された各ページ変数のプロパティを含むオブジェクト。このオブジェクトは、[ライブデータ] および [ブラウザでプレビュー] で使用されます
catalog	表示されるメタデータの量を制限するデータベース識別子を含むストリング
schema	表示されるメタデータの量を制限するデータベース識別子を含むストリング
filename	接続の作成に使用されるダイアログボックスの名前

接続が htmlSource にない場合は、null 値が返されます。

注意 : デベロッパーは、カスタムプロパティ (たとえばメタデータ) を *HTML* ソースに追加できます。*HTML* ソースは *applyConnection()* によって標準プロパティと共に返されます。

inspectConnection()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

ユーザーが既存の接続を編集するとき、Dreamweaver によってこの関数が呼び出され、接続を定義するためのダイアログボックスデータが初期化されます。この処理によって、適切な接続情報がダイアログボックスに読み込まれます。

引数

parameters

parameters 引数は、findConnection() 関数が返すのと同じオブジェクトです。

戻り値

なし

applyConnection()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

ユーザーが接続ダイアログボックスで [OK] をクリックすると、Dreamweaver によってこの関数が呼び出されます。applyConnection() 関数は、接続用の HTML ソースを生成します。この HTML は、Dreamweaver によって "Configuration/Connections/*connection-name.ext*" インクルードファイルに書き込まれます。ここで *connection-name* は接続の名前 (63 ページの接続ダイアログボックスのレイアウトを作成します。を参照) で、.ext はサーバーモデルと関連付けられたデフォルトの拡張子です。

引数

なし

戻り値

接続用の HTML ソース。Dreamweaver によって接続ダイアログボックスも閉じられます。フィールド検証エラーが発生すると、applyConnection() がエラーメッセージを表示し、ダイアログボックスを閉じてはいけないことを示す空白のストリングを返します。

生成されたインクルードファイル

applyConnection() が生成するインクルードファイルは、接続のすべてのプロパティを宣言します。インクルードファイルの名前は接続名で、現在のサイトに関連付けられたサーバーモデルに対して定義されたファイル拡張子が付けられます。

注意 : 接続は共有されているため、*allowMultiple 値*を *false* に設定します。これにより、接続ファイルがドキュメントに一度だけインクルードされ、サーバースクリプトは他のサーバービヘイビアによって使用される場合、そのままページに残ります。

以下の項では、さまざまなデフォルトのサーバーモデルに対して applyConnection() が生成するサンプルインクルードファイルを示しています。

注意 : 新しい接続インクルードファイル形式を作成するには、新しい *EDML* マッピングファイルを定義する必要があります。これは、68 ページの接続タイプの定義ファイルに示す "*connection_includefile.edml*" に似た形式にする必要があります。

ASP JavaScript

ASP および JavaScript インクルードファイルは MyConnection1.asp の形式の名前にする必要があります。ここで MyConnection1 は接続の名前です。次のサンプルは、ADO 接続ストリングのインクルードファイルです。

```
<%
    // Filename="Connection_ado_conn_string.htm"
    // Type="ADO"
    // HTTP="true"
    // Catalog=""
    // Schema=""
    var MM_MyConnection1_STRING = "dsn=pubs";
%>
```

次の例のように、サーバービヘイビアファイルは、相対ファイルインクルードステートメントを使用してこの接続をインクルードします。

```
<!--#include file="../Connections/MyConnection1.asp"-->
```

ColdFusion

UltraDeveloper 4 ColdFusion を使用する場合、Dreamweaver は ColdFusion インクルードファイルに依存してデータソースのリストを取得します。

注意 : 通常の *Dreamweaver ColdFusion* の場合、*Dreamweaver* は、インクルードファイルを無視し、代わりに *RDS* を利用して *ColdFusion* からデータソースのリストを取得します。

UltraDeveloper 4 ColdFusion インクルードファイルは、MyConnection1.cfm の形式の名前にする必要があります。ここで MyConnection1 は接続の名前です。次の例では、products テーブルへの ColdFusion 接続のためのインクルードファイルを示します。

```
<!-- FileName="Connection_cf_dsn.htm" "dsn=products" -->
<!-- Type="ADO" -->
<!-- Catalog="" -->
<!-- Schema="" -->
<!-- HTTP="false" -->
<CFSET MM_MyConnection1_DSN       = "products">
<CFSET MM_MyConnection1_USERNAME  = "">
<CFSET MM_Product_USERNAME        = "">
<CFSET MM_MyConnection1_PASSWORD  = "">
```

次の例に示すように、サーバービヘイビアファイルは、cfinclude ステートメントを使用してこの接続をインクルードします。

```
<cfinclude template="Connections/MyConnection1.cfm">
```

JSP

JSP インクルードファイルは、MyConnection1.jsp の形式の名前にする必要があります。ここで MyConnection1 は接続の名前です。次の例では、データベースへの JDBC 接続用のインクルードファイルを示します。

```
<%
    // Filename="Connection_jdbc_conn1.htm"
    // Type="JDBC"
    // HTTP="false"
    // Catalog=""
    // Schema=""
    String MM_MyConnection1_DRIVER    = "com.inet.tds.TdsDriver";
    String MM_MyConnection1_USERNAME  = "testadmin";
    String MM_MyConnection1_PASSWORD  = "velcro";
    String MM_MyConnection1_URL       = "jdbc:server:test-3:1433?database=pubs";
%>
```

次の例のように、サーバービヘイビアファイルは、相対ファイルインクルードステートメントを使用してこの接続をインクルードします。

```
<%@ include file="Connections/MyConnection1.jsp" %>
```

接続タイプの定義ファイル

サーバーモデルごとに、接続タイプを定義し、インクルードファイルで定義されたプロパティを Dreamweaver インターフェイスのエレメントにマッピングする "connection_includefile.edml" ファイルがあります。

Dreamweaver には、次の表に示す 7 つのデフォルトの定義ファイル (定義済みサーバーモデルごとに 1 つ) が用意されています。

サーバーモデル	**"Configuration/Connections" フォルダ内のサブフォルダ。**
ASP JavaScript	ASP_Js
ASP.NET CSharp	ASP.NET_Csharp
ASP.NET VBScript	ASP.NET_VB
ASP VBScript	ASP_Vbs
ColdFusion	ColdFusion
JavaServer ページ	JSP
PHP MySql	PHP_MySql

Dreamweaver は、`quickSearch` および `searchPattern` パラメータを使用して接続ブロックを認識し、`insertText` パラメータを使用して接続ブロックを作成します。EDML のタグと属性、および正規表現を使った検索パターンの詳細については、『Dreamweaver 拡張ガイド』の「サーバービヘイビア」を参照してください。

注意 : インクルードファイルの形式を変更、または新しいサーバーモデルのインクルードファイルを定義する場合、接続パラメータを *Dreamweaver UI*、[ライブデータ]、および [ブラウザでプレビュー] にマッピングする必要があります。次に示す *EDML* ファイルのサンプルは、デフォルトの *ASP JS* サーバーモデルに関連付けられており、ページをサーバーに送信する前にすべての接続ページ変数をそれぞれのライブ値にマッピングします。*EDML* および正規表現を使った検索パターンの詳細については、『*Dreamweaver* 拡張ガイド』の「サーバービヘイビア」を参照してください。

```
<participant name="connection_includefile" version="5.0">
    <quickSearch>
        <![CDATA[// HTTP=]]></quickSearch>
    <insertText location="">
<![CDATA[<%
// FileName="@@filename@@"
// Type="@@type@@" @@designtimeString@@
// DesigntimeType="@@designtimeType@@"
// HTTP="@@http@@"
// Catalog="@@catalog@@"
// Schema="@@schema@@"
var MM_@@cname@@_STRING = @@string@@
%>
]]>
    </insertText>
    <searchPatterns whereToSearch="directive">
        <searchPattern paramNames="filename">
            <![CDATA[/\/\/\s*FileName="([^"]*)"/]]></searchPattern>
        <searchPattern paramNames="type,designtimeString">
            <![CDATA[/\/\/\s+Type="(\w*)"([^\r\n]*)/]]></searchPattern>
        <searchPattern paramNames="designtimeType" isOptional="true">
            <![CDATA[/\/\/\s*DesigntimeType="(\w*)"/]]></searchPattern>
        <searchPattern paramNames="http">
            <![CDATA[/\/\/\s*HTTP="(\w+)"/]]></searchPattern>
        <searchPattern paramNames="catalog">
            <![CDATA[/\/\/\s*Catalog="(\w*)"/]]></searchPattern>
        <searchPattern paramNames="schema">
            <![CDATA[/\/\/\s*Schema="(\w*)"/]]></searchPattern>
        <searchPattern paramNames="cname,string">
            <![CDATA[/var\s+MM_(\w*)_STRING\s*=\s*([^\r\n]+)/]]></searchPattern>
    </searchPatterns>
</participant>
```

EDML ファイルのトークン (この例の `@@filename@@` など) によって、インクルードファイルの値が接続オブジェクトのプロパティにマッピングされます。接続オブジェクトのプロパティは、JavaScript 実装ファイルで設定します。

Dreamweaver にデフォルトで用意されている接続ダイアログボックスでは、"connection_includefile.edml" マッピングファイルが使用されます。Dreamweaver 側でこのファイルを認識できるようにするには、次の例のように JavaScript 実装ファイルでその名前を設定します。

```
var PARTICIPANT_FILE = "connection_includefile";
```

接続タイプをカスタマイズするときは、カスタムダイアログボックスで任意のマッピングファイルを使用できます。マッピングファイルを作成する場合、EDML ファイルに connection_includefile 以外の名前を使用できます。異なる名前を使用する場合は、`PARTICIPANT_FILE` 変数に代入する値を指定するときに、次の例に示すように、JavaScript 実装ファイルでこの名前を使用する必要があります。

```
var PARTICIPANT_FILE = "myConnection_mappingfile";
```

第 9 章 : JavaBeans API

本章では、JavaBeans 用 API について説明します。`MMJB*()` 関数は、JavaBean をサポートするために Java の内部で呼び出しを実行する JavaScript のフックです。これらの関数は、Dreamweaver のユーザーインターフェイス (UI) に表示されるクラス名、メソッド、プロパティ、およびイベントを JavaBean から取得します。これらの JavaScript 関数を使用して Adobe® Dreamweaver® CS3 から JavaBeans へのアクセスを実現するには、JavaBeans が "Configuration/Classes" フォルダに配置されている必要があります。

注意 : 本章で説明する関数の引数には、*`packageName.className`* という引数が含まれている場合があります。これは、単一の値を表すためです。

JavaBeans API

`MMJB` オブジェクトのメソッドには、以下の関数があります。

MMJB.getClasses()

対応バージョン

Dreamweaver UltraDeveloper 4

説明

"Configuration/Classes" フォルダからすべての JavaBeans クラス名を読み取ります。

引数

なし

戻り値

"Configuration/Classes" フォルダ内にあるクラス名を表すストリングの配列。エラーの場合は空白の配列が返されます。

MMJB.getClassesFromPackage()

対応バージョン

Dreamweaver UltraDeveloper 4

説明

パッケージからすべての JavaBeans クラスを読み取ります。

引数

`packageName.pathName`

- *`packageName.pathName`* 引数は、パッケージへのパスです。JAR または ZIP の Java アーカイブを指定する必要があります。たとえば、`C:¥jdbcdrivers¥Una2000_Enterprise.zip` などを指定します。

戻り値

特定の JAR または ZIP の Java アーカイブに含まれているクラス名を表すストリングの配列。エラーの場合は、空白の配列が返されます。

MMJB.getErrorMessage()

対応バージョン

Dreamweaver UltraDeveloper 4

説明

MMJB インターフェイスの使用中に Dreamweaver から返された最後のエラーメッセージを取得します。

引数

なし

戻り値

最後に発生したエラーに対する Dreamweaver のメッセージを表すストリング。

MMJB.getEvents()

対応バージョン

Dreamweaver UltraDeveloper 4、Dreamweaver MX で機能強化。

説明

JavaBeans クラスを内部的に呼び出して、そのイベントを返します。

引数

packageName.className、*{packagePath}*

- *packageName.className* 引数は、クラスの名前です。このクラスは、JAR または ZIP の Java アーカイブ内に存在している必要があります。*packagePath* を省略する場合、アーカイブはシステムの `classpath` にあるか、または "Configuration/Classes" フォルダにインストールされたクラスファイルである必要があります。
- *packagePath* 引数は、*className* を含む JAR または ZIP の Java アーカイブの場所を示すオプションのストリングです。

戻り値

className に関連付けられたイベントを表すストリングの配列。エラーの場合は、空白の配列が返されます。

MMJB.getIndexedProperties()

対応バージョン

Dreamweaver UltraDeveloper 4、Dreamweaver MX で機能強化。

説明

JavaBeans クラスを内部的に呼び出して、コレクションと同じように動作するデザインパターンであるインデックスプロパティを返します。

引数

packageName.className、*{packagePath}*

- *packageName.className* 引数は、クラスの名前です。このクラスは、JAR または ZIP の Java アーカイブ内に存在している必要があります。*packagePath* を省略する場合、アーカイブはシステムの `classpath` にあるか、または "Configuration/Classes" フォルダにインストールされたクラスファイルである必要があります。
- *packagePath* 引数 (オプション) は、*className* を含む JAR または ZIP の Java アーカイブの場所を示すストリングです。

戻り値

className に関連付けられたインデックスプロパティを表すストリングの配列。エラーの場合は、空白の配列が返されます。

MMJB.getMethods()

対応バージョン

Dreamweaver UltraDeveloper 4、Dreamweaver MX で機能強化。

説明

JavaBeans クラスを内部的に呼び出して、そのメソッドを返します。

引数

packageName.className、*{packagePath}*

- *packageName.className* 引数は、クラスの名前です。このクラスは、JAR または ZIP の Java アーカイブ内に存在している必要があります。*packagePath* を省略する場合、アーカイブはシステムの `classpath` にあるか、または "Configuration/Classes" フォルダにインストールされたクラスファイルである必要があります。
- *packagePath* 引数は、*className* を含む JAR または ZIP の Java アーカイブの場所を示すオプションのストリングです。

戻り値

className に関連付けられたメソッドを表すストリングの配列。エラーの場合は、空白の配列が返されます。

MMJB.getProperties()

対応バージョン

Dreamweaver UltraDeveloper 4、Dreamweaver MX で機能強化。

説明

JavaBeans クラスを内部的に呼び出して、そのプロパティを返します。

引数

packageName.className、*{packagePath}*

- *packageName.className* 引数は、クラスの名前です。このクラスは、JAR または ZIP の Java アーカイブ内に存在している必要があります。*packagePath* を省略する場合、アーカイブはシステムの `classpath` にあるか、または "Configuration/Classes" フォルダにインストールされたクラスファイルである必要があります。
- *packagePath* 引数は、*className* を含む JAR または ZIP の Java アーカイブの場所を示すオプションのストリングです。

戻り値

className に関連付けられたプロパティを表すストリングの配列。エラーの場合は、空白の配列が返されます。

MMJB.getReadProperties()

対応バージョン

Dreamweaver MX

説明

get アクセサの呼び出しをサポートする JavaBeans の読み取り専用プロパティを取得します。

引数

packageName.className、*{packagePath}*

- *packageName.className* 引数は、クラスの名前です。このクラスは、JAR または ZIP の Java アーカイブ内に存在している必要があります。*packagePath* を省略する場合、アーカイブはシステムの `classpath` にあるか、または "Configuration/Classes" フォルダにインストールされたクラスファイルである必要があります。
- *packagePath* 引数 (オプション) は、*className* を含む JAR または ZIP の Java アーカイブの場所を示すストリングです。

戻り値

className に関連付けられた読み取り専用プロパティを表すストリングの配列。エラーの場合は、空白の配列が返されます。

MMJB.getWriteProperties()

対応バージョン

Dreamweaver MX

説明

set メソッドの呼び出しをサポートする JavaBeans の書き込み専用プロパティを取得します。

引数

packageName.className、*{packagePath}*

- *packageName.className* 引数は、クラスの名前です。このクラスは、JAR または ZIP の Java アーカイブ内に存在している必要があります。*packagePath* を省略する場合、アーカイブはシステムの `classpath` にあるか、または "Configuration/Classes" フォルダにインストールされたクラスファイルである必要があります。
- *packagePath* 引数 (オプション) は、*className* を含む JAR または ZIP の Java アーカイブの場所を示すストリングです。

戻り値

className に関連付けられた書き込み専用プロパティを表すストリングの配列。エラーの場合は、空白の配列が返されます。

第 10 章 : ソースコントロール統合 API

ソースコントロール統合 API により、他のソースコントロールシステム (Sourcesafe や CVS など) の機能を使用して、Adobe® Dreamweaver® CS3 のチェックイン / チェックアウト機能を拡張する共有ライブラリを作成することができます。

Dreamweaver をソースコントロールシステムに統合するには、最低限必要な API 関数のセットをライブラリでサポートする必要があります。また、ライブラリは "Program Files/Adobe/Adobe Dreamweaver CS3/Configuration/SourceControl" フォルダに置く必要があります。

Dreamweaver が起動すると、各ライブラリがロードされます。次に、各 API 関数に対して `GetProcAddress()` が呼び出され、ライブラリでサポートされている機能が判別されます。アドレスが存在しない場合は、ライブラリはその API をサポートしないものと見なされます。アドレスが存在する場合は、その関数のライブラリのバージョンによって機能がサポートされます。Dreamweaver を使用してサイトを定義または編集した後で、[ウェブサーバー SCS] タブを選択すると、"Program Files/Adobe/Adobe Dreamweaver CS3/Configuration/SourceControl" フォルダからロードされた DLL に対応する選択項目が、標準項目と共にタブに表示されます。

カスタムの項目を追加できる [サイト]-[SourceControl] メニューを作成するには、"menus.xml" ファイル内の [サイト] メニューに次のコードを追加します。

```
<menu name="Source Control" id="DWMenu_MainSite_Site_Source¬
Control"><menuitem dynamic name="None"file="Menus/MM/¬
File_SCSItems.htm" id="DWMenu_MainSite_Site_NewFeatures_¬
Default" />
</menu>
```

本章では関数をグループ分けして、それぞれを以下の項で説明します。

Dreamweaver とのソースコントロール統合のしくみ

Dreamweaver でサーバー接続、ファイル転送、またはデザインノートの諸機能を選択すると、対応する API 関数の DLL バージョンが呼び出されます。この API 関数には、`Connect()`、`Disconnect()`、`Get()`、`Put()`、`Checkin()`、`Checkout()`、`Undocheckout()`、および `Synchronize()` があります。要求は DLL によって処理されます。その中には、情報を収集するダイアログボックスや、ユーザーが DLL と交信するためのダイアログボックスを表示する処理も含まれます。DLL によって、情報メッセージやエラーメッセージの表示も行われます。

ソースコントロールシステムが、オプションとしてデザインノートおよびチェックイン / チェックアウトをサポートする場合があります。Dreamweaver を使用して、ソースコントロールシステムでデザインノートを使用可能にするには、[サイトの編集] ダイアログボックスで [デザインノート] タブを選択し、この機能を使用可能にするボックスをオンにします。この処理は、FTP および LAN を使うデザインノートを使用可能にする処理と同じです。ソースコントロールシステムでデザインノートがサポートされていない場合に、ユーザーがこの機能を必要とするときは、Dreamweaver がデザインノートファイル (MNO) をトランスポートして、デザインノートを維持します (FTP および LAN を使用して維持する場合と同じ)。

チェックイン / チェックアウト機能の扱い方は、デザインノート機能と異なります。ソースコントロールシステムがこの機能をサポートする場合は、[デザインノート] ダイアログボックスからこの機能を無効にすることはできません。ユーザーがソースコントロールシステムを上書きしようとすると、エラーメッセージが表示されます。

ソースコントロールシステム機能の追加

ソースコントロールシステム機能を Dreamweaver に追加するには、メニュー項目と対応する C 関数のセットを返す `GetNewFeatures` ハンドラを作成します。たとえば、Sourcesafe ライブラリを作成する際に、Dreamweaver ユーザーがファイルの履歴を参照できるようにするには、[ヒストリ] メニュー項目と C の `history` 関数を返す `GetNewFeatures` ハンドラを作成します。これで、ユーザーが Windows でファイルを右クリックすると、[ヒストリ] がメニュー項目の 1 つとして表示されます。ユーザーが [ヒストリ] メニュー項目を選択すると、対応する関数が Dreamweaver によって呼び出され、選択したファイルが DLL に渡されます。DLL によって [ヒストリ] ダイアログボックスが表示され、ユーザーは Sourcesafe を使用する場合と同じように DLL と対話できます。

ソースコントロール統合 API の必須の関数

ソースコントロール統合 API には、必須の関数とオプションの関数があります。この項では、必須の関数を挙げています。

bool SCS_GetAgentInfo()

説明

この関数は、DLL にその名前と説明を返すように要求します。この名前と説明は、[サイトの編集] ダイアログボックスに表示されます。名前 (sourcesafe、webdav、perforce など) は [アクセス] ポップアップメニューに表示され、説明はポップアップメニューの下に表示されます。

引数

`char name[32]`、`char version[32]`、`char description[256]`、`const char *dwAppVersion`

- *`name`* 引数は、ソースコントロールシステムの名前です。この名前は、ソースコントロールシステムを選択するためのコンボボックスに表示されます。このボックスは、[サイトの編集] ダイアログボックスの [SourceControl] タブに表示されます。名前の最大文字数は 32 文字です。
- *`version`* 引数は、DLL のバージョンを示すストリングです。このバージョンは、[サイトの編集] ダイアログボックスの [SourceControl] タブに表示されます。バージョンの最大文字数は 32 文字です。
- *`description`* 引数は、ソースコントロールシステムの説明を示すストリングです。この説明は、[サイトの編集] ダイアログボックスの [SourceControl] タブに表示されます。説明の最大文字数は 256 文字です。
- *`dwAppVersion`* 引数は、DLL を呼び出している Dreamweaver のバージョンを示すストリングです。DLL は、このストリングから Dreamweaver のバージョンと言語を判断します。

戻り値

ブール値。成功した場合は `true`、失敗した場合は `false`。

bool SCS_Connect()

説明

この関数は、ユーザーをソースコントロールシステムに接続します。DLL にログイン情報がない場合、DLL はユーザーに情報の入力を促すダイアログボックスを表示し、そのデータを後で使用するために保存する必要があります。

引数

`void **connectionData`、`const char siteName[64]`

- *`connectionData`* 引数は、Dreamweaver が他の API 関数を呼び出す際にエージェントに渡す必要のあるデータへのハンドルです。
- *`siteName`* 引数は、サイトの名前を示すストリングです。サイト名の最大文字数は 64 文字です。

戻り値

ブール値。成功した場合は `true`、失敗した場合は `false`。

bool SCS_Disconnect()

説明

この関数は、ソースコントロールシステムからユーザーを切断します。

引数

`void *connectionData`

- *`connectionData`* 引数は、`Connect()` の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。

戻り値

ブール値。成功した場合は `true`、失敗した場合は `false`。

bool SCS_IsConnected()

説明

この関数は、接続の状態を判断します。

引数

`void *connectionData`

- *`connectionData`* 引数は、`Connect()` の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。

戻り値

ブール値。成功した場合は `true`、失敗した場合は `false`。

int SCS_GetRootFolderLength()

説明

この関数は、ルートフォルダの名前の長さを返します。

引数

`void *connectionData`

- *`connectionData`* 引数は、`Connect()` の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。

戻り値

ルートフォルダの名前の長さを示す整数。この関数から `0` より小さい値が返された場合は、エラーが発生したと見なされ、サポートされていれば DLL からエラーメッセージが取得されます。

bool SCS_GetRootFolder()

説明

この関数は、ルートフォルダの名前を返します。

引数

`void *connectionData`、*`char remotePath[]`*、*`const int folderLen`*

- *`connectionData`* 引数は、`Connect()` の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *`remotePath`* は、ルートフォルダの完全なリモートパスを格納するバッファです。
- *`folderLen`* 引数は、*`remotePath`* の長さを示す整数です。これは、*`GetRootFolderLength`* から返される値です。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

int SCS_GetFolderListLength()

説明

この関数は、渡されるフォルダ内のアイテムの数を返します。

引数

*void *connectionData*、*const char *remotePath*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePath* 引数は、DLL によってアイテムの数がチェックされるリモートフォルダのフルパスと名前です。

戻り値

現在のフォルダ内にあるアイテムの数を示す整数。この関数から 0 より小さい値が返された場合は、エラーが発生したと見なされ、サポートされていれば DLL からエラーメッセージが取得されます。

bool SCS_GetFolderList()

説明

この関数は、渡されたフォルダ内のファイルとフォルダのリストを返します。このリストには、変更日、サイズ、アイテムがフォルダとファイルのどちらであるかなどの関連情報も含まれます。

引数

*void *connectionData*、*const char *remotePath*、*itemInfo itemList[ ]*、*const int numItems*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePath* は、DLL によってアイテムの数がチェックされるリモートフォルダのパスです。
- *itemList* 引数は、事前に割り当てられた itemInfo 構造体のリストです。

name	char[256]	ファイルまたはフォルダの名前
isFolder	bool	フォルダの場合は true、ファイルの場合は false
month	int	修正日付の月の部分 (1 ～ 12)
day	int	修正日付の日の部分 (1 ～ 31)
year	int	修正日付の年の部分 (1900+)
hour	int	修正日付の時間の部分 (0 ～ 23)
minutes	int	修正日付の分の部分 (0 ～ 59)
seconds	int	修正日付の秒の部分 (0 ～ 59)
type	char[256]	ファイルのタイプ (DLL が設定しない場合は Dreamweaver がファイル拡張子からタイプを決定)
size	int	バイト単位

- *numItems* 引数は、itemList に割り当てられたアイテムの数 (GetFolderListLength からの戻り値) です。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

bool SCS_Get()

説明

この関数は、ファイルまたはフォルダのリストを取得し、ローカルに保存します。

引数

*void *connectionData*、*const char *remotePathList[]*、*const char *localPathList[]*、*const int numItems*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePathList* 引数は、取得するリモートファイルまたはフォルダの完全なパスと名前のリストです。
- *localPathList* 引数は、ローカルファイル名またはフォルダパスのミラーリストです。
- *numItems* 引数は、各リストに含まれるアイテムの数です。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

bool SCS_Put()

説明

ローカルファイルまたはフォルダのリストをソースコントロールシステムに送信します。

引数

*void *connectionData*、*const char *localPathList[]*、*const char *remotePathList[]*、*const int numItems*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *localPathList* 引数は、ソースコントロールシステムに送信されるローカルファイル名またはフォルダパスのリストです。
- *remotePathList* 引数は、リモートファイル名またはフォルダパスのミラーリストです。
- *numItems* 引数は、各リストに含まれるアイテムの数です。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

bool SCS_NewFolder()

説明

この関数は、新しいフォルダを作成します。

引数

*void *connectionData*、*const char *remotePath*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePath* 引数は、DLL によって作成されるリモートフォルダのフルパスです。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

bool SCS_Delete()

説明

この関数は、ファイルまたはフォルダのリストをソースコントロールシステムから削除します。

引数

*void *connectionData*、*const char *remotePathList[]*、*const int numItems*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePathList* 引数は、削除するリモートファイル名またはフォルダパスのリストです。
- *numItems* 引数は、*remotePathList* に含まれるアイテムの数です。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

bool SCS_Rename()

説明

oldRemotePath および *newRemotePath* に指定された値に従って、ファイルやフォルダを移動するか、またはその名前を変更します。たとえば、*oldRemotePath* に "$/folder1/file1" を指定し、*newRemotePath* に "$/folder1/renamefile1" を指定すると、"file1" という名前が "renamefile1" に変更され、このファイルが "folder1" に配置されます。

また、*oldRemotePath* に "$/folder1/file1" を指定し、*newRemotePath* に "$/folder1/subfolder1/file1" を指定すると、"file1" が "subfolder1" フォルダに移動します。

この関数を呼び出したときに移動または名前の変更のどちらが実行されるかを確認するには、2 つの入力値の親パスをチェックします。両方の値が同じであれば、名前が変更されます。

引数

*void *connectionData*、*const char *oldRemotePath*、*const char *newRemotePath*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *oldRemotePath* 引数は、名前を変更するリモートファイルまたはフォルダパスです。
- *newRemotePath* 引数は、このファイルまたはフォルダの新しい名前のリモートパスです。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

bool SCS_ItemExists()

説明

ファイルまたはフォルダがサーバー上に存在するかどうかを判別します。

引数

*void *connectionData*、*const char *remotePath*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePath* 引数は、リモートファイルまたはフォルダパスです。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

ソースコントロール統合 API のオプションの関数

ソースコントロール統合 API には、必須の関数とオプションの関数があります。このセクションでは、オプションの関数を挙げています。

bool SCS_GetConnectionInfo()

説明

ユーザーがこのサイトの接続情報を変更または設定するためのダイアログボックスを表示します。接続の確立は行いません。この関数は、ユーザーが [サイトの編集] ダイアログボックスの [リモート情報] セクションで、[設定] ボタンをクリックすると呼び出されます。

引数

*void **connectionData、const char siteName[64]*

- *connectionData* は、Dreamweaver が他の API 関数を呼び出す際にエージェントに渡す必要のあるデータへのハンドルです。
- *siteName* 引数は、サイトの名前を示すストリングです。名前の最大文字数は 64 文字です。

戻り値

ブール値。成功した場合は `true`、失敗した場合は `false`。

bool SCS_SiteDeleted()

説明

サイトが削除されていること、またはサイトとこのソースコントロールシステムとの関連付けが解除されていることを DLL に通知します。これで、ソースコントロールシステムによって、このサイトの永続的な情報を削除できるようになります。

引数

const char siteName[64]

- *siteName* 引数は、サイトの名前を示すストリングです。名前の最大文字数は 64 文字です。

戻り値

ブール値。成功した場合は `true`、失敗した場合は `false`。

bool SCS_SiteRenamed()

説明

ユーザーがサイトの名前を変更したために、そのサイトに関する永続的な情報が更新可能であることを DLL に通知します。

引数

const char oldSiteName[64]、const char newSiteName[64]

- *oldSiteName* 引数は、サイトの名前を変更する前の元の名前を示すストリングです。名前の最大文字数は 64 文字です。
- *newSiteName* 引数は、サイトの名前を変更した後の新しい名前を示すストリングです。名前の最大文字数は 64 文字です。

戻り値

ブール値。成功した場合は `true`、失敗した場合は `false`。

int SCS_GetNumNewFeatures()

説明

Dreamweaver に追加する新しい機能 (たとえば、[ヒストリ] や [ファイル差異] など) の数を返します。

引数

なし

戻り値

Dreamweaver に追加する新しい機能の数を示す整数。この関数から 0 より小さい値が返された場合は、エラーが発生したと見なされ、サポートされていれば DLL からエラーメッセージが取得されます。

bool SCS_GetNewFeatures()

説明

この関数は、Dreamweaver のメインメニューまたはコンテキストメニューに追加するメニュー項目のリストを返します。たとえば、Sourcesafe DLL では、メインメニューに [ヒストリ] および [ファイル差異] を追加できます。

引数

char menuItemList[][32]、*scFunction functionList[]*、*scFunction enablerList[]*、*const int numNewFeatures*

- *menuItemList* 引数は、DLL によって値が取り込まれるストリングリストです。このリストは、メインメニューおよびコンテキストメニューに追加するメニュー項目を指定します。各ストリングの最大長は 32 文字です。
- *functionList* には、DLL によって値が取り込まれます。このリストは、ユーザーがメニュー項目を選択すると呼び出される DLL 内のルーチンを指定します。
- *enablerList* 引数には、DLL によって値が取り込まれます。このリストは、メニュー項目が使用可能であるかどうかを Dreamweaver が判断する際に呼び出される DLL 内のルーチンを指定します。
- *numNewFeatures* 引数は、DLL によって追加される項目の数です。この値は、GetNumNewFeatures() を呼び出すことで取得されます。

次の関数の署名では、SCS_GetNewFeatures() 呼び出しで functionlist 引数および enablerList 引数に渡される関数とイネーブラを定義します。

```
bool (*scFunction)(void *connectionData, const char *remotePathList[],
    const char *localPathList[], const int numItems)
```

戻り値

ブール値。成功した場合は true、失敗した場合は false。

bool SCS_GetCheckoutName()

説明

この関数は、現在のユーザーのチェックアウト名を返します。ソースコントロールシステムでこの機能がサポートされていない場合に、ユーザーがこの機能を有効にすると、ソースコントロールシステムに対する LCK ファイルの送受信に、Dreamweaver の内部チェックイン / チェックアウト機能が使用されます。

引数

*void *connectionData*、*char checkOutName[64]*、*char emailAddress[64]*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *checkOutName* 引数には、現在のユーザーの名前です。
- *emailAddress* は、現在のユーザーの電子メールアドレスです。

戻り値

ブール値。成功した場合は `true`、失敗した場合は `false`。

bool SCS_Checkin()

説明

この関数は、ローカルファイルまたはフォルダのリストを、ソースコントロールシステムに送信します。DLL では、このファイルを読み取り専用にする必要があります。ソースコントロールシステムでこの機能がサポートされていない場合に、ユーザーがこの機能を有効にすると、ソースコントロールシステムに対する LCK ファイルの送受信に、Dreamweaver の内部チェックイン / チェックアウト機能が使用されます。

引数

`void *connectionData`、*`const char *localPathList[]`*、*`const char *remotePathList[]`*、*`bool successList[]`*、*`const int numItems`*

- *`connectionData`* 引数は、`Connect()` の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *`localPathList`* 引数は、チェックインするローカルファイル名またはフォルダパスのリストです。
- *`remotePathList`* 引数は、リモートファイル名またはフォルダパスのミラーリストです。
- *`successList`* 引数には、DLL によってブール値のリストが取り込まれます。これにより、正常にチェックインしたファイルを Dreamweaver に通知します。
- *`numItems`* 引数は、各リストに含まれるアイテムの数です。

戻り値

ブール値。成功した場合は `true`、失敗した場合は `false`。

bool SCS_Checkout()

説明

ローカルファイルまたはフォルダのリストを、ソースコントロールシステムからチェックアウトします。DLL によって、ファイルを書き込み可能にする特権が与えられます。ソースコントロールシステムでこの機能がサポートされていない場合に、ユーザーがこの機能を有効にすると、ソースコントロールシステムに対する LCK ファイルの送受信に、Dreamweaver の内部チェックイン / チェックアウト機能が使用されます。

引数

`void *connectionData`、*`const char *remotePathList[]`*、*`const char *localPathList[]`*、*`bool successList[]`*、*`const int numItems`*

- *`connectionData`* 引数は、`Connect()` の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *`remotePathList`* 引数は、チェックアウトするリモートファイル名またはフォルダパスのリストです。
- *`localPathList`* 引数は、ローカルファイル名またはフォルダパスのミラーリストです。
- *`successList`* 引数には、DLL によってブール値のリストが取り込まれます。これにより、正常にチェックアウトしたファイルを Dreamweaver に通知します。
- *`numItems`* 引数は、各リストに含まれるアイテムの数です。

戻り値

ブール値。成功した場合は `true`、失敗した場合は `false`。

bool SCS_UndoCheckout()

説明

ファイルまたはフォルダのリストのチェックアウト状態を取り消します。DLL では、このファイルを読み取り専用にする必要があります。ソースコントロールシステムでこの機能がサポートされていない場合に、ユーザーがこの機能を有効にすると、ソースコントロールシステムに対する LCK ファイルの送受信に、Dreamweaver の内部チェックイン / チェックアウト機能が使用されます。

引数

*void *connectionData*、*const char *remotePathList[]*、*const char *localPathList[]*、*bool successList[]*、*const int numItems*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePathList* 引数は、チェックアウトを取り消すリモートファイル名またはフォルダパスのリストです。
- *localPathList* 引数は、ローカルファイル名またはフォルダパスのミラーリストです。
- *successList* 引数には、DLL によってブール値のリストが取り込まれます。これにより、チェックアウトが正常に取り消されたファイルを Dreamweaver に通知します。
- *numItems* 引数は、各リストに含まれるアイテムの数です。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

int SCS_GetNumCheckedOut()

説明

ファイルをチェックアウトしたユーザーの数を返します。

引数

*void *connectionData*、*const char *remotePath*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePath* 引数は、チェックアウトしたユーザーの数を確認するリモートファイルまたはフォルダパスです。

戻り値

ファイルをチェックアウトしたユーザーの数を示す整数。この関数から 0 より小さい値が返された場合は、エラーが発生したと見なされ、サポートされていれば DLL からエラーメッセージが取得されます。

bool SCS_GetFileCheckoutList()

説明

ファイルをチェックアウトしたユーザーのリストを返します。このリストが空であれば、ファイルをチェックアウトしたユーザーがいないことになります。

引数

*void *connectionData*、*const char *remotePath*、*char checkOutList[][64]*、*char emailAddressList[][64]*、*const int numCheckedOut*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePath* 引数は、チェックアウトしたユーザーの数を確認するリモートファイルまたはフォルダパスです。
- *checkOutList* 引数は、ファイルをチェックアウトしたユーザーを表すストリングのリストです。各ユーザーストリングの最大文字数は 64 文字です。

- *emailAddressList* 引数は、ユーザーの電子メールアドレスを表すストリングのリストです。各電子メールアドレスストリングの最大文字数は 64 文字です。
- *numCheckedOut* 引数は、ファイルをチェックアウトしたユーザーの数です。この数値は、GetNumCheckedOut() から返されます。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

int SCS_GetErrorMessageLength()

説明

DLL の現在の内部エラーメッセージの長さを返します。この結果、GetErrorMessage() 関数に渡されるバッファが割り当てられます。この関数は、API 関数からその API 関数が失敗したことを示す false または 0 より小さい値が返された場合にのみ呼び出してください。

引数

*void *connectionData*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。

戻り値

エラーメッセージの長さを表す整数。

bool SCS_GetErrorMessage()

説明

この関数は、最後のエラーメッセージを返します。getErrorMessage() を実装すると、Dreamweaver は API 関数が false 値を返すたびにこの関数を呼び出します。

ルーチンから -1 または false が返された場合は、エラーメッセージが発生しています。

引数

*void *connectionData*、*char errorMsg[]*、*const int *msgLength*

connectionData 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。

- *errorMsg* 引数は、DLL がエラーメッセージを設定するために事前に割り当てられたストリングです。
- *msgLength* は、*errorMsg[]* 引数が示すバッファの長さです。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

int SCS_GetNoteCount()

説明

この関数は、指定されたリモートファイルパスまたはフォルダパスのデザインノートキーの数を返します。ソースコントロールシステムがこの機能をサポートしない場合、Dreamweaver は関連する MNO ファイルからこの情報を取得します。

引数

*void *connectionData*、*const char *remotePath*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePath* 引数は、添付されているデザインノートの数が DLL によってチェックされるリモートファイルまたはフォルダパスです。

戻り値

このファイルに関連付けられているデザインノートの数を示す整数。この関数から 0 より小さい値が返された場合は、エラーが発生したと見なされ、サポートされていれば DLL からエラーメッセージが取得されます。

int SCS_GetMaxNoteLength()

説明

この関数は、指定されたファイルまたはフォルダの最も大きなデザインノートの長さを返します。ソースコントロールシステムがこの機能をサポートしない場合、Dreamweaver は関連する MNO ファイルからこの情報を取得します。

引数

*void *connectionData*、*const char *remotePath*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePath* 引数は、DLL によって最も大きなデザインノートの長さがチェックされるリモートファイルまたはフォルダパスです。

戻り値

このファイルに関連付けられている最も大きなデザインノートのサイズを示す整数。この関数から 0 より小さい値が返された場合は、エラーが発生したと見なされ、サポートされていれば DLL からエラーメッセージが取得されます。

bool SCS_GetDesignNotes()

説明

この関数は、指定されたファイルまたはフォルダのメタ情報からキー / 値ペアを取得します。ソースコントロールシステムがこの機能をサポートしない場合、Dreamweaver は関連する MNO ファイルからこの情報を取得します。

引数

*void *connectionData*、*const char *remotePath*、*char keyList[][64]*、*char *valueList[]*、*bool showColumnList[]*、*const int noteCount*、*const int noteLength*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePath* 引数は、DLL によってアイテムの数がチェックされるリモートファイルまたはフォルダパスです。
- *keyList* 引数は、"Status" などのデザインノートキーのリストです。
- *valueList* 引数は、デザインノートキーに対応するデザインノート値 ("Awaiting Signoff" など) のリストです。
- *showColumnList* 引数は、デザインノートキーに対応するブール値のリストです。このブール値は、Dreamweaver でそのキーを [サイト] パネルに列として表示できるかどうかを示します。
- *noteCount* 引数は、ファイルまたはフォルダに添付されているデザインノートの数です。この値は GetNoteCount() 呼び出しから返されます。
- *noteLength* 引数は、デザインノートの最大長です。これは、GetMaxNoteLength() 呼び出しで返される値です。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

bool SCS_SetDesignNotes()

説明

この関数は、キー / 値ペアを、指定されたファイルまたはフォルダのメタ情報に格納します。これにより、そのファイルのメタ情報のセットが置換されます。ソースコントロールシステムでこの機能がサポートされていない場合、Dreamweaver では MNO ファイルにデザインノートが格納されます。

引数

*void *connectionData*、*const char *remotePath*、*const char keyList[][64]*、*const char *valueList[]*、*bool showColumnList[]*、*const int noteCount*、*const int noteLength*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePath* 引数は、DLL によってアイテムの数がチェックされるリモートファイルまたはフォルダパスです。
- *keyList* 引数は、"Status" などのデザインノートキーのリストです。
- *valueList* 引数は、デザインノートキーに対応するデザインノート値 ("Awaiting Signoff" など) のリストです。
- *showColumnList* 引数は、デザインノートキーに対応するブール値のリストです。このブール値は、Dreamweaver でそのキーを [サイト] パネルに列として表示できるかどうかを示します。
- *noteCount* 引数は、ファイルまたはフォルダに添付されているデザインノートの数です。この数値によって、指定されたリストのサイズを DLL に通知します。*noteCount* が 0 の場合は、ファイルからすべてのデザインノートが削除されます。
- *noteLength* 引数は、指定されたファイルまたはフォルダの最も大きなデザインノートの長さです。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

bool SCS_IsRemoteNewer()

説明

この関数は、指定されたリモートパスをチェックして、そのリモートコピーが最新かどうかを確認します。ソースコントロールシステムでこの機能がサポートされていない場合、Dreamweaver ではその内部 isRemoteNewer アルゴリズムが使用されます。

引数

*void *connectionData*、*const char *remotePathList[]*、*const char *localPathList[]*、*int remoteIsNewerList[]*、*const int numItems*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePathList* 引数は、最新ステータスを確認するために比較するリモートファイル名またはフォルダパスのリストです。
- *localPathList* 引数は、ローカルファイル名またはフォルダパスのミラーリストです。
- *remoteIsNewerList* 引数には、DLL によって整数のリストが取り込まれます。これにより、リモート側で対応するファイルの中で最新のものを Dreamweaver に通知します。有効な値は以下のとおりです。1 はリモートバージョンが最新であることを示し、-1 はローカルバージョンが最新であることを示します。0 は、両方のバージョンが同じであることを示します。
- *numItems* 引数は、各リストに含まれるアイテムの数です。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

イネーブラ

以下に説明するオプションのイネーブラがソースコントロールシステムでサポートされていない場合、またはアプリケーションがサーバーに接続されていない場合、Dreamweaver では所有するリモートファイルに関する情報を基に、メニュー項目を使用可能にするかどうかが決定されます。

bool SCS_canConnect()

説明

この関数は、[接続] メニュー項目を使用可能にするかどうかを返します。

引数

なし

戻り値

ブール値。成功した場合は `true`、失敗した場合は `false`。

bool SCS_canGet()

説明

この関数は、[GET] メニュー項目を使用可能にするかどうかを返します。

引数

`void *connectionData`、*`const char *remotePathList[]`*、*`const char *localPathList[]`*、*`const int numItems`*

- *`connectionData`* 引数は、`Connect()` の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *`remotePathList`* 引数は、取得するリモートファイル名またはフォルダパスのリストです。
- *`localPathList`* 引数は、ローカルファイル名またはフォルダパスのミラーリストです。
- *`numItems`* 引数は、各リストに含まれるアイテムの数です。

戻り値

ブール値。成功した場合は `true`、失敗した場合は `false`。

bool SCS_canCheckout()

説明

この関数は、[チェックアウト] メニュー項目を使用可能にするかどうかを返します。

引数

`void *connectionData`、*`const char *remotePathList[]`*、*`const char *localPathList[]`*、*`const int numItems`*

- *`connectionData`* 引数は、`Connect()` の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *`remotePathList`* 引数は、チェックアウトするリモートファイル名またはフォルダパスのリストです。
- *`localPathList`* 引数は、ローカルファイル名またはフォルダパスのミラーリストです。
- *`numItems`* 引数は、各リストに含まれるアイテムの数です。

戻り値

ブール値。成功した場合は `true`、失敗した場合は `false`。

bool SCS_canPut()

説明

この関数は、[PUT] メニュー項目を使用可能にするかどうかを返します。

引数

*void *connectionData*、*const char *localPathList[]*、*const char *remotePathList[]*、*const int numItems*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *localPathList* 引数は、ソースコントロールシステムに送信されるローカルファイル名またはフォルダパスのリストです。
- *remotePathList* 引数は、ソースコントロールシステムに送信されるリモートファイル名またはフォルダパスのミラーリストです。
- *numItems* 引数は、各リストに含まれるアイテムの数です。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

bool SCS_canCheckin()

説明

この関数は、[チェックイン] メニュー項目を使用可能にするかどうかを返します。

引数

*void *connectionData*、*const char *localPathList[]*、*const char *remotePathList[]*、*const int numItems*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *localPathList* 引数は、チェックインするローカルファイル名またはフォルダパスのリストです。
- *remotePathList* 引数は、リモートファイル名またはフォルダパスのミラーリストです。
- *numItems* 引数は、各リストに含まれるアイテムの数です。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

bool SCS_CanUndoCheckout()

説明

この関数は、[チェックアウトの取り消し] メニュー項目を使用可能にするかどうかを返します。

引数

*void *connectionData*、*const char *remotePathList[]*、*const char *localPathList[]*、*const int numItems*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePathList* 引数は、チェックアウトするリモートファイル名またはフォルダパスのリストです。
- *localPathList* 引数は、ソースコントロールシステムに送信されるローカルファイル名またはフォルダパスのリストです。
- *numItems* 引数は、各リストに含まれるアイテムの数です。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

bool SCS_canNewFolder()

説明

この関数は、[新規フォルダ] メニュー項目を使用可能にするかどうかを返します。

引数

*void *connectionData*、*const char *remotePath*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePath* 引数は、新しいフォルダが作成される場所としてユーザーが選択した、リモートファイル名またはフォルダパスのリストです。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

bool SCS_canDelete()

説明

この関数は、[削除] メニュー項目を使用可能にするかどうかを返します。

引数

*void *connectionData*、*const char *remotePathList[]*、*const int numItems*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePathList* 引数は、削除するリモートファイル名またはフォルダパスのリストです。
- *numItems* 引数は、各リストに含まれるアイテムの数です。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

bool SCS_canRename()

説明

この関数は、[名前の変更] メニュー項目を使用可能にするかどうかを返します。

引数

*void *connectionData*、*const char *remotePath*

- *connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。
- *remotePathList* 引数は、名前を変更できるリモートファイル名またはフォルダパスのリストです。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

bool SCS_BeforeGet()

説明

Dreamweaver は、ファイルを取得またはチェックアウトする前にこの関数を呼び出します。この関数により、チェックアウトコメントの追加などの単一操作が DLL によってファイルのグループに対して実行されます。

引数

**connectionData*

- **connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

例

Dreamweaver では、ファイルのグループを取得するために、次の順序で DLL に対する呼び出しが実行されます。

```
SCS_BeforeGet(connectionData);
SCS_Get(connectionData,remotePathList1,localPathList1,successList1);
SCS_Get(connectionData,remotePathList2,localPathList2,successList2);
SCS_Get(connectionData,remotePathList3,localPathList3,successList3);
SCS_AfterGet(connectionData);
```

bool SCS_BeforePut()

説明

Dreamweaver は、ファイルを送信またはチェックインする前にこの関数を呼び出します。この関数により、チェックインコメントの追加などの単一操作が、DLL によってファイルのグループに対して実行されます。

引数

**connectionData*

- **connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

例

ファイルのグループを取得するために、Dreamweaver は以下の順序で DLL に対する呼び出しを実行します。

```
SCS_BeforePut(connectionData);
SCS_Put(connectionData,localPathList1,remotePathList1,successList1);
SCS_Put(connectionData,localPathList2,remotePathList2,successList2);
SCS_Put(connectionData,localPathList3,remotePathList3,successList3);
SCS_AfterPut(connectionData);
```

bool SCS_AfterGet()

説明

Dreamweaver は、ファイルを取得またはチェックアウトした後でこの関数を呼び出します。この関数により、バッチによる取得またはチェックアウトの後で、[サマリー] ダイアログボックスの作成などの任意の操作が DLL によって実行されます。

引数

**connectionData*

- **connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

例

詳細については、89 ページの bool SCS_BeforeGet() を参照してください。

bool SCS_AfterPut()

説明

Dreamweaver は、ファイルを送信またはチェックインした後でこの関数を呼び出します。この関数により、バッチによる送信またはチェックインの後で、[サマリー] ダイアログボックスの作成などの任意の操作が DLL によって実行されます。

引数

*connectionData

- **connectionData* 引数は、Connect() の呼び出しで Dreamweaver に渡されたエージェントのデータへのポインタです。

戻り値

ブール値。成功した場合は true、失敗した場合は false。

例

詳細については、90 ページの bool SCS_BeforePut() を参照してください。

第 11 章 : アプリケーション

アプリケーション関数は、Adobe® Dreamweaver® CS3 と他のアプリケーションとの対話に関連する操作、および個々のドキュメントと関係のない Dreamweaver の操作 (環境設定や Dreamweaver の終了など) を実行します。

本章では関数をグループ分けして、それぞれを以下の項で説明します。

外部アプリケーション関数

外部アプリケーション関数は、Adobe® Flash® などのアプリケーションに関連する操作、および [環境設定] の [ブラウザでプレビュー] と [外部エディタ] で定義されているブラウザや外部エディタに関連する操作を行います。これらの関数を使用して、外部アプリケーションに関する情報を取得し、外部アプリケーションでファイルを開くことができます。

dreamweaver.browseDocument()

対応バージョン

Dreamweaver 2、3、4 (3 および 4 で機能強化)

説明

指定された URL を、指定されたブラウザで開きます。

引数

`fileName`、`{browser}`

- *`fileName`* 引数には、開くファイルの名前を絶対 URL で指定します。

注意 : 一部のブラウザでは、たとえば "*Configuration/ExtensionHelp/browseHelp.htm#helpyou*" のように *URL* にアンカーが含まれていると、ファイルを見つけることができません。

- Dreamweaver 3 で追加された *`browser`* 引数には、ブラウザを指定します。この引数に指定できるのは、[環境設定] の [ブラウザでプレビュー] に定義されているブラウザの名前か、`'primary'` と `'secondary'` のいずれかです。この引数を省略した場合、URL はユーザーのプライマリブラウザで表示されます。

戻り値

なし

例

次の関数は、`dreamweaver.browseDocument()` 関数を使用してブラウザで HotWired のホームページを開きます。

```
function goToHotwired(){
    dreamweaver.browseDocument('http://www.hotwired.com/');
}
```

Dreamweaver 4 では、次のコードに示すように、この操作を拡張して Microsoft Internet Explorer でドキュメントを開くことができます。

```
function goToHotwired(){
    var prevBrowsers = dw.getBrowserList();
    var theBrowser = "";
    for (var i=1; i < prevBrowsers.length; i+2){
        if (prevBrowsers[i].indexOf('Iexplore.exe') != -1){
            theBrowser = prevBrowsers[i];
            break;
        }
    }
    dw.browseDocument('http://www.hotwired.com/',theBrowser);
}
```

`dreamweaver.getBrowserList()` 関数の詳細については、93 ページの dreamweaver.getBrowserList() を参照してください。

dreamweaver.getBrowserList()

対応バージョン

Dreamweaver 3

説明

[ファイル]-[ブラウザでプレビュー] サブメニューにあるすべてのブラウザのリストを取得します。

引数

なし

戻り値

リスト内のブラウザごとにストリングのペアを格納した配列。各ペアの最初のストリングはブラウザの名前を表し、2 番目のストリングはユーザーのコンピュータにおけるブラウザの場所を file:// URL 形式で表します。[ブラウザでプレビュー] サブメニューにブラウザが表示されない場合、戻り値はありません。

dreamweaver.getExtensionEditorList()

対応バージョン

Dreamweaver 3

説明

[環境設定] の [外部エディタ] から、指定されたファイルに対するエディタのリストを取得します。

引数

`fileURL`

- *`fileURL`* 引数には、file:// で始まる完全な URL、ファイル名、またはファイル拡張子 (ピリオドも含む) を指定します。

戻り値

リスト内のエディタごとにストリングのペアを格納した配列。各ペアの最初のストリングはエディタの名前を表し、2 番目のストリングはユーザーのコンピュータにおけるエディタの場所を file:// URL 形式で表します。[環境設定] にエディタが表示されない場合、この関数は空白のストリング 1 つを含む配列を返します。

例

`dreamweaver.getExtensionEditorList(".gif")` 関数を呼び出すと、以下のストリングを格納した配列が返されます。

- `"Fireworks 3"`
- `"file:///C|/Program Files/Adobe/Fireworks 3/Fireworks 3.exe"`

dreamweaver.getExternalTextEditor()

対応バージョン
Dreamweaver 4

説明
現在設定されている外部テキストエディタの名前を取得します。

引数
なし

戻り値
完全なパスではなく、ユーザーインターフェイス (UI) での表示に適したテキストエディタの名前が格納されたストリング。

dreamweaver.getFlashPath()

対応バージョン
Dreamweaver MX

説明
Flash MX アプリケーションへの完全なパスを、file:// URL 形式で取得します。

引数
なし

戻り値
2 つのエレメントを格納した配列。エレメント [0] は、Flash MX エディタの名前を格納したストリングです。エレメント [1] は、ローカルコンピュータ上の Flash アプリケーションへのパスを file:// で始まる URL の形式で表したストリングです。Flash がインストールされていない場合は、戻り値はありません。

例
次の例は、`dw.getFlashPath()` 関数を呼び出して Flash アプリケーションへのパスを取得し、`dw.openWithApp()` 関数にそのパスを file:// URL 形式で渡し、Flash でドキュメントを開きます。

```
var myDoc = dreamweaver.getDocumentDOM();

if (dreamweaver.validateFlash()) {
    var flashArray = dreamweaver.getFlashPath();
    dreamweaver.openWithApp( myDoc.myForm.swfFilePath, flashArray[1] );
}
```

dreamweaver.getPrimaryBrowser()

対応バージョン
Dreamweaver 3

説明
プライマリブラウザへのパスを取得します。

引数
なし

戻り値

ユーザーのコンピュータ上のプライマリブラウザへのパスを、file:// で始まる URL の形式で表したストリング。プライマリブラウザが定義されていない場合は、戻り値はありません。

dreamweaver.getPrimaryExtensionEditor()

対応バージョン

Dreamweaver 3

説明

指定されたファイルのプライマリエディタを取得します。

引数

fileURL

- *fileURL* 引数には、開くファイルへのパスを file:// URL 形式で指定します。

戻り値

ストリングのペアを格納した配列。ペアの最初のストリングはエディタの名前を表し、2 番目のストリングはユーザーのコンピュータにおけるエディタの場所を file:// URL 形式で表します。プライマリエディタが定義されていない場合、この関数は空白のストリングを 1 つ含む配列を返します。

dreamweaver.getSecondaryBrowser()

対応バージョン

Dreamweaver 3

説明

セカンダリブラウザへのパスを取得します。

引数

なし

戻り値

ユーザーのコンピュータ上のセカンダリブラウザへのパスを、file:// で始まる URL の形式で表したストリング。セカンダリブラウザが定義されていない場合は、戻り値はありません。

dreamweaver.openHelpURL()

対応バージョン

Dreamweaver MX

説明

指定されたヘルプファイルを、オペレーティングシステムのヘルプビューアで開きます。

Dreamweaver では、ブラウザではなく、オペレーティングシステム標準のヘルプビューアでヘルプのコンテンツが表示されます。ヘルプのコンテンツは HTML 形式ですが、Windows HTML ヘルプまたは Mac OS X のヘルプビューアで参照できるようにパッケージされています。

ヘルプコンテンツ全体は、以下の 4 種類のファイルで構成されています。ヘルプファイルの詳細については、オペレーティングシステムのマニュアルを参照してください。

• ヘルプブック

ヘルプブックは、HTML ヘルプファイル、イメージ、および索引で構成されています。Windows では、ヘルプブックは .chm という拡張子の付いたファイルです。Macintosh では、ヘルプブックはフォルダです。

ヘルプブックのファイルは、Dreamweaver の "Help" フォルダにあります。

• "help.xml" ファイル

"help.xml" ファイルは、ブック ID をヘルプブック名にマップします。たとえば、次の XML コードは、Dreamweaver ヘルプのブック ID を Windows と Macintosh の両方のオペレーティングシステムに関するヘルプのファイル名にマップしています。

```
<?xml version = "1.0" ?>
<help-books>
<book-id id="DW_Using" win-mapping="UsingDreamweaver.chm" mac-mapping="Dreamweaver Help"/>
</help-books>
```

各 `book-id` エントリには、以下の属性があります。

- `id` 属性は、"help.map" ファイルおよび "HelpDoc.js" ファイルで使用されるブック ID です。
- `win-mapping` 属性は Windows のブック名で、この例では `"UsingDreamweaver.chm"` です。
- `mac-mapping` 属性は Macintosh のブック名で、この例では `"Dreamweaver Help"` です。

• "help.map" ファイル

"help.map" ファイルは、ヘルプコンテンツ ID を特定のヘルプブックにマップします。Dreamweaver は、内部でヘルプを呼び出すときに、"help.map" ファイルを使用してヘルプコンテンツを見つけます。

• "helpDoc.js" ファイル

"helpDoc.js" ファイルを使用すると、実際のブック ID とページストリングの代わりに変数名を使用してマップすることができます。"helpDoc.js" ファイルは、ヘルプコンテンツ ID を特定のヘルプブックの HTML ページにマップします。Dreamweaver は、JavaScript からヘルプを呼び出すときに "helpDoc.js" ファイルを使用します。

引数

`bookID`

• `bookID` 引数 (必須) は、`ID:page` の形式で指定します。

`ID` の部分は、表示するヘルプコンテンツを含むファイルを "help.xml" ファイル内で指定するエントリの `bookID` です。エントリの `page` 部分には、表示する特定のページを指定します。このページが "help.map" ファイルで参照されます。

戻り値

正常に実行された場合は `true`。Dreamweaver が指定されたファイルをヘルプビューアで開けない場合は `false`。

例

```
openHelpURL("DW_Using:index.htm");
```

dreamweaver.openWithApp()

対応バージョン

Dreamweaver 3

説明

指定されたファイルを、指定されたアプリケーションで開きます。

引数

fileURL、*appURL*

- *fileURL* 引数には、開くファイルへのパスを file:// URL 形式で指定します。
- *appURL* 引数には、ファイルを開くのに使用するアプリケーションへのパスを file:// URL 形式で指定します。

戻り値

なし

dreamweaver.openWithBrowseDialog()

対応バージョン

Dreamweaver 3

説明

[外部エディタの選択] ダイアログボックスを開きます。指定したファイルを開くアプリケーションをユーザーが選択することができます。

引数

fileURL

- *fileURL* 引数には、開くファイルへのパスを file:// URL 形式で指定します。

戻り値

なし

dreamweaver.openWithExternalTextEditor()

対応バージョン

Dreamweaver 3

説明

現在のドキュメントを、[環境設定] ダイアログボックスの [外部エディタ] エントリで指定されている外部テキストエディタで開きます。

引数

なし

戻り値

なし

dreamweaver.openWithImageEditor()

対応バージョン

Dreamweaver 3

説明

指定されたファイルを、指定されたイメージエディタで開きます。

注意 : *Fireworks* がイメージエディタとして指定されている場合、この関数はアクティブなドキュメントに情報を返す特殊な *Adobe Fireworks* 統合機構を呼び出します。アクティブなドキュメントがない場合のエラーを防ぐために、[サイト] パネルからはこの関数を呼び出さないでください。

引数

fileURL、*appURL*

- *fileURL* 引数には、開くファイルへのパスを file:// URL 形式で指定します。
- *appURL* 引数には、ファイルを開くのに使用するアプリケーションへのパスを file:// URL 形式で指定します。

戻り値

なし

dreamweaver.validateFlash()

対応バージョン

Dreamweaver MX

説明

Flash MX 以降のバージョンがローカルコンピュータにインストールされているかどうかを判別します。

引数

なし

戻り値

Flash MX 以降のバージョンがローカルコンピュータにインストールされている場合は `true`、それ以外の場合は `false` を示すブール値。

dom.insertFiles()

対応バージョン

Dreamweaver CS3

説明

現在のドキュメントの現在の挿入ポイントに、または現在の選択範囲にファイル (複数可) を挿入します。必要に応じて、ユーザーにパラメータの指定を要求します。

引数

strFiles

strFiles には、挿入するファイルのパスと名前を表すストリングを指定します。この関数に複数のファイル名を渡すことができます。

戻り値

なし

dreamweaver.activateApp()

対応バージョン

Dreamweaver CS3

説明

指定されたアプリケーションをアクティブにして、最前面に表示します。

引数

applicationID

applicationID には、アクティブにするアプリケーションを示すストリング (dreamweaver など) を指定します。

戻り値

なし

dreamweaver.printDocument()

対応バージョン

Dreamweaver CS3

説明

指定されたファイルに対して、Dreamweaver の [ファイル]-[コードの印刷] コマンドと同じ処理を実行します。

引数

fileName

fileName には、印刷するファイル名を表すストリングを URL で指定します。

戻り値

なし

dreamweaver.revealDocument()

対応バージョン

Dreamweaver CS3

説明

Dreamweaver にオペレーティングシステムのフォーカスを設定し、指定されたファイルが Dreamweaver で開かれている場合、そのファイルを前面に表示します。

引数

fileName

fileName には、表示するファイル名を表すストリングを URL で指定します。

戻り値

なし

グローバルアプリケーション関数

グローバルアプリケーション関数は、アプリケーション全体に対して動作します。これらの関数を使用して、アプリケーションの終了や [環境設定] へのアクセスなどを行います。

dreamweaver.beep()

対応バージョン

Dreamweaver MX

説明
システム警告音を作成します。

引数
なし

戻り値
なし

例
次の例では、alert() 関数が表示するメッセージへの注意を促すため、dw.beep() を呼び出しています。

```
beep(){
    if(confirm("Is your order complete?")
    {
        dreamweaver.beep();
        alert("Click OK to submit your order");
    }
}
```

dreamweaver.getShowDialogsOnInsert()

対応バージョン
Dreamweaver 3

説明
[環境設定] の [一般] カテゴリで、[オブジェクト挿入中にダイアログを表示] オプションがオンに設定されているかどうかをチェックします。

引数
なし

戻り値
オプションがオンかどうかを示すブール値。

dreamweaver.quitApplication()

対応バージョン
Dreamweaver 3

説明
この関数を呼び出すスクリプトの実行が完了した後に、Dreamweaver を終了します。

引数
なし

戻り値
なし

dreamweaver.showAboutBox()

対応バージョン
Dreamweaver 3

説明
[Dreamweaver について] ダイアログボックスを開きます。

引数
なし

戻り値
なし

dreamweaver.showDynamicDataDialog()

対応バージョン
Dreamweaver UltraDev 1 (英語版)

説明
[動的データ] ダイアログボックスまたは [動的テキスト] ダイアログボックスを表示して、ユーザーがそのダイアログボックスを閉じるまで待機します。ユーザーが [OK] をクリックすると、`showDynamicDataDialog()` 関数は、ユーザーのドキュメントに挿入されるストリングを返します。このストリングは、データソース API 関数 `generateDynamicDataRef()` が返し、データフォーマット API 関数 `formatDynamicDataRef()` に渡されます。`formatDynamicDataRef()` からの戻り値は、`showDynamicDataDialog()` が返すものです。

引数
source、*{title}*

- *source* 引数には、動的データオブジェクトを表すソースコードを含むストリングを指定します。このストリングは、前回呼び出したときにこの関数が返したものと同じです。この関数は、*source* 引数の内容を使用してダイアログボックスのすべてのコントロールを初期化するので、ダイアログボックスの表示は、ユーザーが [OK] をクリックしてこのストリングを作成したときとまったく同じになります。

Dreamweaver は、このストリングを `inspectDynamicDataRef()` 関数に渡して、ツリー内にこのストリングと一致するノードがあるかどうかを判別します。一致するノードがある場合は、ダイアログボックスが表示されるときにそのノードが選択されます。空白のストリングを渡すこともできます。この場合、ダイアログボックスは初期化されません。たとえば、新規項目を作成するために使用する場合、ダイアログボックスは初期化されません。

- *title* 引数 (オプション) には、ダイアログボックスのタイトルバーに表示するテキストが含まれるストリングを指定します。この引数を省略した場合は、タイトルバーに "Dynamic Data" と表示されます。

戻り値
動的データオブジェクトを表すストリング (ユーザーが [OK] をクリックした場合)。

dreamweaver.showPasteSpecialDialog()

対応バージョン
Dreamweaver 8

説明
[ペーストスペシャル] ダイアログボックスを表示します。ユーザーが [OK] をクリックすると、`showPasteSpecialDialog()` 関数はペーストを実行します。

引数
なし

戻り値
なし

例
```
dw.showPasteSpecialDialog();
```

dreamweaver.showPreferencesDialog()

対応バージョン
Dreamweaver 3 (Dreamweaver 8 で *strCategory* 引数を追加)

説明
[環境設定] ダイアログボックスを開きます。

引数
{strCategory}

- *strCategory* 引数 (オプション) に以下のいずれかのストリングを指定すると、[環境設定] ダイアログボックスの対応するカテゴリが開かれます。指定できるストリングは、"general"、"accessibility"、"html colors" ([コードカラーリング] カテゴリ)、"html format" ([コードフォーマット] カテゴリ)、"code hints"、"html rewriting" ([コードの書き換え] カテゴリ)、"copyPaste"、"css styles"、"file compare"、"external editors" ([ファイルタイプ / エディタ] カテゴリ)、"fonts"、"highlighting"、"invisible elements"、"layers"、"layout mode"、"new document"、"floaters" ([パネル] カテゴリ)、"browsers" ([ブラウザでプレビュー] カテゴリ)、"site ftp" ([サイト] カテゴリ)、"status bar"、および "validator" です。この引数に指定した値が Dreamweaver によって有効なペイン名として認識されない場合や、引数を省略した場合は、[環境設定] ダイアログボックスで前回アクティブだったペインが開きます。

戻り値
なし

例
次の例では、[環境設定] ダイアログボックスが開き、[コードカラーリング] カテゴリが選択されます。

```
dw.showPreferencesDialog("html colors");
```

dreamweaver.showTagChooser()

対応バージョン
Dreamweaver MX

説明
ユーザーがコードビューにタグを挿入するための [タグ選択] ダイアログボックスの表示を切り替えます。この関数を呼び出すと、[タグ選択] ダイアログボックスは Dreamweaver の他のすべてのウィンドウの上に表示されます。ダイアログボックスが表示されていない場合は、関数がダイアログボックスを開き、前面に表示してフォーカスを設定します。[タグ選択] が表示されている場合は、ダイアログボックスを非表示にします。

引数
なし

戻り値

なし

dw.registerIdleHandler()

対応バージョン

Dreamweaver CS3

説明

アイドル処理時に定期的に呼び出す JavaScript 関数を登録します。

引数

id、*idleFunction*、*interval*

- *id*: 登録するアイドルタスクの識別に使用される一意のストリング。一意性を確保するために、ID には接頭辞として固有の識別子を付けます。たとえば、5 秒ごとに警告音を鳴らす必要があるが、"beep" という名前のタスクの呼び出しは、他のユーザーが同じ名前のタスクを作成している可能性があるので行わない、という場合を考えます。この場合、"acme_beep_task" のような名前を指定すると、タスクの内容表示と名前の一意性の両方を実現できます。
- *idleFunction*: アイドル処理時に呼び出す JavaScript 関数。
- *interval*: *idleFunction* の呼び出し間隔を示す秒数。アイドル時間が発生することが前提です。

戻り値

アイドルタスクが正常に登録されたかどうかを示すブール値。

例

次の例では、5 秒ごとに 1 回、システム警告音を鳴らします。

```
dw.registerIdleHandler("acme_beep_task", function() { dw.beep();}, 5);
```

dw.revokeIdleHandler()

対応バージョン

Dreamweaver CS3

説明

registerIdleHandler() 関数によって以前に呼び出されたアイドルタスクを削除します。この関数は、以前に登録されたアイドルタスクを削除するために使用します。アプリケーションを終了するまでアイドルタスクがアクティブな状態を保持する必要がある場合は、この関数を呼び出す必要はありません。この場合、アイドルタスクはアプリケーションの終了前に自動的に削除されます。

引数

id

- *id*: 削除対象となる登録済みアイドルタスクの識別に使用される一意のストリング。この ID は、タスクの登録時に使用されたものと同じです。

戻り値

アイドルタスクが正常に削除されたかどうかを示すブール値。

例

次の例では、"dw_beep_task" という名前のアイドルタスクをアイドルタスクキューから削除します。

```
dw.revokeIdleHandler("acme_beep_task");
```

Bridge 通信関数

Bridge 通信関数を使用すると、Dreamweaver と Bridge アプリケーションとの間で通信が可能になります。この通信機能を使用して、Bridge のファイルを Dreamweaver から簡単に参照できます。

BridgeTalk.bringToFront()

対応バージョン

Dreamweaver CS3

説明

`BridgeTalk:bringToFront()` 関数を呼び出し、指定されたアプリケーションを最前面のプロセスにします。

引数

`applicationID`

`applicationID` には、アクティブにするアプリケーションを示すストリング (`bridge`、`dreamweaver` など) を指定します。

戻り値

なし

例

この例では、Dreamweaver で `browseInBridge()` 関数がどのように実装されるかを示しています。まず BridgeTalk インスタンスを作成し、次に `target` および `body` という最も重要な 2 つのプロパティを設定します。`<target>` は、ターゲットアプリケーションです。この例では、Bridge アプリケーションを指定します。識別子は `bridge` です。`<body>` は、送信するメッセージです。通常、`<body>` には、ターゲットアプリケーションで受信後に解釈および実行できるスクリプトを指定します。その後、`send()` 関数を呼び出して、`<body>` を `<target>` に送信します。

```
if (!JSBridge.isRunning('bridge'))
{
var bt = new BridgeTalk;
var scriptSavePath = browsePath.replace(/['"\\]/g, "\\$&");
var script = "app.document.thumbnail = new Thumbnail(decodeURI('" + scriptSavePath + "'));";

// スクリプトを Bridge に送信し、エラーと見なす前に 10 秒間 Bridge の起動を待機します。
bt.target = "bridge";
bt.body = script;
result = bt.send(10);
}

if (result)
BridgeTalk.bringToFront('bridge');
```

Bridgetalk.send()

対応バージョン

Dreamweaver CS3

説明

Bridge アプリケーションとの通信を確立します。

引数

`timeout`

この属性 (オプション) には、タイムアウトまでの時間を秒単位で設定します。

戻り値

Bridge アプリケーションとの通信が成功したかどうかを示すブール値 (成功の場合は `True`、失敗の場合は `False`)。

例

```
result = bridgeTalk.send(10);
```

BridgeTalk.suppressStartupScreen()

対応バージョン

Dreamweaver CS3

説明

`-nostartupscreen` という起動オプションを検索して、起動後のモーダルウィンドウを非表示にするかどうかを判別します。

戻り値

起動画面を非表示にするかどうかを示すブール値。

dw.browseInBridge()

対応バージョン

Dreamweaver CS3

説明

この関数を使用すると、Bridge のファイルを Dreamweaver から参照できます。`dw.browseInBridge()` 関数は、Bridge アプリケーションを起動します。Bridge が既に実行されている場合、`dw.browseInBridge` は Bridge アプリケーションに切り替えます。

引数

なし

戻り値

Bridge アプリケーションに、参照に使用するスクリプトが正常に送信されたかどうかを示すブール値 (成功の場合は `true`、失敗の場合は `false`)。

第 12 章 : ワークスペース

ワークスペース API 関数は、Adobe® Dreamweaver® CS3 ワークスペースのエレメントを作成または操作する関数です。これらの関数は、[ヒストリ] パネルに表示されるステップのやり直し、[挿入] バーへのオブジェクトの配置、キーボード関数による移動、メニューのリロード、スタンドアロンまたは組み込みの [結果] ウィンドウの操作、オプションの設定、ツールバーの配置、フォーカスの取得または設定などのタスクを実行します。

本章では関数をグループ分けして、それぞれを以下の項で説明します。

ヒストリ関数

ヒストリ関数は、[ヒストリ] パネルに表示されるステップの取り消し、やり直し、記録、および再生を行います。ステップとは、ドキュメントまたはドキュメント内の選択範囲に対して行う、繰り返し可能な変更のことです。`dreamweaver.historyPalette` オブジェクトのメソッドは、現在のドキュメントではなく、[ヒストリ] パネル内の選択範囲に対して制御または動作します。

dom.redo()

対応バージョン

Dreamweaver 3

説明

ドキュメントで最後に取り消したステップをやり直します。

引数

なし

戻り値

なし

イネーブラ

詳細については、413 ページの dom.canRedo() を参照してください。

dom.undo()

対応バージョン
Dreamweaver 3

説明
ドキュメントで前のステップを取り消します。

引数
なし

戻り値
なし

イネーブラ
詳細については、415 ページの dom.canUndo() を参照してください。

dreamweaver.getRedoText()

対応バージョン
Dreamweaver 3

説明
ユーザーが [編集]-[やり直し] を選択した場合、あるいは Ctrl + Y (Windows) または Command + Y (Macintosh) を押した場合にやり直す編集操作に関連付けられたテキストを取得します。

引数
なし

戻り値
やり直す編集操作に関連付けられたテキストを含むストリング。

例
選択されたテキストにユーザーが最後のアクションでボールドを適用した場合は、`dreamweaver.getRedoText()` を呼び出すと、`"Repeat Apply Bold"` という値が返されます。

dreamweaver.getUndoText()

対応バージョン
Dreamweaver 3

説明
ユーザーが [編集]-[取り消し] を選択した場合、あるいは Ctrl + Z (Windows) または Command + Z (Macintosh) を押した場合に取り消す編集操作に関連付けられたテキストを取得します。

引数
なし

戻り値
取り消す編集操作に関連付けられたテキストを含むストリング。

例

ユーザーが最後のアクションで選択範囲のテキストに CSS (Cascading Style Sheet) スタイルを適用した場合は、`dreamweaver.getUndoText()` 関数を呼び出すと、`"Undo Apply <span>"` という値が返されます。

dreamweaver.playRecordedCommand()

対応バージョン

Dreamweaver 3

説明

アクティブなドキュメントで、記録したコマンドを再生します。

引数

なし

戻り値

なし

イネーブラ

詳細については、420 ページの dreamweaver.canPlayRecordedCommand() を参照してください。

dreamweaver.redo()

対応バージョン

Dreamweaver 3

説明

アクティブなドキュメントウィンドウ、ダイアログボックス、フローティングパネル、または [サイト] パネルで、最後に取り消されたステップをやり直します。

引数

なし

戻り値

なし

イネーブラ

詳細については、421 ページの dreamweaver.canRedo() を参照してください。

dreamweaver.startRecording()

対応バージョン

Dreamweaver 3

説明

アクティブなドキュメントで、ステップの記録を開始します。前回記録されたコマンドは直ちに破棄されます。

引数

なし

戻り値
なし

イネーブラ
427 ページの dreamweaver.isRecording() を参照してください (値 `false` を返す必要があります)。

dreamweaver.stopRecording()

対応バージョン
Dreamweaver 3

説明
ユーザーに通知せずに記録を停止します。

引数
なし

戻り値
なし

イネーブラ
427 ページの dreamweaver.isRecording() を参照してください (値 `true` を返す必要があります)。

dreamweaver.undo()

対応バージョン
Dreamweaver 3

説明
フォーカスのあるドキュメントウィンドウ、ダイアログボックス、フローティングパネル、または [サイト] パネルで、前のステップを取り消します。

引数
なし

戻り値
なし

イネーブラ
詳細については、415 ページの dom.canUndo() を参照してください。

dreamweaver.historyPalette.clearSteps()

対応バージョン
Dreamweaver 3

説明
[ヒストリ] パネルのステップをすべてクリアして、[取り消し] と [やり直し] の各メニュー項目を使用不可にします。

引数
なし

戻り値
なし

dreamweaver.historyPalette.copySteps()

対応バージョン
Dreamweaver 3

説明
指定されたヒストリステップをクリップボードにコピーします。指定されたヒストリステップに繰り返し不可能なアクションが含まれる場合は、予期しない操作結果が生じる可能性を伝える警告が表示されます。

引数
arrayOfIndices

- *arrayOfIndices* 引数には、[ヒストリ] パネルでの位置を示すインデックスの配列を指定します。

戻り値
指定されたヒストリステップに対応する JavaScript が格納されたストリング。

例
次の例では、[ヒストリ] パネルの最初の 4 つのステップをコピーします。

```
dreamweaver.historyPalette.copySteps([0,1,2,3]);
```

dreamweaver.historyPalette.getSelectedSteps()

対応バージョン
Dreamweaver 3

説明
[ヒストリ] パネルのどの部分が選択されているか判別します。

引数
なし

戻り値
選択されたすべてのステップの位置を示すインデックスを格納した配列。最初の位置は 0 (ゼロ) です。

例
次の図のように [ヒストリ] パネルの 2、3、4 番目のステップが選択されている場合は、`dreamweaver.historyPalette.getSelectedSteps()` 関数を呼び出すと `[1,2,3]` が返されます。

dreamweaver.historyPalette.getStepCount()

対応バージョン
Dreamweaver 3

説明
[ヒストリ] パネルにあるステップの数を取得します。

引数
なし

戻り値
[ヒストリ] パネルに現在表示されているステップの数を表す整数。

dreamweaver.historyPalette.getStepsAsJavaScript()

対応バージョン
Dreamweaver 3

説明
指定されたヒストリステップに相当する JavaScript を取得します。

引数
arrayOfIndices

- *arrayOfIndices* 引数には、[ヒストリ] パネルでの位置を示すインデックスの配列を指定します。

戻り値
指定されたヒストリステップに対応する JavaScript が格納されたストリング。

例
次の図のように、[ヒストリ] パネルで 3 つの手順が選択されている場合に `dreamweaver.historyPalette.getStepsAsJavaScript(dw.historyPalette.getSelectedSteps())` 関数を呼び出すと、`"dw.getDocumentDOM().insertText('Hey diddle diddle, a cat and a fiddle, the cow jumped over the moon.');\n dw.getDocumentDOM().newBlock();\n dw.getDocumentDOM().insertHTML('<img src=\ "../wdw99/50browsers/images/sun.gif\">', true);\n"` が返されます。

dreamweaver.historyPalette.getUndoState()

対応バージョン
Dreamweaver 3

説明
現在の取り消し状態を取得します。

引数
なし

戻り値
[ヒストリ] パネルでのスライダの位置。

dreamweaver.historyPalette.replaySteps()

対応バージョン
Dreamweaver 3

説明
アクティブなドキュメントで、指定されたヒストリステップを再生します。指定されたヒストリステップに繰り返し不可能なアクションが含まれる場合は、予期しない操作結果が生じる可能性を伝える警告が表示されます。

引数
arrayOfIndices

- *arrayOfIndices* 引数には、[ヒストリ] パネルでの位置を示すインデックスの配列を指定します。

戻り値
指定されたヒストリステップに対応する JavaScript が格納されたストリング。

例
`dreamweaver.historyPalette.replaySteps([0,2,3])` 関数を呼び出すと、[ヒストリ] パネルの 1、3、4 番目のステップが再生されます。

dreamweaver.historyPalette.saveAsCommand()

対応バージョン
Dreamweaver 3

説明
指定されたステップをコマンドとして保存できるように、[コマンドとして保存] ダイアログボックスを開きます。ステップに繰り返し不可能なアクションが含まれる場合は、予期しない操作結果が生じる可能性を伝える警告が表示されます。

引数
arrayOfIndices

- *arrayOfIndices* 引数には、[ヒストリ] パネルでの位置を示すインデックスの配列を指定します。

戻り値
指定されたヒストリステップに対応する JavaScript が格納されたストリング。

例
次の例では、[ヒストリ] パネルの 4、6、および 8 番目のステップをコマンドとして保存します。

```
dreamweaver.historyPalette.saveAsCommand([3,5,7]);
```

dreamweaver.historyPalette.setSelectedSteps()

対応バージョン
Dreamweaver 3

説明
[ヒストリ] パネルの指定されたステップを選択します。

引数
arrayOfIndices

- *arrayOfIndices* 引数には、[ヒストリ] パネルでの位置を示すインデックスの配列を指定します。引数を指定しない場合は、すべてのステップの選択が解除されます。

戻り値
なし

例
次の例では、[ヒストリ] パネルの 1、2、3 番目のステップを選択します。

```
dreamweaver.historyPalette.setSelectedSteps([0,1,2]);
```

dreamweaver.historyPalette.setUndoState()

対応バージョン
Dreamweaver 3

説明
指定された取り消し状態に到達するまで、適切な数の取り消しまたはやり直しの操作を実行します。

引数
undoState

- *undoState* 引数は、`dreamweaver.historyPalette.getUndoState()` 関数が返すオブジェクトです。

戻り値
なし

オブジェクト挿入関数

オブジェクト挿入関数は、[挿入] バー上のオブジェクト、または [挿入] メニューに表示されるオブジェクトに関連する操作を制御します。

dom.insertFlashElement()

対応バージョン
Dreamweaver MX 2004

説明
指定された Flash エレメント (SWC ファイル) を現在のドキュメントに挿入します。この関数は、Flash エレメントが挿入バーに追加されており、"Configuration/Objects/FlashElements" フォルダまたはサブフォルダにコンポーネントファイルがあるものと見なして動作します。

引数

swcFilename

- *swcFilename* ストリングには、目的の Flash コンポーネントのパスと名前を、"Configuration/Objects/FlashElements" フォルダを基準に指定します。

戻り値

なし

例

次の例では、"Components/Objects/FlashElements/Navigation" フォルダにあるナビゲーションバー Flash コンポーネントを、現在のドキュメントに挿入しています。

```
dom.insertFlashElement("¥Navigation¥navBar.swc");
```

dreamweaver.objectPalette.getMenuDefault()

対応バージョン

Dreamweaver MX 2004

説明

関連付けられたメニューのデフォルト項目の ID ストリングを取得します。

引数

menuId

- *menuId* 引数には、"insertbar.xml" ファイル内のメニューを表すストリングを定義します。

戻り値

デフォルト項目の ID を定義するストリング値。

例

次の例では、[メディア] メニューの現在のデフォルトオブジェクトを、`defID` 変数に割り当てています。

```
var defId = dw.objectPalette.getMenuDefault("DW_Media");
```

dreamweaver.objectPalette.setMenuDefault()

対応バージョン

Dreamweaver MX 2004

説明

ポップアップメニューのデフォルトオブジェクトを設定します。デフォルトオブジェクトのアイコンは、[挿入] バー上の指定されたポップアップメニューを表します。ユーザーは、デフォルトオブジェクトをクリックして挿入したり、デフォルトオブジェクトの横にある矢印をクリックしてポップアップメニューを開き、そのメニューに含まれる他のオブジェクトを表示することができます。ユーザーが次に Dreamweaver を開くか、[拡張機能のリロード] コマンドを使用すると、メニューの新しいデフォルト値が設定されます。

引数

menuId、*defaultId*

- *menuId* 引数には、"insertbar.xml" ファイル内のメニューを表すストリングを定義します。
- *defaultId* 引数には、"insertbar.xml" ファイルの新しいデフォルトオブジェクトを表すストリングを定義します。

戻り値

ブール値。新しいデフォルト値の設定に成功した場合は `true`、失敗した場合は `false`。

例

次の例では、Flash オブジェクトを [メディア] メニューのデフォルトオブジェクトとして設定しています。

```
dw.objectPalette.setMenuDefault("DW_Media", "DW_Flash");
```

dreamweaver.reloadObjects()

対応バージョン

Dreamweaver MX 2004

説明

[挿入] バーのすべてのオブジェクトをリロードします。この関数は、[挿入] バーのカテゴリで Ctrl キーを押しながらクリックし、[拡張機能のリロード] メニューオプションを選択したときと同じ操作を実行します。

引数

なし

戻り値

ブール値。オブジェクトのロードに成功した場合は `true`、失敗した場合は `false`。

dom.convertActiveContent()

対応バージョン

Dreamweaver CS3

説明

指定されたドキュメント内のすべてのアクティブコンテンツを変換します。

引数

`forceUpdate`

- *`forceUpdate`* 引数には、ユーザーの環境設定を上書きするかどうかを示すブール値を指定します。上書きする場合は `true` です。この引数はオプションです。

戻り値

ブール値。すべてのアクティブコンテンツが正常に変換された場合は `true`。テンプレートインスタンス内のロックされた領域にあるオブジェクトタグなど、変換の必要なアクティブコンテンツの一部が正常に変換されなかった場合は `false`。

例

```
if( !dom.convertActiveContent(true) ) {
alert(dw.loadString("ActiveContent/notAllConverted"));
}
```

dom.convertNextActiveContent()

対応バージョン

Dreamweaver CS3

説明

現在の編集のリマインダー (元に戻すことのできる 1 つのアクション) に対して挿入される次の object タグに、スクリプトが組み込まれていることを指定します。この関数を使用すると、特定のアクティブコンテンツに対して、サードパーティの拡張機能を使用して適切なスクリプトを生成することができます。

引数

なし

戻り値

なし

例

```
dom.convertNextActiveContent();
dom.insertHTML("<object classid=\"clsid:D27CDB6E-AE6D-11cf-96B8-444553540000\" codebase=\
"http://download.Macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,19,0\"
width=\"100\" height=\"22\"><param name=\"movie\" value=\"button1.swf\" /><param name=\
"quality\" value=\"high\" /><embed src=\"button1.swf\" quality=\"high\" pluginspage=\
"http://www.Macromedia.com/go/getflashplayer\" type=\"application/
x-shockwave-flash\"width=\"100\" height=\"22\"></embed></object>\");
```

キーボード関数

キーボード関数は、矢印、BackSpace、Delete、PageUp、および PageDown の各キーを押した場合に実行するドキュメントナビゲーションタスクに相当する操作を実行します。Dreamweaver には、`arrowLeft()` や `backspaceKey()` などの一般的な矢印やキーの関数の他にも、次の単語、前の単語、次の段落、前の段落、行の先頭、行の終わり、ドキュメントの先頭、ドキュメントの終わりに移動するメソッドが用意されています。

dom.arrowDown()

対応バージョン

Dreamweaver 3

説明

挿入ポイントを指定した回数だけ下に移動します。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数には、挿入ポイントを下に移動する回数を指定します。この引数を指定しない場合のデフォルト値は、1 です。
- *bShiftIsDown* 引数には、選択範囲を拡張するかどうかを示すブール値を指定します。この引数を指定しない場合、`false` に初期設定されます。

戻り値

なし

dom.arrowLeft()

対応バージョン

Dreamweaver 3

説明

挿入ポイントを指定した回数だけ左に移動します。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを左に移動する回数を指定します。この引数を指定しない場合のデフォルト値は、1 です。
- *bShiftIsDown* 引数 (オプション) には、選択範囲を拡張するかどうかを示すブール値を指定します。この引数を指定しない場合、`false` に初期設定されます。

戻り値

なし

dom.arrowRight()

対応バージョン

Dreamweaver 3

説明

挿入ポイントを指定した回数だけ右に移動します。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを右に移動する回数を指定します。この引数を指定しない場合のデフォルト値は、1 です。
- *bShiftIsDown* 引数 (オプション) には、選択範囲を拡張するかどうかを示すブール値を指定します。この引数を指定しない場合、`false` に初期設定されます。

戻り値

なし

dom.arrowUp()

対応バージョン

Dreamweaver 3

説明

挿入ポイントを指定した回数だけ上に移動します。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを上に移動する回数を指定します。この引数を指定しない場合のデフォルト値は、1 です。
- *bShiftIsDown* 引数 (オプション) には、選択範囲を拡張するかどうかを示すブール値を指定します。この引数を指定しない場合、`false` に初期設定されます。

戻り値

なし

dom.backspaceKey()

対応バージョン
Dreamweaver 3

説明
この関数は、指定した回数だけ BackSpace キーを押したときと同じ操作を実行します。実際の動作は、選択範囲があるか、それとも挿入ポイントのみであるかによって異なります。

引数
{nTimes}

- *nTimes* 引数 (オプション) には、BackSpace キーと同じ操作を実行する回数を指定します。この引数を指定しない場合、1 に初期設定されます。

戻り値
なし

dom.deleteKey()

対応バージョン
Dreamweaver 3

説明
この関数は、指定した回数だけ Delete キーを押したときと同じ操作を実行します。実際の動作は、選択範囲があるか、それとも挿入ポイントのみであるかによって異なります。

引数
{nTimes}

- *nTimes* 引数 (オプション) には、Delete キーと同じ操作を実行する回数を指定します。この引数を指定しない場合、1 に初期設定されます。

戻り値
なし

dom.endOfDocument()

対応バージョン
Dreamweaver 3

説明
挿入ポイントをドキュメントの最後に移動します。ドキュメントの最後とは、ドキュメントウィンドウにフォーカスがある場合は最後の表示可能なコンテンツの後、コードインスペクタにフォーカスがある場合は終了 HTML タグの後のことです。

引数
{bShiftIsDown}

- *bShiftIsDown* 引数 (オプション) には、選択範囲を拡張するかどうかを示すブール値を指定します。この引数を指定しない場合、false に初期設定されます。

戻り値
なし

dom.endOfLine()

対応バージョン
Dreamweaver 3

説明
挿入ポイントを行の末尾に移動します。

引数
{bShiftIsDown}

- *bShiftIsDown* 引数 (オプション) には、選択範囲を拡張するかどうかを示すブール値を指定します。この引数を指定しない場合、`false` に初期設定されます。

戻り値
なし

dom.nextParagraph()

対応バージョン
Dreamweaver 3

説明
挿入ポイントを次の段落の先頭に移動するか、または *nTimes* が 1 より大きい場合は複数の段落をスキップします。

引数
{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを下に移動する段落数を指定します。この引数を指定しない場合のデフォルト値は、1 です。
- *bShiftIsDown* 引数には、選択範囲を拡張するかどうかを示すブール値を指定します。この引数を指定しない場合、`false` に初期設定されます。

戻り値
なし

dom.nextWord()

対応バージョン
Dreamweaver 3

説明
挿入ポイントを次の単語の先頭に移動するか、または *nTimes* が 1 より大きい場合は複数の単語をスキップします。

引数
{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを前方に移動する単語数を指定します。この引数を指定しない場合のデフォルト値は、1 です。
- *bShiftIsDown* 引数 (オプション) には、選択範囲を拡張するかどうかを示すブール値を指定します。この引数を指定しない場合、`false` に初期設定されます。

戻り値

なし

dom.pageDown()

対応バージョン

Dreamweaver 3

説明

挿入ポイントを 1 ページ下に移動します。これは、PageDown キーを押すのと同じ操作です。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを何ページ下に移動するかを指定します。この引数を指定しない場合のデフォルト値は、1 です。
- *bShiftIsDown* 引数 (オプション) には、選択範囲を拡張するかどうかを示すブール値を指定します。この引数を指定しない場合、`false` に初期設定されます。

戻り値

なし

dom.pageUp()

対応バージョン

Dreamweaver 3

説明

挿入ポイントを 1 ページ上に移動します。これは、PageUp キーを押すのと同じ操作です。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを何ページ上に移動するかを指定します。この引数を指定しない場合のデフォルト値は、1 です。
- *bShiftIsDown* 引数 (オプション) には、選択範囲を拡張するかどうかを示すブール値を指定します。この引数を指定しない場合、`false` に初期設定されます。

戻り値

なし

dom.previousParagraph()

対応バージョン

Dreamweaver 3

説明

挿入ポイントを前の段落の先頭に移動するか、または *nTimes* が 1 より大きい場合は複数の段落を上方向にスキップします。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを上に移動する段落数を指定します。この引数を指定しない場合のデフォルト値は、1 です。
- *bShiftIsDown* 引数 (オプション) には、選択範囲を拡張するかどうかを示すブール値を指定します。この引数を指定しない場合、`false` に初期設定されます。

戻り値

なし

dom.previousWord()

対応バージョン

Dreamweaver 3

説明

挿入ポイントを前の単語の先頭に移動するか、または *nTimes* が 1 より大きい場合は複数の単語を上方向にスキップします。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを逆方向に移動する単語数を指定します。この引数を指定しない場合のデフォルト値は、1 です。
- *bShiftIsDown* 引数 (オプション) には、選択範囲を拡張するかどうかを示すブール値を指定します。この引数を指定しない場合、`false` に初期設定されます。

戻り値

なし

dom.startOfDocument()

対応バージョン

Dreamweaver 3

説明

挿入ポイントをドキュメントの先頭に移動します。ドキュメントの先頭とは、ドキュメントウィンドウにフォーカスがある場合は最初の表示可能なコンテンツの前、コードインスペクタにフォーカスがある場合は開始 HTML タグの前のことです。

引数

{bShiftIsDown}

- *bShiftIsDown* 引数 (オプション) には、選択範囲を拡張するかどうかを示すブール値を指定します。この引数を指定しない場合、`false` に初期設定されます。

戻り値

なし

dom.startOfLine()

対応バージョン
Dreamweaver 3

説明
挿入ポイントを行の先頭に移動します。

引数
{bShiftIsDown}

- *bShiftIsDown* 引数 (オプション) には、選択範囲を拡張するかどうかを示すブール値を指定します。この引数を指定しない場合、false に初期設定されます。

戻り値
なし

dreamweaver.mapKeyCodeToChar()

対応バージョン
Dreamweaver 4

説明
イベントオブジェクトの keyCode フィールドから取得したキーコードをそのまま取得し、文字に変換します。取得したキーコードが、Home キー、PageUp キーなどの特殊キーかどうかを確認する必要があります。キーコードが特殊キーでない場合は、このメソッドを使用して、そのキーコードを表示に適した文字コードに変換することができます。

引数
keyCode

- *keyCode* 引数には、文字に変換するキーコードを指定します。

戻り値
マッピングが成功した場合は変換した文字コード。それ以外の場合は 0 (ゼロ)。

メニュー関数

メニュー関数は、Dreamweaver のメニューの最適化とリロードを制御します。dreamweaver.getMenuNeedsUpdating() 関数と dreamweaver.notifyMenuUpdated() 関数は、Dreamweaver に組み込まれた動的メニューに対して不要な更新ルーチンが実行されるのを避けるように設計されています。詳細については、122 ページの dreamweaver.getMenuNeedsUpdating() および 123 ページの dreamweaver.notifyMenuUpdated() を参照してください。

dreamweaver.getMenuNeedsUpdating()

対応バージョン
Dreamweaver 3

説明
指定したメニューを更新する必要があるかどうかをチェックします。

引数

menuId

- *menuId* 引数には、"menus.xml" ファイルで指定したメニュー項目の id 属性の値を含むストリングを指定します。

戻り値

メニューを更新する必要があるかどうかを示すブール値。この関数は、dreamweaver.notifyMenuUpdated() がこの *menuId* を指定して呼び出され、*menuListFunction* の戻り値がそれ以降変更されていない場合に限り、false を返します。詳細については、123 ページの dreamweaver.notifyMenuUpdated() を参照してください。

dreamweaver.notifyMenuUpdated()

対応バージョン

Dreamweaver 3

説明

指定したメニューを更新する必要がある場合に Dreamweaver に通知します。

引数

menuId、*menuListFunction*

- *menuId* 引数には、"menus.xml" ファイルで指定したメニュー項目の id 属性の値を含むストリングを指定します。
- *menuListFunction* 引数は、"dw.cssStylePalette.getStyles()"、"dw.getDocumentDOM().getFrameNames()"、"dw.getDocumentDOM().getEditableRegionList"、"dw.getBrowserList()"、"dw.getRecentFileList()"、"dw.getTranslatorList()"、"dw.getFontList()"、"dw.getDocumentList()"、"dw.htmlStylePalette.getStyles()"、または "site.getSites()" のいずれかである必要があります。

戻り値

なし

dreamweaver.reloadMenus()

対応バージョン

Dreamweaver 3

説明

"Configuration" フォルダ内の "menus.xml" ファイルからメニュー構造全体をリロードします。

引数

なし

戻り値

なし

結果ウィンドウ関数

結果ウィンドウ関数を使用すると、[結果] パネルグループの組み込みパネルを対話式に操作したり、フォーマットされたデータの列を表示するスタンドアロン形式のウィンドウを作成することができます。

組み込み済みの [結果] パネルグループの操作

以下の関数は、[結果] パネルグループ内に出力を生成します。[結果] パネルグループには、検索、ソースバリデート、サイト間レポート、ブラウザ互換性チェック、サーバーデバッグ、FTP ログ、およびリンクチェックに使用するタブ付きのレポートが表示されます。

特定の子パネルの操作

以下の子パネルは、Dreamweaver インターフェイスに常に表示され、直接アクセスできる、組み込み [結果] ウィンドウです。

- `dreamweaver.resultsPalette.siteReports`
- `dreamweaver.resultsPalette.validator`
- `dreamweaver.resultsPalette.bcc`

これらのパネルも [結果] ウィンドウなので、スタンドアロン形式の [結果] ウィンドウに定義されている次のメソッドを使用できます。

- `getItem()`
- `getItemCount()`
- `getSelectedItem()`
- `setSelectedItem()`

`resWin` メソッドの使用の詳細については、128 ページのスタンドアローン形式の [結果] ウィンドウの作成を参照してください。

アクティブな子パネルの操作

以下の一般的な API 関数は、アクティブな子パネルすべてに適用されます。子パネルの中には、一部の関数を無視するものもあります。アクティブな子パネルで関数がサポートされていない場合は、その関数を呼び出しても何も実行されません。

dreamweaver.showResults()

対応バージョン

Dreamweaver MX 2004

説明

指定された [結果] フローティングパネルを開き、項目を選択します。

注意 : この関数は、[結果] パネルグループの [バリデータ]、[ブラウザ互換性チェック]、および [サイトレポート] の各パネルだけでサポートされています。

引数

`floaterName`、*`floaterIndex`*

- *`floaterName`* 引数には、開く [結果] フローティングパネルを表すストリングを指定します。有効な値は、`'validation'`、または `'reports'` です。
- *`floaterIndex`* 引数は、数値またはストリングです。[結果] パネルで選択する項目のインデックスを指定するには、数値を使用します。ドキュメントの URL を指定するには、ストリングを使用します。URL を指定すると、そのドキュメントで表示可能な最初の項目が選択されます。

戻り値

なし

例

次の例では、ドキュメント内の現在の選択範囲のオフセット位置でエラーをチェックし、エラーがある場合は、[結果] パネルの指定されたウィンドウ (floaterName) に表示します。それ以外の場合は、[結果] パネルの [ブラウザ互換性チェック] ウィンドウを開き、現在のドキュメントで表示可能な最初の項目を表示します。

```
var offset = dw.getDocumentDOM().source.getSelection()[0];
var errors = dw.getDocumentDOM().source.getValidationErrorsForOffset(offset);
if ( errors && errors.length > 0 )
    dw.showResults( errors[0].floaterName, errors[0].floaterIndex );
else
    dw.showResults('bcc', dw.getDocumentDOM().URL);
```

dreamweaver.resultsPalette.siteReports.addResultItem()

対応バージョン

Dreamweaver 4

説明

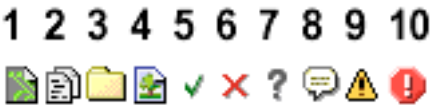
processfile() 関数が処理するファイル内の情報に基づいて、[サイトレポート] パネルに新しい結果エントリを追加します。

この関数は、サイトレポートの processFile() コールバックだけで使用できます。サイトレポートの詳細については、『Dreamweaver 拡張ガイド』の「レポート」を参照してください。

引数

strFilePath, strIcon, strDisplay, strDesc, {iLineNo}, {iStartSel}, {iEndSel}

- *strFilePath* 引数には、処理するファイルの完全修飾 URL パスを指定します。
- *strIcon* 引数には、使用するアイコンのパスを指定します。組み込み済みのアイコンを表示するには、アイコンの完全修飾パスを指定する代わりに、値 1 ～ 10 を使用します。アイコンを表示しない場合は、0 を指定します。次の表は、値 1 ～ 10 に相当するアイコンを示します。



- *strDisplay* 引数には、[結果] ウィンドウの 1 列目に表示するストリングを指定します。通常はファイル名になります。
- *strDesc* 引数には、エントリに付随する説明を指定します。
- *iLineNo* 引数には、ファイル内の行数を指定します (オプション)。
- *iStartSel* 引数には、ファイルへのオフセットの開始点を指定します。これはオプションですが、この引数を使用する場合は、*iEndSel* 引数も指定する必要があります。
- *iEndSel* 引数には、ファイルへのオフセットの終了点を指定します。この引数は、*iStartSel* を使用した場合に必須となります。

戻り値

なし

dreamweaver.resultsPalette.clear()

対応バージョン
Dreamweaver MX

説明
フォーカスのあるパネルのコンテンツをクリアします。

引数
なし

戻り値
なし

イネーブラ
詳細については、428 ページの dreamweaver.resultsPalette.canClear() を参照してください。

dreamweaver.resultsPalette.Copy()

対応バージョン
Dreamweaver MX

説明
コピーされたメッセージを、フォーカスのあるウィンドウに送ります。この関数は、一般に FTP ログウィンドウに使用します。

引数
なし

戻り値
なし

イネーブラ
詳細については、428 ページの dreamweaver.resultsPalette.canCopy() を参照してください。

dreamweaver.resultsPalette.cut()

対応バージョン
Dreamweaver MX

説明
切り取られたメッセージを、フォーカスのあるウィンドウに送ります。この関数は、一般に FTP ログウィンドウに使用します。

引数
なし

戻り値
なし

イネーブラ
詳細については、428 ページの dreamweaver.resultsPalette.canCut() を参照してください。

dreamweaver.resultsPalette.Paste()

対応バージョン
Dreamweaver MX

説明
ペーストされたメッセージを、フォーカスのあるウィンドウに送ります。この関数は、一般に FTP ログウィンドウに使用します。

引数
なし

戻り値
なし

イネーブラ
詳細については、428 ページの dreamweaver.resultsPalette.canPaste() を参照してください。

dreamweaver.resultsPalette.openInBrowser

対応バージョン
Dreamweaver MX

説明
サイトレポート、ブラウザターゲットチェック、バリデート、およびリンクチェックのレポートを、デフォルトのブラウザに送ります。

引数
なし

戻り値
なし

イネーブラ
詳細については、429 ページの dreamweaver.resultsPalette.canOpenInBrowser() を参照してください。

dreamweaver.resultsPalette.openInEditor()

対応バージョン
Dreamweaver MX

説明
特定のレポート (サイトレポート、ブラウザターゲットチェック、バリデート、およびリンクチェック) の選択行にジャンプして、エディタにドキュメントを開きます。

引数
なし

戻り値
なし

イネーブラ

詳細については、429 ページの dreamweaver.resultsPalette.canOpenInEditor() を参照してください。

dreamweaver.resultsPalette.save()

対応バージョン

Dreamweaver MX

説明

Save 関数 (サイトレポート、ブラウザターゲットチェック、バリデート、およびリンクチェック) をサポートするウィンドウに、[保存] ダイアログボックスを開きます。

引数

なし

戻り値

なし

イネーブラ

詳細については、429 ページの dreamweaver.resultsPalette.canSave() を参照してください。

dreamweaver.resultsPalette.selectAll()

対応バージョン

Dreamweaver MX

説明

[すべて選択] コマンドをフォーカスのあるウィンドウに送ります。

引数

なし

戻り値

なし

イネーブラ

詳細については、430 ページの dreamweaver.resultsPalette.canSelectAll() を参照してください。

スタンドアローン形式の [結果] ウィンドウの作成

`dreamweaver.createResultsWindow()` 関数を使用して、[結果] ウィンドウを作成します。

dreamweaver.createResultsWindow()

対応バージョン

Dreamweaver 4

説明

新規の [結果] ウィンドウを作成して、このウィンドウに JavaScript オブジェクトリファレンスを返します。

引数

strName、*arrColumns*

- *strName* 引数には、ウィンドウのタイトルに使用するストリングを指定します。
- *arrColumns* 引数には、リストコントロールで使用する列名の配列を指定します。

戻り値

作成したウィンドウにオブジェクトリファレンスが返されます。

resWin.addItem()

対応バージョン

Dreamweaver 4

説明

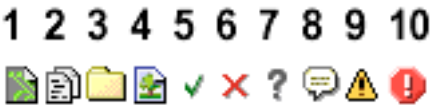
[結果] ウィンドウに新規項目を追加します。

注意 : 128 ページの dreamweaver.createResultsWindow() で作成したスタンドアローン形式の [結果] ウィンドウにのみ使用します。[バリデータ]、[ブラウザ互換性チェック]、[サイトレポート] などの組み込み [結果] ウィンドウには、*resWin.addItem()* 関数は使用できません。

引数

resultWindowObj、*strIcon*、*strDesc*、*itemData*、*iStartSel*、*iEndSel*、*colNdata*

- *resultWindowObj* 引数には、createResultsWindow() 関数が返すオブジェクトを指定します。
- *strIcon* 引数には、使用するアイコンのパスを表すストリングを指定します。組み込み済みのアイコンを表示するには、アイコンの完全修飾パス名を指定する代わりに、値 1 ～ 10 を使用します。アイコンを表示しない場合は、0 (ゼロ) を指定します。次の表は、値 1 ～ 10 に相当するアイコンを示します。



- *strDesc* 引数には、項目の詳細説明を指定します。説明がない場合は、0 を指定します。
- *itemData* 引数は、ドキュメントの行番号など、追加する項目固有のデータを格納するストリングとして使用できます。
- *iStartSel* 引数には、ファイル内の選択範囲のオフセットの開始点を指定します。オフセットを使用しない場合は、値 null を指定します。
- *iEndSel* 引数には、ファイル内の選択範囲のオフセットの終了点を指定します。オフセットを使用しない場合は、値 null を指定します。
- *colNdata* 引数は、各列のデータを表すストリングの配列です。たとえば、列が 3 つある場合は 3 つのストリングから成る配列を指定します。

戻り値

ブール値。項目の追加に成功した場合は true、失敗した場合は false。

例

次の例では、"Frodo"、"Sam"、および "Gollum" の列見出しがある resWin という [結果] ウィンドウを作成しています。resWin.addItem() 関数を呼び出すと、フォルダアイコンが追加され、ウィンドウに対して定義された 3 つの列に、msg1、msg2、および msg3 の 3 つのストリングが追加されます。

```
var resWin = dw.createResultsWindow("Test Window", ["Frodo", "Sam", "Gollum"]);
resWin.addItem(resWin, "3", "Description", null, null, null, ["msg1", "msg2", "msg3"]);
```

resWin.getItem()

対応バージョン
Dreamweaver 4

説明
項目を追加したコマンドの名前、および `addItem()` 関数に渡されたストリングと同じストリングを含む、項目の配列を取得します。

引数
itemIndex

- *itemIndex* 引数は、データが取得される項目のインデックスです。

戻り値
ストリングの配列。配列の最初の要素は項目を追加したコマンドの名前です。残りの要素は、`addItem()` 関数に渡されたものと同じストリングです。

resWin.getItemCount()

対応バージョン
Dreamweaver 4

説明
リスト内の項目の数を取得します。

引数
なし

戻り値
リスト内の項目の数。

resWin.getSelectedItem()

対応バージョン
Dreamweaver 4

説明
選択されている項目のインデックスを取得します。

引数
なし

戻り値
現在選択されている項目のインデックス。

resWin.setButtons()

対応バージョン
Dreamweaver 4

説明

arrButtons 引数で指定されたボタンを設定します。

引数

cmdDoc、*arrButtons*

- *cmdDoc* 引数は、関数を呼び出しているコマンドを表すドキュメントオブジェクトです。コマンドではキーワード *this* を使用する必要があります。
- *arrButtons* 引数は、ボタンテキストおよびボタンがクリックされたときに実行する JavaScript コードに対応するストリングの配列です。これは、commandButtons() 関数のコマンドに対する動作方法と同様です。ウィンドウに設定できるボタンは 2 つだけです。

戻り値

なし

resWin.setCallbackCommands()

対応バージョン

Dreamweaver 4

説明

[結果] ウィンドウに、processFile() メソッドを呼び出すコマンドを伝えます。この関数が呼び出されない場合、[結果] ウィンドウで作成されたコマンドが呼び出されます。

引数

arrCmdNames

- *arrCmdNames* 引数には、processFile() 関数を呼び出すコマンド名の配列を指定します。

戻り値

なし

resWin.setColumnWidths()

対応バージョン

Dreamweaver 4

説明

各列の幅を設定します。

引数

arrWidth

- *arrWidth* 引数には、コントロール内の各列の幅を表す整数の配列を指定します。

戻り値

なし

resWin.setFileList()

対応バージョン

Dreamweaver 4

説明

[結果] ウィンドウに、ファイルのリスト、フォルダのリスト、またはその両方を与えて、処理する一連のコマンドを呼び出します。

引数

arrFilePaths、*bRecursive*

- *arrFilePaths* 引数には、反復するファイルまたはフォルダのパスの配列を指定します。
- *bRecursive* 引数には、反復を再帰的にする必要がある (`true`) か、ない (`false`) かを示すブール値を指定します。

戻り値

なし

resWin.setSelectedItem()

対応バージョン

Dreamweaver 4

説明

選択されている項目を、*itemIndex* で指定された項目に設定します。

引数

itemIndex

- 選択するリスト内の項目のインデックス。

戻り値

前に選択されていた項目のインデックス。

resWin.setTitle()

対応バージョン

Dreamweaver 4

説明

ウィンドウのタイトルを設定します。

引数

strTitle

- *strTitle* 引数には、フローティングウィンドウの新規の名前を指定します。

戻り値

なし

resWin.startProcessing()

対応バージョン

Dreamweaver 4

説明

ファイル処理を開始します。

引数
なし

戻り値
なし

resWin.stopProcessing()

対応バージョン
Dreamweaver 4

説明
ファイル処理を停止します。

引数
なし

戻り値
なし

サーバーのデバッグ
Dreamweaver では、ColdFusion のファイルを要求して、その応答を組み込みブラウザに表示することができます。サーバーから応答が返されると、その応答の中から、認識済みの署名が入った XML のパケットが検索されます。検出された署名入りの XML は Dreamweaver で処理され、内容がツリーコントロールに表示されます。このツリーには、以下の項目に関する情報が表示されます。

- レンダリングされた CFM ページの生成に使用するすべてのテンプレート、カスタムタグ、およびインクルードファイル
- 例外
- SQL クエリー
- オブジェクトクエリー
- 変数
- トレース記録

さらに、[サーバーデバッグ] パネルには、他のサーバーモデルのデバッグデータも表示できます。他のサーバーモデルをデバッグするように Dreamweaver を設定するには、`dreamweaver.resultsPalette.debugWindow.addDebugContextData()` 関数を使用します。

dreamweaver.resultsPalette.debugWindow.addDebugContextData()

対応バージョン
Dreamweaver MX

説明
[サイト定義] ダイアログボックスで指定されているサーバーから返された、カスタマイズ XML ファイルを解釈します。XML ファイルのコンテンツは、[サーバーデバッグ] パネルにツリーデータとして表示されるため、このパネルを使用して、さまざまなサーバーモデルで生成されたコンテンツを評価できます。

引数

treedata

- *treedata* 引数には、サーバーから返される XML ストリングを指定します。XML ストリングには、以下のフォーマットを使用する必要があります。

server debug node	デバッグ XML データのルートノード
debugnode	全ノードに対応
context	コンテキストリストに表示される項目名
icon	ツリーノードに使用するアイコン
name	表示する名前
value	表示する値
timestamp	コンテキストノードのみに適用可能
オプションのストリング	
jumptoline	特定の行番号へのリンク
template	URL の一部であるテンプレートファイルの名前
path	サーバーからのファイルの相対パス
line number	ファイル内の行番号
start position	行内の開始文字オフセット
end position	行内の終了文字オフセット

以下に例を挙げます。

```
<serverdebuginfo>
    <context>
        <template><![CDATA[/ooo/master.cfm]]></template>
        <path><![CDATA[C:¥server¥wwwroot¥ooo¥master.cfm]]></path>
        <timestamp><![CDATA[0:0:0.0]]></timestamp>
    </context>
    <debugnode>
        <name><![CDATA[CGI]]></name>
        <icon><![CDATA[ServerDebugOutput/ColdFusion/CGIVariables.gif]]></icon>
        <debugnode>
            <name><![CDATA[Pubs.name.sourceURL]]></name>
            <icon><![CDATA[ServerDebugOutput/ColdFusion/Variable.gif]]></icon>
            <value><![CDATA[jdbc:Macromedia:sqlserver:
                //name.Macromedia.com:1111;databaseName=Pubs]]></value>
        </debugnode>
    </debugnode>
    <debugnode>
        <name><![CDATA[Element Snippet is undefined in class
coldfusion.compiler.TagInfoNotFoundException]]></name>
        <icon><![CDATA[ServerDebugOutput/ColdFusion/Exception.gif]]></icon>
        <jumptoline linenumber="3" startposition="2" endposition="20">
            <template><![CDATA[/ooo/master.cfm]]></template>
            <path><![CDATA[C:¥Neo¥wwwroot¥ooo¥master.cfm]]></path>
        </jumptoline>
    </debugnode>
</serverdebuginfo>
```

戻り値

なし

オン / オフ関数

オン / オフ関数は、さまざまなオプションのオンまたはオフの状態を取得および設定します。

dom.getEditNoFramesContent()

対応バージョン
Dreamweaver 3

説明
この関数は、[修正]-[フレームセット]-[フレームなしコンテンツの編集] オプションの現在の状態を取得します。

引数
なし

戻り値
ブール値。`true` は `NOFRAMES` コンテンツがアクティブなビューであることを示し、`false` はそうでないことを示します。

dom.getHideAllVisualAids()

対応バージョン
Dreamweaver 4

説明
この関数は、ビジュアルエイドが非表示に設定されているかどうかを判別します。

引数
なし

戻り値
ブール値。[すべてのビジュアルエイドを非表示] がオンの場合は `true`、オフの場合は `false`。

dom.getPreventLayerOverlaps()

対応バージョン
Dreamweaver 3

説明
この関数は、[レイヤーのオーバーラップ防止] オプションの現在の状態を取得します。

引数
なし

戻り値
ブール値。[レイヤーのオーバーラップ防止] オプションがオンの場合は `true`、オフの場合は `false`。

dom.getShowAutoIndent()

対応バージョン
Dreamweaver 4

説明
この関数は、ドキュメントウィンドウのコードビューで、自動インデントがオンになっているかどうかを判別します。

引数
なし

戻り値
ブール値。自動インデントがオンになっている場合は `true`、オフになっている場合は `false`。

dom.getShowFrameBorders()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[フレームボーダー] オプションの現在の状態を取得します。

引数
なし

戻り値
ブール値。`true` はフレームボーダーの表示がオンであることを示し、`false` はオフであることを示します。

dom.getShowGrid()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[グリッド]-[表示] オプションの現在の状態を取得します。

引数
なし

戻り値
ブール値。`true` はグリッドの表示がオンであることを示し、`false` はオフであることを示します。

dom.getShowHeadView()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[ヘッドコンテンツ] オプションの現在の状態を取得します。

引数
なし

戻り値
ブール値。true はヘッドコンテンツの表示がオンであることを示し、false はオフであることを示します。

dom.getShowInvalidHTML()

対応バージョン
Dreamweaver 4

説明
この関数は、ドキュメントウィンドウのコードビューで、無効な HTML コードが現在ハイライト表示されているかどうかを判別します。

引数
なし

戻り値
ブール値。無効な HTML コードがハイライト表示されている場合は true、そうでない場合は false。

dom.getShowImageMaps()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[イメージマップ] オプションの現在の状態を取得します。

引数
なし

戻り値
ブール値。true はイメージマップの表示がオンであることを示し、false はオフであることを示します。

dom.getShowInvisibleElements()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[不可視エレメント] オプションの現在の状態を取得します。

引数
なし

戻り値
ブール値。true は不可視エレメントのマーカーの表示がオンであることを示し、false はオフであることを示します。

dom.getShowLayerBorders()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[レイヤーボーダー] オプションの現在の状態を取得します。

引数
なし

戻り値
ブール値。`true` はレイヤーボーダーの表示がオンであることを示し、`false` はオフであることを示します。

dom.getShowLineNumbers()

対応バージョン
Dreamweaver 4

説明
この関数は、コードビューで行番号が表示されているかどうかを判別します。

引数
なし

戻り値
ブール値。`true` は行番号の表示がオンであることを示し、`false` はオフであることを示します。

dom.getShowRulers()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[ルーラ]-[表示] オプションの現在の状態を取得します。

引数
なし

戻り値
ブール値。`true` はルーラの表示がオンであることを示し、`false` はオフであることを示します。

dom.getShowSyntaxColoring()

対応バージョン
Dreamweaver 4

説明
この関数は、ドキュメントウィンドウのコードビューで、シンタックスカラーリングがオンになっているかどうかを判別します。

引数
なし

戻り値
ブール値。シンタックスカラーリングがオンになっている場合は `true`、オフになっている場合は `false`。

dom.getShowTableBorders()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[テーブルボーダー] オプションの現在の状態を取得します。

引数
なし

戻り値
ブール値。`true` はテーブルボーダーの表示がオンであることを示し、`false` はオフであることを示します。

dom.getShowToolbar()

対応バージョン
Dreamweaver 4

説明
この関数は、ツールバーが表示されるかどうかを判別します。

引数
なし

戻り値
ブール値。ツールバーが表示される場合は `true`、そうでない場合は `false`。

dom.getShowTracingImage()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[トレーシングイメージ]-[表示] オプションの現在の状態を取得します。

引数
なし

戻り値
ブール値。`true` はオプションがオンであることを示し、`false` はオフであることを示します。

dom.getShowWordWrap()

対応バージョン
Dreamweaver 4

説明
この関数は、ドキュメントウィンドウのコードビューで、テキストの折り返し (ワードラップ) がオンになっているかどうかを判別します。

引数
なし

戻り値
ブール値。テキストの折り返し (ワードラップ) がオンになっている場合は `true`、オフになっている場合は `false`。

dom.getSnapToGrid()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[グリッド]-[吸着] オプションの現在の状態を取得します。

引数
なし

戻り値
ブール値。`true` はグリッド吸着オプションがオンであることを示し、`false` はオフであることを示します。

dom.setEditNoFramesContent()

対応バージョン
Dreamweaver 3

説明
この関数は、[修正]-[フレームセット]-[フレームなしコンテンツの編集] オプションのオンとオフを切り替えます。

引数
`bEditNoFrames`

- ブール値。*`bEditNoFrames`* 引数はブール値です。`true` では [フレームなしコンテンツの編集] オプションがオンになり、`false` ではオフになります。

戻り値
なし

イネーブラ
詳細については、410 ページの dom.canEditNoFramesContent() を参照してください。

dom.setHideAllVisualAids()

対応バージョン
Dreamweaver 4

説明
この関数は、すべてのボーダー、イメージマップ、および不可視エレメントの表示を、[表示] メニューにおける個々の設定にかかわらず、オフにします。

引数
bSet

- *bSet* 引数はブール値です。true ではビジュアルエイドの表示がオフになり、false ではオンになります。

戻り値
なし

dom.setPreventLayerOverlaps()

対応バージョン
Dreamweaver 3

説明
この関数は、[レイヤーのオーバーラップ防止] オプションのオンとオフを切り替えます。

引数
bPreventLayerOverlaps

- *bPreventLayerOverlaps* 引数はブール値です。true では [レイヤーのオーバーラップ防止] オプションがオンになり、false ではオフになります。

戻り値
なし

dom.setShowFrameBorders()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[フレームボーダー] オプションのオンとオフを切り替えます。

引数
bShowFrameBorders

- *bShowFrameBorders* 引数はブール値です。true では [フレームボーダー] がオンになり、false ではオフになります。

戻り値
なし

dom.setShowGrid()

対応バージョン
Dreamweaver 3

説明
この関数は、View Grid Show[表示]-[グリッド]-[表示] オプションのオンとオフを切り替えます。

引数
bShowGrid

- *bShowGrid*引数はブール値です。true では [表示]-[グリッド]-[表示] オプションがオンになり、false ではオフになります。

戻り値
なし

dom.setShowHeadView()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[ヘッドコンテンツ] オプションのオンとオフを切り替えます。

引数
bShowHead

- *bShowHead*引数はブール値です。true では [ヘッドコンテンツ] オプションがオンになり、false ではオフになります。

戻り値
なし

dom.setShowInvalidHTML()

対応バージョン
Dreamweaver 4

説明
この関数は、ドキュメントウィンドウのコードビューで、無効な HTML コードのハイライト表示のオンとオフを切り替えます。

この関数は、無効な HTML コードが現在ハイライト表示されているかどうかを判別します。

引数
bShow

- *bShow*引数はブール値です。true では、無効な HTML コードのハイライト表示がオンになり、false ではオフになります。

戻り値
なし

dom.setShowImageMaps()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[イメージマップ] オプションのオンとオフを切り替えます。

引数
bShowImageMaps

- *bShowImageMaps* 引数はブール値です。true では [イメージマップ] オプションがオンになり、false ではオフになります。

戻り値
なし

dom.setShowInvisibleElements()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[不可視エレメント] オプションのオンとオフを切り替えます。

引数
bViewInvisibleElements

- *bViewInvisibleElements* 引数はブール値です。true では [不可視エレメント] オプションがオンになり、false ではオフになります。

戻り値
なし

dom.setShowLayerBorders()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[レイヤーボーダー] オプションのオンとオフを切り替えます。

引数
bShowLayerBorders

- *bShowLayerBorders* 引数はブール値です。true では [レイヤーボーダー] オプションがオンになり、false ではオフになります。

戻り値
なし

dom.setShowLineNumbers()

対応バージョン
Dreamweaver 4

説明
この関数は、ドキュメントウィンドウのコードビューで、行番号の表示と非表示を切り替えます。

引数
bShow

- *bShow* 引数はブール値です。true では、行番号の表示がオンになり、false ではオフになります。

戻り値
なし

dom.setShowRulers()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[ルーラ]-[表示] オプションのオンとオフを切り替えます。

引数
bShowRulers

- *bShowRulers* 引数はブール値です。true では [表示] オプションがオンになり、false ではオフになります。

戻り値
なし

dom.setShowSyntaxColoring()

対応バージョン
Dreamweaver 4

説明
この関数は、ドキュメントウィンドウのコードビューで、シンタックスカラーリングのオンとオフを切り替えます。

引数
bShow

- *bShow* 引数はブール値です。true では、シンタックスカラーリングがオンになり、false ではオフになります。

戻り値
なし

dom.setShowTableBorders()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[テーブルボーダー] オプションのオンとオフを切り替えます。

引数
bShowTableBorders

- *bShowTableBorders* 引数はブール値です。true では [テーブルボーダー] オプションがオンになり、false ではオフになります。

戻り値
なし

dom.setShowToolbar()

対応バージョン
Dreamweaver 4

説明
この関数は、ツールバーの表示と非表示を切り替えます。

引数
bShow

- *bShow* 引数はブール値です。true では、ツールバーの表示がオンになり、false ではオフになります。

戻り値
なし

dom.setShowTracingImage()

対応バージョン
Dreamweaver 3

説明
この関数は、[表示]-[トレーシングイメージ]-[表示] オプションのオンとオフを切り替えます。

引数
bShowTracingImage

- *bShowTracingImage* 引数はブール値です。true では [表示] オプションがオンになり、false ではオフになります。

戻り値
なし

dom.setShowWordWrap()

対応バージョン
Dreamweaver 4

説明
この関数は、ドキュメントウィンドウのコードビューで、[ワードラップ] オプションのオンとオフを切り替えます。

引数

bShow

- *bShow* 引数はブール値です。true では、行の折り返しがオンになり、false ではオフになります。

戻り値

なし

dom.setSnapToGrid()

対応バージョン

Dreamweaver 3

説明

この関数は、[表示]-[グリッド]-[吸着] オプションのオンとオフを切り替えます。

引数

bSnapToGrid

- *bSnapToGrid* 引数はブール値です。true では [吸着] オプションがオンになり、false ではオフになります。

戻り値

なし

dreamweaver.getHideAllFloaters()

対応バージョン

Dreamweaver 3

説明

この関数は、[パネルの非表示] オプションの現在の状態を取得します。

引数

なし

戻り値

ブール値。true は [パネルの非表示] オプションがオンであることを示し、false は [パネルの表示] オプションがオンであることを示します。

dreamweaver.getShowStatusBar()

対応バージョン

Dreamweaver 3

説明

この関数は、[表示]-[ステータスバー] オプションの現在の状態を取得します。

引数

なし

戻り値

ブール値。true はステータスバーの表示がオンであることを示し、false はオフであることを示します。

dreamweaver.htmlInspector.getShowAutoIndent()

対応バージョン
Dreamweaver 4

説明
この関数は、コードインスペクタで [自動インデント] オプションがオンになっているかどうかを判別します。

引数
なし

戻り値
ブール値。自動インデントがオンになっている場合は `true`、オフになっている場合は `false`。

dreamweaver.htmlInspector.getShowInvalidHTML()

対応バージョン
Dreamweaver 4

説明
この関数は、コードインスペクタで無効な HTML コードが現在ハイライト表示されているかどうかを判別します。

引数
なし

戻り値
ブール値。無効な HTML コードがハイライト表示されている場合は `true`、そうでない場合は `false`。

dreamweaver.htmlInspector.getShowLineNumbers()

対応バージョン
Dreamweaver 4

説明
この関数は、コードインスペクタで行番号が表示されているかどうかを判別します。

引数
なし

戻り値
ブール値。行番号が表示される場合は `true`、そうでない場合は `false`。

dreamweaver.htmlInspector.getShowSyntaxColoring()

対応バージョン
Dreamweaver 4

説明
この関数は、コードインスペクタでシンタックスカラーリングがオンになっているかどうかを判別します。

引数
なし

戻り値
ブール値。シンタックスカラーリングがオンになっている場合は `true`、オフになっている場合は `false`。

dreamweaver.htmlInspector.getShowWordWrap()

対応バージョン
Dreamweaver 4

説明
この関数は、コードインスペクタでテキストの折り返しがオンになっているかどうかを判別します。

引数
なし

戻り値
ブール値。テキストの折り返し (ワードラップ) がオンになっている場合は `true`、オフになっている場合は `false`。

dreamweaver.htmlInspector.setShowAutoIndent()

対応バージョン
Dreamweaver 4

説明
この関数は、コードインスペクタで [自動インデント] オプションのオンとオフを切り替えます。

引数
`bShow`

- *`bShow`* 引数はブール値です。true では自動インデントがオンになり、`false` ではオフになります。

戻り値
なし

dreamweaver.htmlInspector.setShowInvalidHTML()

対応バージョン
Dreamweaver 4

説明
この関数は、コードインスペクタで無効な HTML コードのハイライト表示のオンとオフを切り替えます。

引数
`bShow`

- *`bShow`* 引数はブール値です。`true` では、無効な HTML コードのハイライト表示がオンになり、`false` ではオフになります。

戻り値
なし

dreamweaver.htmlInspector.setShowLineNumbers()

対応バージョン
Dreamweaver 4

説明
この関数は、コードインスペクタのコードビューで、行番号の表示と非表示を切り替えます。

引数
bShow

- *bShow* 引数はブール値です。`true` では、行番号の表示がオンになり、`false` ではオフになります。

戻り値
なし

dreamweaver.htmlInspector.setShowSyntaxColoring()

対応バージョン
Dreamweaver 4

説明
この関数は、コードインスペクタのコードビューで、シンタックスカラーリングのオンとオフを切り替えます。

引数
bShow

- *bShow* 引数はブール値です。`true` では、シンタックスカラーリングがオンになり、`false` ではオフになります。

戻り値
なし

dreamweaver.htmlInspector.setShowWordWrap()

対応バージョン
Dreamweaver 4

説明
この関数は、コードインスペクタで [ワードラップ] オプションのオンとオフを切り替えます。

引数
bShow

- *bShow* 引数はブール値です。`true` では [ワードラップ] がオンになり、`false` ではオフになります。

戻り値
なし

dreamweaver.setHideAllFloaters()

対応バージョン
Dreamweaver 3

説明

この関数は、[パネルの非表示] オプションまたは [パネルの表示] オプションをオンにします。

引数

bShowFloatingPalettes

- *bShowFloatingPalettes* 引数はブール値です。true では [パネルの非表示] オプションがオンになり、false ではオフになります。

戻り値

なし

dreamweaver.setShowStatusBar()

対応バージョン

Dreamweaver 3

説明

この関数は、[表示]-[ステータスバー] オプションのオンとオフを切り替えます。

引数

bShowStatusBar

- *bShowStatusBar* 引数はブール値です。true では [ステータスバー] オプションがオンになり、false ではオフになります。

戻り値

なし

site.getShowDependents()

対応バージョン

Dreamweaver 3

説明

この関数は、[依存ファイルの表示] オプションの現在の状態を取得します。

引数

なし

戻り値

ブール値。true はサイトマップに依存ファイルが表示されていることを示し、false は表示されていないことを示します。

site.getShowHiddenFiles()

対応バージョン

Dreamweaver 3

説明

この関数は、[非表示としてマークしたファイルの表示] オプションの現在の状態を取得します。

引数
なし

戻り値
ブール値。true はサイトマップに非表示ファイルが表示されていることを示し、false は表示されていないことを示します。

site.getShowPageTitles()

対応バージョン
Dreamweaver 3

説明
この関数は、[ページタイトルの表示] オプションの現在の状態を取得します。

引数
なし

戻り値
ブール値。true はサイトマップにページタイトルが表示されていることを示し、false は表示されていないことを示します。

site.getShowToolTips()

対応バージョン
Dreamweaver 3

説明
この関数は、[ツールヒント] オプションの現在の状態を取得します。

引数
なし

戻り値
ブール値。true は [サイト] パネルにツールヒントが表示されていることを示し、false は表示されていないことを示します。

site.setShowDependents()

対応バージョン
Dreamweaver 3

説明
この関数は、サイトマップの [依存ファイルの表示] オプションのオンとオフを切り替えます。

引数
bShowDependentFiles

- *bShowDependentFiles* 引数はブール値です。true では [依存ファイルの表示] オプションがオンになり、false ではオフになります。

戻り値
なし

site.setShowHiddenFiles()

対応バージョン
Dreamweaver 3

説明
この関数は、サイトマップの [非表示としてマークしたファイルの表示] オプションのオンとオフを切り替えます。

引数
bShowHiddenFiles

- *bShowHiddenFiles* 引数はブール値です。true では [非表示としてマークされたファイルの表示] オプションがオンになり、false ではオフになります。

戻り値
なし

site.setShowPageTitles()

対応バージョン
Dreamweaver 3

説明
この関数は、サイトマップの [ページタイトルの表示] オプションのオンとオフを切り替えます。

引数
bShowPageTitles

- *bShowPageTitles* 引数はブール値です。true では [ページタイトルの表示] オプションがオンになり、false ではオフになります。

戻り値
なし

イネーブラ
詳細については、440 ページの site.canShowPageTitles() を参照してください。

site.setShowToolTips()

対応バージョン
Dreamweaver 3

説明
この関数は、[ツールヒント] オプションのオンとオフを切り替えます。

引数
bShowToolTips

- *bShowToolTips* 引数はブール値です。true では [ツールヒント] オプションがオンになり、false ではオフになります。

戻り値
なし

ツールバー関数

次の JavaScript 関数を使用して、ツールバーおよびツールバーラベルの表示と表示設定、現在のウィンドウのツールバー項目のラベルの取得、ツールバーの配置、およびツールバー ID の取得を行うことができます。ツールバーの作成および修正の詳細については、『Dreamweaver 拡張ガイド』の「ツールバー」を参照してください。

dom.forceToolbarUpdate()

対応バージョン

Dreamweaver MX 2004

説明

指定したツールバーのすべての項目に対して、更新ハンドラを強制的に実行します。

引数

toolbarID

- *toolbarID* 引数には、項目の更新を実行するツールバーの ID を指定します。

戻り値

なし

dom.getShowToolbarIconLabels()

対応バージョン

Dreamweaver MX

説明

この関数は、現在のドキュメントウィンドウでボタンのラベルが表示されているかどうかを判別します。Dreamweaver では、ボタン以外のコントロールに定義されているラベルは常に表示されます。

引数

なし

戻り値

ブール値。現在のドキュメントウィンドウでボタンのラベルが表示されている場合は `true`、表示されていない場合は `false`。

例

次の例では、ボタンのラベルの表示をオンにしています。

```
var dom = dw.getDocumentDom();
if (dom.getShowToolbarIconLabels() == false)
{
    dom.setShowToolbarIconLabels(true);
}
```

dom.getToolbarIdArray()

対応バージョン

Dreamweaver MX

説明

この関数は、アプリケーションのすべてのツールバーの ID 配列を返します。`dom.getToolbarIdArray()` を使用して、すべてのツールバーの表示をオフにして再配置し、必要なツールバーのみを表示することができます。

引数

なし

戻り値

すべてのツールバー ID の配列。

例

次の例では、ツールバー ID の配列を tb_ids 変数に格納しています。

```
var tb_ids = new Array();
tb_ids = dom.getToolbarIdArray();
```

dom.getToolbarItemValue()

対応バージョン

Dreamweaver MX 2004

説明

指定したツールバー項目の値を取得します。

引数

toolbarID、*itemID*

- *toolbarID* 引数には、値を取得する項目が含まれるツールバーの ID を表すストリングを指定します。
- *itemID* 引数には、値を取得する項目の ID を表すストリングを指定します。

戻り値

ツールバー項目の値を表すストリング。

例

次に示す receiveArguments() の例は、[サイズ] テキストフィールドの動作を制御するツールバーコマンド内で使われます。この例では、[サイズ] フィールドの値を引数として取得し、CSS プロパティの font-size 関数に対する有効な値を生成するために [単位] フィールドの値を読み取っています。

```
receiveArguments(newSize){
var dom = dw.getDocumentDOM();
if (newSize != ""){
    dom.applyFontMarkupAsStyle('font-size', newSize +
    dom.getToolbarItemValue("DW_Toolbar_Text","DW_Text_Units"));
    }
else{
    dom.removeFontMarkupAsStyle('font-size');
    }
}
```

dom.getToolbarLabel()

対応バージョン

Dreamweaver MX

説明

この関数は、指定したツールバーのラベルを取得します。dom.getToolbarLabel() は、ツールバーを表示または非表示にするメニューに対して使用できます。

引数

toolbar_id

- *toolbar_id* 引数には、ツールバーの ID を指定します。ツールバーの ID は、"toolbars.xml" ファイル内の toolbar タグの ID 属性の値です。

戻り値

toolbar タグの属性として割り当てられているラベル名のストリング。

例

次の例では、myEditbar のラベルを変数 label に格納しています。

```
var label = dom.getToolbarLabel("myEditbar");
```

dom.getToolbarVisibility()

対応バージョン

Dreamweaver MX

説明

この関数は、*toolbar_id* で指定したツールバーが表示されているかどうかを示すブール値を返します。

引数

toolbar_id

- *toolbar_id* 引数には、ツールバーに割り当てられている ID ストリングを指定します。

戻り値

ブール値。ツールバーが表示されている場合は true、ツールバーがないか、または表示されていない場合は false。

例

次の例では、ツールバー myEditbar がドキュメントウィンドウに表示されているかどうかをチェックし、値を変数 retval に格納しています。

```
var retval = dom.getToolbarVisibility("myEditbar");

return retval;
```

dom.setToolbarItemAttribute()

対応バージョン

Dreamweaver MX 2004

説明

ツールバー項目の 3 つのイメージ属性、または tooltip 属性の属性値を変更します。

引数

toolbarID、*toolbarItemId*、*attrName*、*attrValue*

- *toolbarID* 引数には、ツールバーの ID を表すストリングを指定します。
- *toolbarItemId* 引数には、ツールバー項目の ID を表すストリングを指定します。
- *attrName* 引数には、設定する属性の名前を表すストリングを指定します。有効な値は、'image'、'overImage'、'disabledImage'、または 'tooltip' です。
- *attrValue* 引数には、設定する値を表すストリングを指定します。

戻り値

なし

例

次の例では、dom.setToolbarItemAttribute() を 3 回呼び出して、DW_Toolbar_Main という ID を持つツールバーのツールバー項目 MyButton に、image、imageOver、および tooltip 属性を設定しています。

```
var dom = dw.getDocumentDOM();
dom.setToolbarItemAttribute('DW_Toolbar_Main', 'MyButton', 'image',
    'Toolbars/imgs/newimage.gif');
dom.setToolbarItemAttribute('DW_Toolbar_Main', 'MyButton', 'imageOver',
    'Toolbars/imgs/newimageOver.gif');
dom.setToolbarItemAttribute('DW_Toolbar_Main', 'MyButton', 'tooltip', 'One fine button');
```

dom.setShowToolbarIconLabels()

対応バージョン

Dreamweaver MX

説明

この関数は、ラベル付きボタンのラベルを表示するように Dreamweaver に指示します。Dreamweaver では、ボタン以外のコントロールに定義されているラベルは常に表示されます

引数

bShow

- *bShow* 引数はブール値です。true ではボタンのラベルが表示され、false では表示されません。

戻り値

なし

例

次の例では、ツールバーのボタンのラベルを表示するように指示しています。

```
dom.setShowToolbarIconLabels(true);
```

dom.setToolbarPosition()

対応バージョン

Dreamweaver MX

説明

この関数は、指定したツールバーを指定した位置に移動します。

注意 : ツールバーの現在の位置を判別する方法はありません。

引数

toobar_id、*position*、*relative_to*

- *toolbar_id* 引数には、ツールバーの ID を指定します。ツールバーの ID は、"toolbars.xml" ファイル内の toolbar タグの ID 属性の値です。
- *position* 引数には、その他のツールバーを基準として、ツールバーを配置する位置を指定します。指定できる *position* の値について、以下で説明します。
- top は、デフォルトの位置です。ツールバーはドキュメントウィンドウの最上部に表示されます。

- below を指定すると、ツールバーは *relative_to* で指定したツールバーのすぐ下の列の先頭に表示されます。*relative_to* で指定したツールバーが見つからない場合は、Dreamweaver によってエラーが報告されます。
- floating を指定すると、ツールバーはドキュメント上にフローティング状態で表示されます。ツールバーは自動的にその他のフローティングツールバーからずらして配置されます。Macintosh の場合、floating は top と同じように扱われます。
- *relative_to*="*toolbar_id*" は、*position* に below を指定した場合は必須です。それ以外の場合、この引数は無視されます。指定した ID のツールバーの下にこのツールバーを配置します。

戻り値

なし

例

次の例では、myEditbar の位置を myPicturebar ツールバーの下に設定しています。

```
dom.setToolbarPosition("myEditbar", "below", "myPicturebar");
```

dom.setToolbarVisibility()

対応バージョン

Dreamweaver MX

説明

この関数は、指定したツールバーの表示と非表示を切り替えます。

引数

toolbar_id、*bShow*

- *toolbar_id* 引数には、ツールバーの ID を指定します。ツールバーの ID は、"toolbars.xml" ファイル内の toolbar タグの ID 属性の値です。
- *bShow* 引数には、ツールバーの表示または非表示を示すブール値を指定します。*bshow* に true を指定すると、dom.setToolbarVisibility() によってツールバーが表示されます。*bShow* に false を指定すると、dom.setToolbarVisibility() によりツールバーは非表示になります。

戻り値

なし

例

次の例では、ドキュメントウィンドウに myEditbar ツールバーが表示されているかどうかをチェックし、表示されていない場合は表示されるように設定しています。

```
var dom = dw.getDocumentDOM();
if(dom != null && dom.getToolbarVisibility("myEditbar") == false)
{
    dom.setToolbarVisibility("myEditbar", true);
{
```

ウィンドウ関数

ウィンドウ関数は、ドキュメントウィンドウおよびフローティングパネルに関する操作に使用されます。ウィンドウ関数を使用して、フローティングパネルの表示と非表示、ドキュメントウィンドウでフォーカスがある部分の判別、およびアクティブなドキュメントの設定を実行することができます。[サイト] パネルに固有の操作に関する詳細については、178 ページのサイト関数を参照してください。

注意 : この項で扱う関数の中には、*Windows* のみで動作するものがあります。関数についての説明では、これに該当するかどうかを示してあります。

dom.getFocus()

対応バージョン

Dreamweaver 3

説明

この関数は、ドキュメントで現在フォーカスのある部分を判別します。

引数

なし

戻り値

次のストリングのいずれかを返します。

- `HEAD` 領域がアクティブな場合は、`"head"`。
- `BODY` または `NOFRAMES` 領域がアクティブな場合は、`"body"`。
- フレームセットまたはそこに含まれるフレームが選択されている場合は `"frameset"`。
- このドキュメントにフォーカスがない場合 (たとえば、プロパティインスペクタや別のフローティングパネルにフォーカスがある場合) は `"none"`。

dom.getView()

対応バージョン

Dreamweaver 4

説明

この関数は、表示されているビューを判別します。

引数

なし

戻り値

表示ビューに応じて、`"design"`、`"code"`、または `"split"`。

dom.getWindowTitle()

対応バージョン

Dreamweaver 3

説明

この関数は、ドキュメントを含むウィンドウのタイトルを取得します。

引数

なし

戻り値

ドキュメントの TITLE タグに囲まれたテキストを含むストリング。開いているウィンドウにドキュメントがない場合は、何も返されません。

dom.setView()

対応バージョン

Dreamweaver 4

説明

この関数は、デザインビューまたはコードビューの表示と非表示を切り替えて、デザイン専用、コード専用、または分割のビューを作成します。

引数

viewString

- *viewString* 引数には、作成するビューを指定します。この値は、"design"、"code"、または "split" のいずれかにする必要があります。

戻り値

なし

dreamweaver.bringAttentionToFloater()

対応バージョン

Dreamweaver MX

説明

指定したパネルまたはインスペクタが目立つように、前面に移動して点滅させます。この機能は、dw.toggleFloater() とは多少異なります。

引数

floaterName

- *floaterName* 引数には、ウィンドウ、パネル、またはインスペクタの名前を指定します。

戻り値

なし

例

次の例では、[アセット] パネルを開き、点滅させています。

```
dw.bringAttentionToFloater("library");
```

dreamweaver.cascade()

対応バージョン

Dreamweaver MX (Windows のみ)、Dreamweaver 8 (Macintosh サポート追加)。

説明
ドキュメントウィンドウを重ねて表示します。左上隅に最初のウィンドウを表示し、少しずつ斜め下にずらしながらウィンドウを重ねて表示します。

引数
なし

戻り値
なし

例
次の例では、開いているドキュメントを重ねて表示します。

```
dw.cascade()
```

dreamweaver.getActiveWindow()

対応バージョン
Dreamweaver 3

説明
この関数は、アクティブなウィンドウにあるドキュメントを取得します。

引数
なし

戻り値
アクティブなウィンドウにあるドキュメントに対応するドキュメントオブジェクト。ドキュメントがフレーム内にある場合は、フレームセットに対応するドキュメントオブジェクトが返されます。

dreamweaver.getDocumentList()

対応バージョン
Dreamweaver 3

説明
この関数は、開いているすべてのドキュメントのリストを取得します。

引数
なし

戻り値
それぞれが、開いているドキュメントウィンドウに対応するドキュメントオブジェクトから成る配列。ドキュメントウィンドウにフレームセットが含まれる場合は、フレームのコンテンツではなくフレームセットを参照するドキュメントオブジェクトが返されます。

dreamweaver.getFloaterVisibility()

対応バージョン
Dreamweaver 3

説明

この関数は、指定したフローティングパネルまたはインスペクタが表示されているかどうかをチェックします。

引数

floaterName

- *floaterName* 引数には、フローティングパネルの名前を指定します。*floaterName* に指定された名前が組み込みパネルのいずれの名前にも一致しない場合、Dreamweaver では "Configuration¥Floaters" フォルダで *"floaterName*.htm" というファイルが検索されます。このファイル名の *floaterName* は、フローティングパネルの名前です。

次のリストのパネル名の右側のストリングは、Dreamweaver に組み込まれているパネルに対応する *floaterName* の値です。

アセット = "assets"
ビヘイビア = "behaviors"
バインディング = "data bindings"
コードインスペクタ = "html"
コンポーネント = "server components"
CSS スタイル = "css styles"
フレーム = "frames"
ヒストリ = "history"
挿入バー = "objects"
レイヤー = "layers"
ライブラリ = "library"
リンクチェック = "linkchecker"
プロパティ = "properties"
リファレンス = "reference"
サイトレポート = "reports"
検索 = "search"
選択インスペクタ = "selection inspector"
サーバービヘイビア = "server behaviors"
サイト = "site"
サイトファイル = "site files"
サイトマップ = "site map"
スニペット = "snippets"
ブラウザ互換性チェック = "bcc"
バリデータ = "validation"

戻り値

ブール値。フローティングパネルが前面に表示されている場合は `true`。それ以外の場合、または Dreamweaver で *floaterName* という名前のフローティングパネルが見つからない場合は `false` が返されます。

dreamweaver.getFocus()

対応バージョン

Dreamweaver 4

説明

この関数は、アプリケーションで現在フォーカスのある部分を判別します。

引数

bAllowFloaters

- bAllowFloaters 引数はブール値です。フローティングパネルにフォーカスがある場合にその名前を受け取るには true、それ以外の場合は false を指定します。

戻り値

次のストリングのいずれかを返します。

- ドキュメントウィンドウにフォーカスがある場合は "document"。
- [サイト] パネルにフォーカスがある場合は "site"。
- テキストビューにフォーカスがある場合は "textView"。
- コードインスペクタにフォーカスがある場合は "html"。
- bAllowFloaters が true で、フローティングパネルにフォーカスがある場合は floaterName ストリング。ここで、floaterName は "objects"、"properties"、"launcher"、"library"、"css styles"、"html styles"、"behaviors"、"timelines"、"layers"、"frames"、"templates"、または "history" のいずれかです。
- Macintosh では、[サイト] パネルもドキュメントウィンドウも開いていない場合は "none"。

dreamweaver.getPrimaryView()

対応バージョン

Dreamweaver 4

説明

この関数は、プライマリビューとして前面に表示されているビューを判別します。

引数

なし

戻り値

表示ビュー、または分割ビューでの最上部のビューに応じて、"design" または "code"。

dreamweaver.getSnapDistance()

対応バージョン

Dreamweaver 4

説明

この関数は、吸着距離をピクセル単位で返します。

引数

なし

戻り値

吸着距離をピクセル単位で表した整数。デフォルト値は 10 ピクセルです。0 は、吸着機能がオフになっていることを示します。

dreamweaver.minimizeRestoreAll()

対応バージョン
Dreamweaver 4

説明
この関数は、Dreamweaver 内のすべてのウィンドウを最小化 (アイコンに縮小) するか、または復元します。

引数
bMinimize

- *bMinimize* 引数はブール値です。ウィンドウを最小化する場合は true、最小化されたウィンドウを復元する場合は false を指定します。

戻り値
なし

dreamweaver.setActiveWindow()

対応バージョン
Dreamweaver 3

説明
この関数は、指定したドキュメントを含むウィンドウをアクティブにします。

引数
documentObject、*{bActivateFrame}*

- *documentObject* 引数には、ドキュメントの DOM ツリーのルートにあるオブジェクト (dreamweaver.getDocumentDOM() 関数で返される値) を指定します。
- *bActivateFrame* 引数はオプションです。*documentObject* がフレームセット内にある場合にのみ使用できます。*bActivateFrame* 引数はブール値です。フレームセットを含むウィンドウだけでなくドキュメントを含むフレームもアクティブにするには true、それ以外の場合は false を指定します。

戻り値
なし

dreamweaver.setFloaterVisibility()

対応バージョン
Dreamweaver 3

説明
この関数は、特定のフローティングパネルまたはインスペクタを表示するかどうかを指定します。

引数
floaterName、*bIsVisible*

- *floaterName* 引数には、フローティングパネルの名前を指定します。*floaterName* に指定された名前が組み込みパネルのいずれの名前にも一致しない場合、Dreamweaver では "Configuration¥Floaters" フォルダで *"floaterName*.htm" というファイルが検索されます。このファイル名の *floaterName* は、フローティングパネルの名前です。Dreamweaver で *floaterName* というフローティングパネルが見つからない場合は、何も行われません。

次のリストのパネル名の右側のストリングは、Dreamweaver に組み込まれているパネルに対応する *floaterName* の値です。

アセット = "assets"
ビヘイビア = "behaviors"
バインディング = "data sources"
コードインスペクタ = "html"
コンポーネント = "server components"
CSS スタイル = "css styles"
フレーム = "frames"
ヒストリ = "history"
HTML スタイル = "html styles"
挿入バー = "objects"
レイヤー = "layers"
ライブラリ = "library"
リンクチェック = "linkchecker"
プロパティ = "properties"
リファレンス = "reference"
サイトレポート = "reports"
検索 = "search"
サーバービヘイビア = "server behaviors"
サイト = "site"
サイトファイル = "site files"
サイトマップ = "site map"
スニペット = "snippets"
タグインスペクタ = "tag inspector"
ブラウザ互換性チェック = "bcc"
テンプレート = "templates"
バリデータ = "validation"

bIsVisible 引数には、フローティングパネルを表示するかどうかを示すブール値を指定します。

戻り値

なし

dreamweaver.setPrimaryView()

対応バージョン

Dreamweaver 4

説明

この関数は、指定したビューをドキュメントウィンドウの最上部に表示します。

引数

viewString

- *viewString* 引数には、ドキュメントウィンドウの最上部に表示するビューを指定します。"design" または "code" のいずれかを指定します。

戻り値

なし

dreamweaver.setSnapDistance()

対応バージョン

Dreamweaver 4

説明

この関数は、吸着距離をピクセル単位で設定します。

引数

snapDistance

- *snapDistance* 引数には、吸着距離をピクセル単位で表した整数を指定します。デフォルト値は 10 ピクセルです。0 を指定すると、吸着機能がオフになります。

戻り値

なし

dreamweaver.showProperties()

対応バージョン

Dreamweaver 3

説明

この関数は、プロパティインスペクタを表示して、フォーカスを与えます。

引数

なし

戻り値

なし

dreamweaver.tileHorizontally()

対応バージョン

Dreamweaver MX (Windows のみ)、Dreamweaver 8 (Macintosh サポート追加)。

説明

ドキュメントウィンドウを水平に並べます。ドキュメントは、互いに重ならないように横に並べられます。この処理は、ワークスペースを垂直に分割する場合と似ています。

引数

なし

戻り値

なし

例

次の例では、開いているドキュメントを水平に並べて表示します。

```
dw.tileHorizontally()
```

dreamweaver.tileVertically()

対応バージョン

Dreamweaver MX (Windows のみ)、Dreamweaver 8 (Macintosh サポート追加)。

説明

ドキュメントウィンドウを縦に並べます。ドキュメントは、互いに重ならないように縦に並べられます。この処理は、ワークスペースを水平に分割する場合と似ています。

引数

なし

戻り値

なし

例

次の例では、開いているドキュメントを縦に並べて表示します。

```
dw.tileVertically()
```

dreamweaver.toggleFloater()

対応バージョン

Dreamweaver 3

説明

この関数は、指定したパネルまたはインスペクタを表示、非表示、または前面に表示します。

注意 : この関数の効果があるのは、"*menus.xml*" ファイル内で使用した場合だけです。フローティングパネルを表示、非表示、または前面に表示するには、*`dw.setFloaterVisibility()`* を使用します。

引数

`floaterName`

- *`floaterName`* 引数には、ウィンドウの名前を指定します。フローティングパネル名が `reference` の場合、[リファレンス] パネルの表示 / 非表示の状態は、コードビューにおけるユーザーの選択範囲によって更新されます。その他のパネルはすべて、常にユーザーの選択範囲をトラッキングします。ただし、[リファレンス] パネルがコードビューにおける選択範囲をトラッキングするのは、ユーザーがトラッキングを呼び出すときだけです。

戻り値

なし

dreamweaver.updateReference()

対応バージョン
Dreamweaver 4

説明
この関数は、[リファレンス] フローティングパネルを更新します。[リファレンス] フローティングパネルが表示されていない場合は、`dw.updateReference()` によりフローティングパネルが表示および更新されます。

引数
なし

戻り値
なし

コードの折りたたみ関数

コードの折りたたみ関数を使用すると、コードを視覚的に折りたたんだり展開したりできます。コードの任意の選択範囲、または開始タグと終了タグの間の部分を折りたたんだり展開したりできます。コード折りたたみ関数は dom と htmlInspector の両方に存在しますが、折りたたまれる範囲はコードビューでもコードインスペクタでも同じです。

dom.collapseFullTag()

対応バージョン
Dreamweaver 8

説明
この関数は、コードビューでの選択範囲が、開始タグと終了タグの 1 つのペアで完全に囲まれているか、または開始タグと終了タグの 1 つのペアを含んでいるかどうかを判別します。その場合は、開始タグの直前から終了タグの直後までのコード部分を折りたたみます。それ以外の場合は何もしません。

引数
`allowCodeFragmentAdjustment`

- *`allowCodeFragmentAdjustment`* 引数は必須で、ブール値です。現在のところ、この引数に `true` を指定しても効果はありません。つまり、値 `false` と同じ効果になります。`false` の場合は、開始タグの直前から終了タグの直後までのコードが、変更されることなく折りたたまれます。

戻り値
なし

例
次の例では、コードビューで現在選択されている範囲内の開始タグの直前から終了タグの直後までのコード部分を折りたたみます。

```
var currentDOM = dw.getDocumentDOM();
currentDOM.collapseFullTag(false);
```

dom.collapseFullTagInverse()

対応バージョン

Dreamweaver 8

説明

この関数は、コードビューでの選択範囲が、開始タグと終了タグの 1 つのペアで完全に囲まれているか、または開始タグと終了タグの 1 つのペアを含んでいるかどうかを判別します。その場合は、開始タグより前のコードと終了タグより後のコードを折りたたみます。それ以外の場合は何もしません。

引数

allowAdjustmentOfCodeFragments

- *allowAdjustmentOfCodeFragments* 引数は必須で、ブール値です。true の場合は、開始タグより前のコードと終了タグより後のコードの境界が調整され、スマート折りたたみが実行されます。スマート折りたたみでは、現在のインデントと余白が保たれます。false の場合は、開始タグより前と終了タグより後のコード部分が、選択範囲で示されるとおりに折りたたまれます。

戻り値

なし

例

次の例では、開始タグより前と終了タグより後のコードの境界を調整して、インデントと余白を保つスマート折りたたみを実行します。

```
var currentDOM = dw.getDocumentDOM();
currentDOM.collapseFullTagInverse(true);
```

dom.collapseSelectedCodeFragment()

対応バージョン

Dreamweaver 8

説明

この関数は、コードビューで選択されているコードを折りたたみます。選択範囲が既に折りたたまれている場合は、何も実行されません。

引数

allowCodeFragmentAdjustment

- *allowCodeFragmentAdjustment* は必須で、ブール値です。true の場合、現在の選択範囲の境界が調整され、スマート折りたたみが実行されます。スマート折りたたみでは、現在のインデントと余白が保たれます。false の場合、現在選択されたいるコード部分が、選択範囲で示されるとおりに折りたたまれます。

戻り値

なし

例

次の例では、コードビューで選択されているコード部分を、変更することなく折りたたみます。

```
var currentDOM = dw.getDocumentDOM();
currentDOM.collapseSelectedCodeFragment(false);
```

dom.collapseSelectedCodeFragmentInverse()

対応バージョン
Dreamweaver 8

説明
この関数は、コードビューで選択されたコードの前と後のすべてのコードを折りたたみます。

引数
allowAdjustmentOfCodeFragments

- *allowAdjustmentOfCodeFragments* 引数は必須で、ブール値です。true の場合、現在の選択範囲の前と後のコードの境界が調整され、スマート折りたたみが実行されます。スマート折りたたみでは、現在のインデントと余白が保たれます。false の場合、選択範囲によって示されるとおりに、コード部分が折りたたまれます。

戻り値
なし

例
次の例では、コードビューで選択されているコードの前後のすべてのコードを調整して折りたたみます。

```
var currentDOM = dw.getDocumentDOM();
currentDOM.collapseSelectedCodeFragmentInverse(true);
```

dom.expandAllCodeFragments()

対応バージョン
Dreamweaver 8

説明
この関数は、コードビューで折りたたまれているコード部分を、ネストされて折りたたまれている部分も含めて、すべて展開します。

引数
なし

戻り値
なし

例
次の例では、コードビューで折りたたまれているコードをすべて展開します。

```
var currentDOM = dw.getDocumentDOM();
currentDOM.expandAllCodeFragments();
```

dom.expandSelectedCodeFragments()

対応バージョン
Dreamweaver 8

説明
この関数は、コードビューの現在の選択範囲内で折りたたまれているコード部分をすべて展開します。選択範囲が既に展開されている場合は、何も実行されません。

引数

なし

戻り値

なし

例

次の例では、コードビューの現在の選択範囲内で折りたたまれているコードをすべて展開します。

```
var currentDOM = dw.getDocumentDOM();
currentDOM.expandSelectedCodeFragments();
```

dreamweaver.htmlInspector.collapseFullTag()

対応バージョン

Dreamweaver 8

説明

この関数は、コードインスペクタでの選択範囲が、開始タグと終了タグの 1 つのペアで完全に囲まれているか、または開始タグと終了タグの 1 つのペアを含んでいるかどうかを判別します。true の場合は、開始タグの直前から終了タグの直後までのコード部分を折りたたみます。それ以外の場合は何もしません。

引数

allowACodeFragmentAdjustment

- *allowCodeFragmentAdjustment* 引数は必須で、ブール値です。現在のところ、この引数に `true` を指定しても効果はありません。つまり、値 `false` と同じ効果になります。`false` の場合は、開始タグの直前から終了タグの直後までのコードが、変更されることなく折りたたまれます。

戻り値

なし

例

次の例では、コードインスペクタで現在選択されている範囲内の開始タグの直前から終了タグの直後までのコード部分を折りたたみます。

```
dreamweaver.htmlInspector.collapseFullTag(false);
```

dreamweaver.htmlInspector.collapseFullTagInverse()

対応バージョン

Dreamweaver 8

説明

この関数は、コードインスペクタでの選択範囲が、開始タグと終了タグの 1 つのペアで完全に囲まれているか、または開始タグと終了タグの 1 つのペアを含んでいるかどうかを判別します。その場合は、開始タグの前のコードと終了タグの後のコードを折りたたみます。それ以外の場合は何もしません。

引数

allowAdjustmentOfCodeFragments

- *allowAdjustmentOfCodeFragments* 引数は必須で、ブール値です。`true` の場合、開始タグより前のコードと終了タグより後のコードの境界が調整され、スマート折りたたみが実行されます。スマート折りたたみでは、既存のインデントと余白が保たれます。`false` の場合は、開始タグの前のコードと終了タグの後のコードが、変更されることなく折りたたまれます。

戻り値

なし

例

次の例では、現在の選択範囲の開始タグの前と終了タグの後に存在するコードセクションに対して、スマート折りたたみを実行します。

```
dreamweaver.htmlInspector.collapseFullTagInverse(true);
```

dreamweaver.htmlInspector.collapseSelectedCodeFragment()

対応バージョン

Dreamweaver 8

説明

この関数は、コードインスペクタで選択されているコードを折りたたみます。選択範囲が既に折りたたまれている場合は、何も実行されません。

引数

allowCodeFragmentAdjustment

- *allowCodeFragmentAdjustment* は必須で、ブール値です。true の場合、現在の選択範囲が変更され、スマート折りたたみが実行されます。スマート折りたたみでは、既存のインデントと余白が保たれます。false の場合、現在選択されたいるコード部分が、選択範囲で示されるとおりに折りたたまれます。

戻り値

なし

例

次の例では、コードインスペクタで選択されているコードを調整して折りたたみます。

```
dreamweaver.htmlInspector.collapseSelectedCodeFragment(true);
```

dreamweaver.htmlInspector.collapseSelectedCodeFragmentInverse()

対応バージョン

Dreamweaver 8

説明

この関数は、コードインスペクタで選択されたコードより前と後のすべてのコードを折りたたみます。選択範囲が既に折りたたまれている場合は、何も実行されません。

引数

allowAdjustmentOfCodeFragments

- *allowAdjustmentOfCodeFragments* 引数は必須で、ブール値です。true の場合、現在の選択範囲の前と後のコードセクションの境界が調整され、スマート折りたたみが実行されます。スマート折りたたみでは、現在のインデントと余白が保たれます。false の場合、選択範囲によって示されるとおりに、コードセクションが折りたたまれます。

戻り値

なし

例

次の例では、コードインスペクタで選択されているコードの前後のコードを、選択範囲で示されるとおりにすべて折りたたみます。

```
dreamweaver.htmlInspector.collapseSelectedCodeFragmentInverse(false);
```

dreamweaver.htmlInspector.expandAllCodeFragments()

対応バージョン

Dreamweaver 8

説明

この関数は、コードインスペクタで折りたたまれているコード部分を、ネストされて折りたたまれている部分も含めて、すべて展開します。

引数

なし

戻り値

なし

例

次の例では、コードインスペクタで折りたたまれているコードをすべて展開します。

```
dreamweaver.htmlInspector.expandAllCodeFragments();
```

dreamweaver.htmlInspector.expandSelectedCodeFragments()

対応バージョン

Dreamweaver 8

説明

この関数は、コードインスペクタの現在の選択範囲内で折りたたまれているコード部分をすべて展開します。選択範囲が既に展開されている場合は、何も実行されません。

引数

なし

戻り値

なし

例

次の例では、コードインスペクタの現在の選択範囲内で折りたたまれているコードをすべて展開します。

```
dreamweaver.htmlInspector.expandSelectedCodeFragments();
```

コードビューツールバー関数

コードビューツールバー関数を使用すると、テキストの挿入、コメントの削除、コードビューのホワイトスペースを表す特殊文字の表示と非表示の切り替え、および現在のドキュメントのパスの取得が可能です。

注意 : コードビュー用とコードインスペクタ用の *2* つのコーディングツールバーがあります。いずれのツールバーも、"*Configuration/Toolbars/toolbars.xml*" ファイルでカスタマイズできます。

dom.getOpenPathName()

対応バージョン
Dreamweaver 8

説明
この関数は、開いているドキュメントの絶対ファイルパスを取得します。

引数
なし

戻り値
開いているドキュメントの絶対ファイルパスのストリング。

例
次の例では、現在開いているドキュメントのパスを格納しているストリングを変数 `fileName` に割り当てます。

```
var fileName = dom.getOpenPathName();
```

dom.getShowHiddenCharacters()

対応バージョン
Dreamweaver 8

説明
この関数は、ドキュメントウィンドウのコードビューで、ホワイトスペースを表す特殊文字が表示されているかどうかを判別します。

引数
なし

戻り値
ブール値。非表示の文字が表示される場合は `true`、そうでない場合は `false`。

例
次の例では、ホワイトスペースを表す特殊文字の表示が最初にオンになっていた場合に、特殊文字の表示をオフにします。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.getShowHiddenCharacters()){
    currentDOM.setShowHiddenCharacters(false);
    }
```

dom.setShowHiddenCharacters()

対応バージョン
Dreamweaver 8

説明
この関数は、コードインスペクタのコードビューで、ホワイトスペースの特殊文字の表示と非表示を切り替えます。

例については、173 ページの dom.getShowHiddenCharacters() を参照してください。

引数

show

- 必須の *show* 引数はブール値で、非表示の文字を表示するかどうかを示します。

戻り値

なし

dom.source.applyComment()

対応バージョン

Dreamweaver 8

説明

この関数は、*beforeText* 引数で指定されるテキストを現在の選択範囲の前に、*afterText* 引数で指定されるテキストを現在の選択範囲の後に挿入します。さらに、現在の選択範囲を拡張し、追加したテキストを含めます。ただし、選択範囲がない場合、この関数は何も選択しません。*afterText* 引数で選択されたテキストがヌルの場合、この関数は、*beforeText* 引数で指定されたテキストを現在の選択範囲のすべての行の先頭に挿入します。

引数

beforeText、*afterText*

- *beforeText* 引数の指定は必須です。この引数には、選択範囲の先頭に挿入するテキストを指定します。または、*afterText* 引数の値がヌルの場合、選択範囲のすべての行の先頭に挿入するテキストを指定します。
- オプションの *afterText* 引数には、選択範囲の末尾に挿入するテキストを指定します。

戻り値

なし

例

次の例では、現在の選択範囲を HTML コメントにします。

```
dw.getDocumentDOM().source.applyComment('<!--', '-->')
```

dom.source.refreshVariableCodeHints()

対応バージョン

Dreamweaver CS3

説明

ページを再スキャンし、変数およびそれに対応するクラスの関連付けを検出します。この関数は、カラーステートエンジンおよび変数リストを再構築します。

引数

bSyncDoc

- ブール値を指定します。デフォルトは、`false` です。`true` に設定すると、デザインビューとコードビューが同期します。

戻り値

なし

例

```
dom.source.refreshVariableCodeHints();
```

dom.source.removeComment()

対応バージョン
Dreamweaver 8

説明
この関数は、コメントを削除します。引数を指定しない場合、サーバーサイドインクルードおよび Dreamweaver 独自のコメントを除くすべての種類のコメントを、現在の選択範囲から削除します。コメントがネストされている場合、外側のコメントだけを削除します。現在の選択範囲がない場合、カーソルがある行の最初の行コメントだけを削除します。引数を指定した場合、この関数は、一致するコメントが他の種類のコメントの内部にネストされている場合でも *beforeText* と *afterText* の各引数に指定された値と一致するコメントだけを削除します。

引数
beforeText、*afterText*

- *beforeText* 引数はオプションです。この引数には、選択範囲から削除するコメントの先頭を識別するテキストを指定します。または、*afterText* 引数の値がヌルの場合は、現在の選択範囲から削除する行コメントの種類を指定します。
- オプションの *afterText* 引数には、選択範囲から削除するコメントの末尾を識別するテキストを指定します。

戻り値
なし

例
次の例では、HTML コメントを削除します。

```
dw.getDocumentDOM().source.removeComment('<!--', '-->')
```

dreamweaver.htmlInspector.getShowHiddenCharacters()

対応バージョン
Dreamweaver 8

説明
この関数は、コードインスペクタのコードビューで、ホワイトスペースを表す特殊文字が表示されているかどうかを判別します。

引数
なし

戻り値
ブール値。非表示の文字が表示される場合は `true`、そうでない場合は `false`。

例
次の例では、コードインスペクタで、ホワイトスペースを表す特殊文字の表示が最初にオンになっていた場合に、特殊文字の表示をオフにします。

```
if (dreamweaver.htmlinspector.getShowHiddenCharacters()){
    dreamweaver.htmlinspector.setShowHiddenCharacters(false);
    }
```

dreamweaver.htmlInspector.setShowHiddenCharacters()

対応バージョン

Dreamweaver 8

説明

この関数は、コードインスペクタのコードビューで、ホワイトスペースの特殊文字の表示と非表示を切り替えます。

引数

show

- 必須の *show* 引数はブール値で、ホワイトスペースの非表示の文字を表示するかどうかを示します。

戻り値

ブール値。非表示の文字が表示される場合は `true`、そうでない場合は `false`。

例

詳細については、175 ページの dreamweaver.htmlInspector.getShowHiddenCharacters() を参照してください。

第 13 章 : サイト

Adobe® Dreamweaver® CS3 のサイト関数は、Web サイトの管理に関連する操作を行います。サイト関数が実行する操作には、レポートのカスタマイズ、新しいサイトの定義、ファイルのチェックイン / チェックアウト、サイトに対する検証の実行などがあります。

本章では関数をグループ分けして、それぞれを以下の項で説明します。

レポート関数

レポート関数を使用すると、レポート機能にアクセスして、レポート処理を開始、監視、およびカスタマイズすることができます。詳細については、『Dreamweaver 拡張ガイド』の「レポート」を参照してください。

dreamweaver.isReporting()

対応バージョン
Dreamweaver 4

説明
レポート処理が現在実行中かどうかをチェックします。

引数
なし

戻り値
処理が実行中の場合は `true`、そうでない場合は `false` のブール値。

dreamweaver.showReportsDialog()

対応バージョン
Dreamweaver 4

説明
[レポート] ダイアログボックスを開きます。

引数
なし

戻り値
なし

サイト関数

サイト関数は、サイトファイルやサイトマップ内のファイルに関する操作を行います。これらの関数を使用すると、以下のタスクを実行できます。

- ファイル間のリンクの作成
- ファイルの GET、PUT、チェックイン、チェックアウト
- ファイルの選択、選択解除
- ファイルの作成、削除
- ユーザーが定義したサイトに関する情報の取得
- サイト情報の読み込み、書き出し

dom.getSiteURLPrefixFromDoc()

対応バージョン

Dreamweaver 8

説明

この関数は、[サイト定義] ダイアログボックスの [ローカル情報] セクションで定義される HTTP アドレスから抽出したサイトの URL 接頭辞を取得します。

引数

なし

戻り値

サイトの URL 接頭辞を指定するストリング。

例

次の例では、現在のドキュメントのサイトの URL 接頭辞を取得します。

```
var currentDOM = dw.getDocumentDOM();
var sitePrefix = dom.getSiteURLPrefixFromDoc();
```

dom.localPathToSiteRelative()

対応バージョン

Dreamweaver 8

説明

ローカルファイルパスをサイト相対 URI 参照に変換します。

引数

localFilePath

- *localFilePath* 属性 (必須) には、ローカルコンピュータ上のローカルファイルへのパスを含むストリングを指定します。

戻り値

サイト相対 URI を指定するストリング。

例

次の例では、[サイト定義] ダイアログボックスの [ローカル情報] セクションに指定した HTTP アドレスおよびサイトマッピングに基づいて、"/myWebApp/myFile.cfm" を返します。

```
var siteRelativeURI = site.localPathToSiteRelative("C:¥Inetpub¥wwwroot¥siteA¥myFile.cfm")
```

dom.siteRelativeToLocalPath()

対応バージョン

Dreamweaver 8

説明

サイト相対 URI 参照をローカルファイルパスに変換します。

引数

siteRelativeURI

- *siteRelativeURI* 属性 (必須) には、サイト相対 URI を含むストリングを指定します。

戻り値

ローカルコンピュータ上のローカルファイルへのパスを指定するストリング。

例

次の例では、`var filePath = siteRelativeToLocalPath("/myWebApp/myFile.xml");`

[サイト定義] ダイアログボックスの [ローカル情報] セクションで指定した HTTP アドレスおよびサイトマッピングに基づいて、"C:¥Inetpub¥wwwroot¥siteA¥myFile.xml" を返します。

dreamweaver.compareFiles()

対応バージョン

Dreamweaver 8

説明

この関数は、[環境設定] ダイアログボックスの [差分] セクションでユーザーがインストールしたファイル比較ツール起動します。

引数

file1、*file2*

- *file1* 属性 (必須) には、比較する最初のファイルへのフルパスを含むストリングを指定します。
- file2 属性 (必須) には、比較する 2 番目のファイルへのフルパスを含むストリングを指定します。

戻り値

なし

例

次の例では、2 つのファイル "red.htm" と "blue.htm" を比較します。

```
dw.compareFiles(hc:¥data¥red.htm", "e:¥data¥blue.htm");
```

dreamweaver.loadSitesFromPrefs()

対応バージョン
Dreamweaver 4

説明
すべてのサイトのサイト情報を、システムレジストリ (Windows) または Dreamweaver の環境設定ファイル (Macintosh) から Dreamweaver にロードします。サイトがリモートサーバーに接続している場合にこの関数を呼び出すと、そのサイトは自動的に切断されます。

引数
なし

戻り値
なし

dreamweaver.saveSitesToPrefs()

対応バージョン
Dreamweaver 4

説明
ユーザーが定義した各サイトのすべてのサイト情報を、システムレジストリ (Windows) または Dreamweaver の環境設定ファイル (Macintosh) に保存します。

引数
なし

戻り値
なし

dreamweaver.siteSyncDialog.compare()

対応バージョン
Dreamweaver 8

説明
この関数は、[環境設定] ダイアログボックスの [ファイルの比較] カテゴリで指定したファイル比較アプリケーションを実行し、ローカルサイトとリモートサイトの選択したファイルを比較します。

引数
なし

戻り値
なし

イネーブラ
詳細については、430 ページの dreamweaver.siteSyncDialog.canCompare() を参照してください。

dreamweaver.siteSyncDialog.markDelete()

対応バージョン
Dreamweaver 8

説明
この関数は、[同期] ダイアログボックスで選択した項目に対するアクションを [削除] に変更します。

引数
なし

戻り値
なし

イネーブラ
詳細については、430 ページの dreamweaver.siteSyncDialog.canMarkDelete() を参照してください。

dreamweaver.siteSyncDialog.markGet()

対応バージョン
Dreamweaver 8

説明
この関数は、[同期] ダイアログボックスで選択した項目に対するアクションを [GET] に変更します。

引数
なし

戻り値
なし

イネーブラ
詳細については、430 ページの dreamweaver.siteSyncDialog.canMarkGet() を参照してください。

dreamweaver.siteSyncDialog.markIgnore()

対応バージョン
Dreamweaver 8

説明
この関数は、[同期] ダイアログボックスで選択した項目に対するアクションを [無視] に変更します。

引数
なし

戻り値
なし

イネーブラ
詳細については、431 ページの dreamweaver.siteSyncDialog.canMarkIgnore() を参照してください。

dreamweaver.siteSyncDialog.markPut()

対応バージョン
Dreamweaver 8

説明
この関数は、[同期] ダイアログボックスで選択した項目に対するアクションを [PUT] に変更します。

引数
なし

戻り値
なし

イネーブラ
詳細については、431 ページの dreamweaver.siteSyncDialog.canMarkPut() を参照してください。

dreamweaver.siteSyncDialog.markSynced()

対応バージョン
Dreamweaver 8

説明
この関数は、[同期] ダイアログボックスで選択した項目に対するアクションを [同期] に変更します。

引数
なし

戻り値
なし

イネーブラ
詳細については、431 ページの dreamweaver.siteSyncDialog.canMarkSynced() を参照してください。

dreamweaver.siteSyncDialog.toggleShowAllFiles()

対応バージョン
Dreamweaver 8

説明
この関数を使用すると、リモートサイト上とローカルサイト上で同じであると Dreamweaver が見なすファイルを [同期] プレビューダイアログボックスに表示できます。[すべてのファイルを表示] チェックボックスがオンのときにこの関数が呼び出されると、チェックボックスはオフになります。逆に、[すべてのファイルを表示] チェックボックスがオフのときにこの関数が呼び出されると、チェックボックスはオンになります。

引数
なし

戻り値
なし

site.addLinkToExistingFile()

対応バージョン
Dreamweaver 3

説明
[HTML ファイルの選択] ダイアログボックスを開き、ユーザーにファイルの選択を求め、選択されているドキュメントからそのファイルへのリンクを作成します。

引数
なし

戻り値
なし

イネーブラ
詳細については、433 ページの site.canAddLink() を参照してください。

site.addLinkToNewFile()

対応バージョン
Dreamweaver 3

説明
[新規ファイルへリンク] ダイアログボックスを開き、ユーザーに新規ファイルの詳細を指定するように求め、選択されているドキュメントからそのファイルへのリンクを作成します。

引数
なし

戻り値
なし

イネーブラ
詳細については、433 ページの site.canAddLink() を参照してください。

site.changeLinkSitewide()

対応バージョン
Dreamweaver 3

説明
[サイト全体のリンクの変更] ダイアログボックスを開きます。

引数
なし

戻り値
なし

site.changeLink()

対応バージョン
Dreamweaver 3

説明
リンクを作成する新規ファイルをユーザーが選択できるように、[HTML ファイルの選択] ダイアログボックスを開きます。

引数
なし

戻り値
なし

イネーブラ
詳細については、433 ページの site.canChangeLink() を参照してください。

site.checkIn()

対応バージョン
Dreamweaver 3

説明
選択したファイルをチェックインして、依存ファイルを次のいずれかの方法で処理します。

- [サイト] の FTP 環境設定で [PUT/ チェックインでダイアログを表示] を選択した場合は、[依存ファイル] ダイアログボックスが表示されます。
- 以前に [依存ファイル] ダイアログボックスで [次からこのメッセージを表示しない] オプションを選択していて、[はい] をクリックした場合、ダイアログボックスは表示されずに依存ファイルがアップロードされます。
- 以前に [依存ファイル] ダイアログボックスで [次からこのメッセージを表示しない] オプションを選択していて、[いいえ] をクリックした場合、ダイアログボックスは表示されず、依存ファイルもアップロードされません。

引数
`siteOrURL`

- *`siteOrURL`* 引数には、この関数が [サイト] パネルの選択範囲に対して動作することを示すキーワード `"site"`、または 1 つのファイルの URL を指定する必要があります。

戻り値
なし

イネーブラ
詳細については、433 ページの site.canCheckIn() を参照してください。

site.checkLinks()

対応バージョン
Dreamweaver 3

説明
[リンクチェック] ダイアログボックスを開き、指定されたファイル内のリンクをチェックします。

引数

scopeOfCheck

- *scopeOfCheck* 引数には、リンクチェックのスコープを指定します。"document"、"selection"、"site" のいずれかの値を指定する必要があります。

戻り値

なし

site.checkOut()

対応バージョン

Dreamweaver 3

説明

選択したファイルをチェックアウトして、依存ファイルを次のいずれかの方法で処理します。

- [サイト] の FTP 環境設定で [GET/ チェックアウトでダイアログを表示] を選択した場合は、[依存ファイル] ダイアログボックスが表示されます。
- 以前に [依存ファイル] ダイアログボックスで [次からこのメッセージを表示しない] オプションを選択していて、[はい] をクリックした場合、ダイアログボックスは表示されずに依存ファイルがダウンロードされます。
- 以前に [依存ファイル] ダイアログボックスで [次からこのメッセージを表示しない] オプションを選択していて、[いいえ] をクリックした場合、ダイアログボックスは表示されず、依存ファイルもダウンロードされません。

引数

siteOrURL

- *siteOrURL* 引数には、この関数が [サイト] パネルの選択範囲に対して動作することを示すキーワード "site"、または 1 つのファイルの URL を指定する必要があります。

戻り値

なし

イネーブラ

詳細については、434 ページの site.canCheckOut() を参照してください。

site.checkTargetBrowsers()

対応バージョン

Dreamweaver 3

説明

選択したファイルに対してターゲットブラウザチェックを実行します。

引数

なし

戻り値

なし

site.cloak()

対応バージョン
Dreamweaver MX

説明
[サイト] パネルまたは指定フォルダ内の現在の選択範囲をクロークします。

引数
siteOrURL

siteOrURL 引数には、以下の 2 つの値のうち、いずれかを含める必要があります。

- キーワード `"site"`。これは、`cloak()` が [サイト] パネル内の選択範囲に対して動作することを示します。
- 特定フォルダの URL。これは、`cloak()` が指定したフォルダとそのすべてのコンテンツに対して動作することを示します。

戻り値
なし

イネーブラ
詳細については、434 ページの site.canCloak() を参照してください。

site.compareFiles()

対応バージョン
Dreamweaver 8

説明
この関数は、ファイル比較ツール統合アプリケーションを起動し、2 つのファイルを比較します。

引数
url

url 引数 (必須) には、以下の 2 つの値のうち、いずれかを含める必要があります。

- キーワード `"site"`。これは、`compare()` が [サイト] パネル内の選択範囲に対して動作することを示します。
- リモートバージョンと比較するローカルファイルの URL。

戻り値
比較が成功した場合は `true`、そうでない場合は `false` のブール値。

イネーブラ
詳細については、434 ページの site.canCompareFiles() を参照してください。

例
次の例では、[サイト] パネルで選択したファイルとそのリモートバージョンを比較します。

```
site.compareFiles("site");
```

site.defineSites()

対応バージョン
Dreamweaver 3

説明
この関数は、[サイトの編集] ダイアログボックスを開きます。

引数
なし

戻り値
なし

site.deleteSelection()

対応バージョン
Dreamweaver 3

説明
選択したファイルを削除します。

引数
なし

戻り値
なし

site.deployFilesToTestingServerBin()

対応バージョン
Dreamweaver MX

説明
指定されたファイル (複数可) を、テストサーバーの "bin" フォルダに配置します。サポートファイルの配置に必要な設定が現在のサイトに定義されていない場合、この関数は [テストサーバーにサポートファイルを配置] ダイアログボックスを開きます。

引数
filesToDeploy

- *filesToDeploy* 引数には、Dreamweaver で配置するファイル名の配列を指定します。

戻り値
ファイルの配置に成功した場合は `true`、失敗した場合は `false` のブール値。

例
次の例は、"image1.jpg" ファイルと "script1.js" ファイルをテストサーバーの "bin" フォルダに配置します。

```
site.deployFilesToTestingServerBin("image1.jpg", "script1.js");
```

site.displaySyncInfoForFile()

対応バージョン

Dreamweaver CS3

説明

渡されたパラメータに対応するファイルのローカル時間、リモート時間、およびテスト時間を示すダイアログボックスを表示します。この情報は、"dwsync.xml" 同期ファイルに格納されています。

ダイアログボックスには、以下の 4 つの時間が表示されます。

- ローカルリモート時間 : ローカルファイルについて、リモートサーバーに対する最後の [GET] コマンドまたは [PUT] コマンドのタイムスタンプを示します。
- リモート時間 : リモートサーバーのファイルについて、リモートサーバーに対する最後の [GET] コマンドまたは [PUT] コマンドのタイムスタンプを示します。
- ローカルテスト時間 : ローカルファイルについて、テストサーバーに対する最後の [GET] コマンドまたは [PUT] コマンドのタイムスタンプを示します。
- テスト時間 : テストサーバーのファイルについて、テストサーバーに対する最後の [GET] コマンドまたは [PUT] コマンドのタイムスタンプを示します。

dwsync.xml ファイルに、ファイルの情報が含まれていない場合、情報が利用できないことを示すメッセージが表示されます。この xml ファイルに時間が設定されている場合、そのロケールの日付 / 時刻形式で表示されます (6/24/05 2:43pm など)。ファイルのエントリに時間が設定されていない場合、ダッシュ記号 (-) が表示されます。

この関数は、'site' が指定された場合は [ローカルファイルビュー] で選択されたファイルに対して動作し、URL が指定された場合は、ローカルの URL に対応するファイルに対して動作します。

引数

path、'site'

- *path* は、ローカルファイルの URL です。
- 'site' は、この関数が [サイト] パネルで選択されたファイルを使用することを示します。

戻り値

なし

イネーブラ

詳細については、435 ページの site.canDisplaySyncInfoForFile() を参照してください。

site.editColumns()

説明

この関数は、[表示列] セクションを含む [サイトの編集] ダイアログボックスを表示します。

引数

なし

戻り値

なし

site.exportSite()

対応バージョン

Dreamweaver MX

説明

Dreamweaver サイトを XML ファイルに書き出します。このファイルを別の Dreamweaver インスタンスに読み込むと、前のサイトを複製することができます。

[サイト定義] ダイアログボックスに表示されるすべての情報は、XML ファイルに保存されます。このファイルには、クロークされたフォルダのリストや、デフォルトのドキュメントタイプに関する情報が含まれます。例外は、FTP アクセスを設定すると、ユーザーログインとパスワードを省略できることです。

引数

siteName

- *siteName* 引数には、書き出すサイトを指定します。*siteName* に空白のストリングを指定すると、現在のサイトが書き出されます。

戻り値

指定されたサイトが存在し、XML ファイルが正常に書き出された場合は `true`、それ以外の場合は `false` のブール値。

例

次の例は、サイトを書き出したときに Dreamweaver で作成されるサンプルの XML ファイルを示しています。

```
<?xml version="1.0" ?>
<site>
    <localinfo
        sitename="DW00"
        localroot="C:¥Documents and Settings¥jlondon¥Desktop¥DWServer¥"
        imagefolder="C:¥Documents and Settings¥jlondon¥Desktop¥DWServer¥Images¥"
        spacerfilepath=""
        refreshlocal="TRUE"
        cache="FALSE"
        httpaddress="http://" curserver="webserver" />
    <remoteinfo
        accesstype="ftp"
        host="dreamweaver"
        remoteroot="kojak/"
        user="dream"
        checkoutname="Jay"
        emailaddress="jay@Adobe.com"
        usefirewall="FALSE"
        usepasv="TRUE"
        enablecheckin="TRUE"
        checkoutwhenopen="TRUE" />
    <designnotes
        usedesignnotes="TRUE"
        sharedesignnotes="TRUE" />
    <sitemap
        homepage="C:¥Documents and Settings¥jlondon¥Desktop¥DWServer¥Untitled-2.htm"
        pagesperrow="200" columnwidth="125" showdependentfiles="TRUE"
        showpagetitles="FALSE" showhiddenfiles="TRUE" />
    <fileviewcolumns sharecolumns="TRUE">
        <column name="Local Folder"
            align="left" show="TRUE" share="FALSE" builtin="TRUE"
            localwidth="180" remotewidth="180" />
        <column name="Notes"
```

```
            align="center" show="TRUE" share="FALSE" builtin="TRUE"
            localwidth="36" remotewidth="36" />
        <column name="Size"
            align="right" show="TRUE" share="FALSE" builtin="TRUE"
            localwidth="-2" remotewidth="-2" />
        <column name="Type"
            align="left" show="TRUE" share="FALSE" builtin="TRUE"
            localwidth="60" remotewidth="60" />
        <column name="Modified"
            align="left" show="TRUE" share="FALSE" builtin="TRUE"
            localwidth="102" remotewidth="102" />
        <column name="Checked Out By"
            align="left" show="TRUE" share="FALSE" builtin="TRUE"
            localwidth="50" remotewidth="50" />
        <column name="Status" note="status"
            align="left" show="TRUE" share="FALSE" builtin="FALSE"
            localwidth="50" remotewidth="50" />
    </fileviewcolumns>
    <appserverinfo
        servermodel="ColdFusion"
        urlprefix="http://dreamweaver/kojak/"
        serverscripting="CFML"
        serverpageext=""
        connectionsmigrated="TRUE"
        useUD4andUD5pages="TRUE"
        defaultdoctype=""
        accesstype="ftp"
        host="dreamweaver"

        remoteroot="kojak/"
        user="dream"
        usefirewall="FALSE"
        usepasv="TRUE" />
    <cloaking enabled="TRUE" patterns="TRUE">
        <cloakedfolder folder="databases/" />
        <cloakedpattern pattern=".png" />
        <cloakedpattern pattern=".jpg" />
        <cloakedpattern pattern=".jpeg" />
    </cloaking>
</site>
```

site.findLinkSource()

対応バージョン

Dreamweaver 3

説明

選択したリンクまたは依存ファイルを含むファイルを開き、その依存ファイルへのリンクまたは参照のテキストをハイライト表示します。この関数は、サイトマップビューのファイルだけに使用できます。

引数

なし

戻り値

なし

イネーブラ

詳細については、435 ページの site.canFindLinkSource() を参照してください。

site.get()

対応バージョン
Dreamweaver 3

説明
指定したファイルを取得して、依存ファイルを以下のいずれかの方法で処理します。

- [サイト] の FTP 環境設定で [GET/ チェックアウトでダイアログを表示] を選択した場合は、[依存ファイル] ダイアログボックスが表示されます。
- 以前に [依存ファイル] ダイアログボックスで [次からこのメッセージを表示しない] オプションを選択していて、[はい] をクリックした場合、ダイアログボックスは表示されずに依存ファイルがダウンロードされます。
- 以前に [依存ファイル] ダイアログボックスで [次からこのメッセージを表示しない] オプションを選択していて、[いいえ] をクリックした場合、ダイアログボックスは表示されず、依存ファイルもダウンロードされません。

引数
`siteOrURL`

- *`siteOrURL`* 引数には、この関数が [サイト] パネルの選択範囲に対して動作することを示すキーワード `"site"`、または 1 つのファイルの URL を指定する必要があります。

戻り値
なし

イネーブラ
詳細については、436 ページの site.canGet() を参照してください。

site.getAppServerAccessType()

対応バージョン
Dreamweaver MX

説明
現在のサイトのアプリケーションサーバーにある、すべてのファイルに使用されるアクセスメソッドを返します。現在のサイトとは、フォーカスが現在置かれているドキュメントに関連付けられているサイトです。ドキュメントにフォーカスが置かれていない場合は、Dreamweaver で開いているサイトが使用されます。

注意 : この関数は、*ColdFusion Component Explorer* によって使用されます。詳細については、192 ページの site.getAppServerPathToFiles() および 195 ページの site.getLocalPathToFiles() を参照してください。

引数
なし

戻り値
次のストリングのいずれかを返します。

- `"none"`
- `"local/network"`
- `"ftp"`
- `"source_control"`

site.getAppServerPathToFiles()

対応バージョン
Dreamweaver MX

説明
現在のサイトに定義されているアプリケーションサーバー上のリモートファイルへのパスを判別します。現在のサイトとは、フォーカスが現在置かれているドキュメントに関連付けられているサイトです。ドキュメントにフォーカスが置かれていない場合は、Dreamweaver で開いているサイトが使用されます。

注意 : この関数は、*ColdFusion Component Explorer* によって使用されます。詳細については、191 ページの site.getAppServerAccessType() および 195 ページの site.getLocalPathToFiles() を参照してください。

引数
なし

戻り値
アプリケーションサーバーファイルへのアクセスタイプが `local/network` の場合はパス。そうでない場合は空白のストリング。

site.getAppURLPrefixForSite()

対応バージョン
Dreamweaver MX

説明
[サイト定義] ダイアログボックスの [ローカル情報] セクションで定義される HTTP アドレスから抽出した URL 接頭辞の値を取得します。これは、http:// ホスト名 : ポート番号 / の後に来るパスです。

引数
{ siteName }

siteName 引数 (オプション) には、URL 接頭辞を取得するサイトの名前を指定します。サイトを指定しない場合は、現在のサイトの URL 接頭辞を取得します。

戻り値
現在選択されているサイトの URL 接頭辞を含むストリング。

例
```
var sitePrefix = site.getAppURLPrefixForSite();
```

site.getCheckOutUser()

対応バージョン
Dreamweaver 3

説明
現在のサイトに関連付けられているログイン名とチェックアウト名を取得します。

引数
なし

戻り値

ログイン名とチェックアウト名が定義されている場合は、それらを含むストリング。チェックイン / チェックアウトが使用不可の場合は空白のストリング。

例

`site.getCheckOutUser()` を呼び出すと、`"denise (deniseLaptop)"` などが返されます。チェックアウト名が指定されていない場合は、ログイン名だけが返されます (`"denise"` など)。

site.getCheckOutUserForFile()

対応バージョン

Dreamweaver 3

説明

指定したファイルをチェックアウトしたユーザーのログイン名とチェックアウト名を取得します。

引数

`fileName`

- *`fileName`* 引数には、ログイン名とチェックアウト名を調べるファイルへのパスを file:// URL 形式で指定します。

戻り値

指定したファイルをチェックアウトしたユーザーのログイン名とチェックアウト名を含むストリング。または、ファイルがチェックアウトされていない場合は空白のストリング。

例

`site.getCheckOutUserForFile("file://C:¥sites¥avocado8¥index.html")` を呼び出すと、`"denise (deniseLaptop)"` などの戻り値が返されます。チェックアウト名が指定されていない場合は、ログイン名だけが返されます (`"denise"` など)。

site.getCloakingEnabled()

対応バージョン

Dreamweaver MX

説明

現在のサイトのクロークが有効かどうかを判別します。

引数

なし

戻り値

現在のサイトのクロークが有効な場合は `true`、それ以外の場合は `false` のブール値。

site.getConnectionState()

対応バージョン

Dreamweaver 3

説明

現在の接続状態を取得します。

引数
なし

戻り値
リモートサイトが接続されているかどうかを示すブール値。

イネーブラ
詳細については、435 ページの site.canConnect() を参照してください。

site.getCurrentSite()

対応バージョン
Dreamweaver 3

説明
現在のサイトを取得します。

引数
なし

戻り値
現在のサイトの名前を含むストリング。

例
複数のサイトが定義されている場合に `site.getCurrentSite()` を呼び出すと、[サイト] パネルの [現在のサイト] リストに現在表示されているサイトが返されます。

site.getFocus()

対応バージョン
Dreamweaver 3

説明
[サイト] パネルでフォーカスされているペインを判別します。

引数
なし

戻り値
次のストリングのいずれかを返します。

- `"local"`
- `"remote"`
- `"site map"`

site.getLinkVisibility()

対応バージョン
Dreamweaver 3

説明
サイトマップで選択したすべてのリンクが表示されている (つまり非表示に指定されていない) かどうかをチェックします。

引数
なし

戻り値
選択したすべてのリンクが表示されている場合は `true`、それ以外の場合は `false` のブール値。

site.getLocalPathToFiles()

対応バージョン
Dreamweaver MX

説明
現在のサイトに定義されているローカルファイルへのパスを判別します。現在のサイトとは、フォーカスが現在置かれているドキュメントに関連付けられているサイトです。ドキュメントにフォーカスが置かれていない場合は、Dreamweaver で開いているサイトが使用されます。

注意 : この関数は、*ColdFusion Component Explorer* によって使用されます。詳細については、191 ページの site.getAppServerAccessType() および 192 ページの site.getAppServerPathToFiles() を参照してください。

引数
なし

戻り値
現在のサイトのローカルコンピュータに常駐するファイルのパス。

site.getSelection()

対応バージョン
Dreamweaver 3

説明
[サイト] パネルで現在選択されているファイルを判別します。

引数
なし

戻り値
選択したファイルおよびフォルダへのパスを file:// URL 形式で表すストリングの配列。選択したファイルまたはフォルダがない場合は空白の配列。

site.getSiteForURL()

対応バージョン
Dreamweaver MX

説明
特定のファイルに関連付けられているサイトがある場合は、そのサイトの名前を取得します。

引数

fileURL

- *fileURL* 引数には、指定したファイルの完全修飾 URL (ストリング *"file://"* を含む) を指定します。

戻り値

指定したファイルが存在するサイトがある場合は、そのサイトの名前を含むストリング。指定したファイルが定義されているサイトにない場合は、空白のストリング。

site.getSites()

対応バージョン

Dreamweaver 3

説明

定義されているサイトのリンクを取得します。

引数

なし

戻り値

定義されているサイトの名前を表すストリングの配列。サイトが定義されていない場合は空白の配列。

site.getSiteURLPrefix()

対応バージョン

Dreamweaver 8

説明

[ローカル情報] セクションで定義される HTTP アドレスから抽出したサイトの URL 接頭辞を取得します。

引数

なし

戻り値

サイトの URL 接頭辞を含むストリング。

例

```
sitePrefix = getSiteURLPrefix();
```

site.importSite()

対応バージョン

Dreamweaver MX

説明

XML ファイルから Dreamweaver サイトを作成します。読み込み時に、<localinfo> エレメントの localroot 属性で指定されているフォルダがローカルコンピュータに存在しない場合は、別のローカルルートフォルダを指定するように求められます。Dreamweaver は、<localinfo> エレメントの imagefolder 属性で指定されているデフォルトのイメージフォルダを検索する場合と同じように動作します。

引数

fileURL

- *fileURL* 引数には、XML ファイルの URL を含むストリングを指定します。Dreamweaver は、この XML ファイルを使用して新規サイトを作成します。fileURL に空白のストリングを指定すると、読み込む XML ファイルを選択するように求められます。

戻り値

指定された XML ファイルが存在し、サイトが正常に作成された場合は true、それ以外の場合は false のブール値。

site.invertSelection()

対応バージョン

Dreamweaver 3

説明

サイトマップの選択範囲を反転します。

引数

なし

戻り値

なし

site.isCloaked()

対応バージョン

Dreamweaver MX

説明

[サイト] パネルまたは指定フォルダ内の現在の選択範囲がクロークされているかどうかを判別します。

引数

siteOrURL

- *siteOrURL* 引数には、以下の 2 つの値のうち、いずれかを含める必要があります。
  - キーワード "site"。これは、isCloaked() 関数で [サイト] パネル内の選択範囲をテストすることを示します。
  - 特定のフォルダのファイル URL。これは、isCloaked() でそのフォルダをテストすることを示します。

戻り値

指定したオブジェクトがクロークされている場合は true、そうでない場合は false のブール値。

site.locateInSite()

対応バージョン

Dreamweaver 3

説明

[サイト] パネルの指定ペインで指定したファイルを検索し、選択します。

引数

localOrRemote、*siteOrURL*

- *localOrRemote* 引数には、"local" か "remote" のいずれかを指定する必要があります。
- *siteOrURL* 引数には、この関数が [サイト] パネルの選択範囲に対して、または単一ファイルの URL に対して動作することを表すキーワード "site" を指定する必要があります。

戻り値

なし

イネーブラ

詳細については、436 ページの site.canLocateInSite() を参照してください。

site.makeEditable()

対応バージョン

Dreamweaver 3

説明

選択したファイルの読み取り専用フラグをオフにします。

引数

なし

戻り値

なし

イネーブラ

詳細については、437 ページの site.canMakeEditable() を参照してください。

site.makeNewDreamweaverFile()

対応バージョン

Dreamweaver 3

説明

[サイト] パネルで新規 Dreamweaver ファイルを作成します。新規ファイルは、最初に選択したファイルまたはフォルダと同じフォルダに保存されます。

引数

なし

戻り値

なし

イネーブラ

詳細については、437 ページの site.canMakeNewFileOrFolder() を参照してください。

site.makeNewFolder()

対応バージョン
Dreamweaver 3

説明
[サイト] パネルで新規フォルダを作成します。新規フォルダは、最初に選択したファイルまたはフォルダと同じフォルダに保存されます。

引数
なし

戻り値
なし

イネーブラ
詳細については、437 ページの site.canMakeNewFileOrFolder() を参照してください。

site.newHomePage()

対応バージョン
Dreamweaver 3

説明
ユーザーが新規のホームページを作成できるように、[新規ホームページ] ダイアログボックスを開きます。

注意 : この関数は、サイトマップビューのファイルだけに使用できます。

引数
なし

戻り値
なし

site.newSite()

対応バージョン
Dreamweaver 3

説明
名前のない新規サイトに対して、[サイト定義] ダイアログボックスを開きます。

引数
なし

戻り値
なし

site.open()

対応バージョン
Dreamweaver 3

説明
[サイト] パネルで現在選択されているファイルを開きます。フォルダが選択されている場合、サイトファイルビューでこのフォルダが展開表示されます。

引数
なし

戻り値
なし

イネーブラ
詳細については、437 ページの site.canOpen() を参照してください。

site.put()

対応バージョン
Dreamweaver 3

説明
選択したファイルを PUT して依存ファイルを次のいずれかの方法で処理します。

- [サイト] の FTP 環境設定で [PUT/ チェックインでダイアログを表示] を選択した場合は、[依存ファイル] ダイアログボックスが表示されます。
- 以前に [依存ファイル] ダイアログボックスで [次からこのメッセージを表示しない] オプションを選択していて、[はい] をクリックした場合、ダイアログボックスは表示されずに依存ファイルがアップロードされます。
- 以前に [依存ファイル] ダイアログボックスで [次からこのメッセージを表示しない] オプションを選択していて、[いいえ] をクリックした場合、ダイアログボックスは表示されず、依存ファイルもアップロードされません。

引数
siteOrURL

- *siteOrURL* 引数には、この関数が [サイト] パネルの選択範囲に対して動作することを示すキーワード `"site"`、または 1 つのファイルの URL を指定する必要があります。

戻り値
なし

イネーブラ
詳細については、437 ページの site.canPut() を参照してください。

site.recreateCache()

対応バージョン
Dreamweaver 3

説明
現在のサイトのキャッシュを再作成します。

引数
なし

戻り値
なし

イネーブラ
詳細については、438 ページの site.canRecreateCache() を参照してください。

site.refresh()

対応バージョン
Dreamweaver 3

説明
[サイト] パネルの指定した側にあるファイルリストを更新します。

引数
whichSide

- *whichSide* 引数には、"local" か "remote" を指定する必要があります。サイトマップにフォーカスがある場合に *whichSide* を "local" に指定すると、サイトマップが更新されます。

戻り値
なし

イネーブラ
詳細については、438 ページの site.canRefresh() を参照してください。

site.remoteIsValid()

対応バージョン
Dreamweaver 3

説明
リモートサイトが有効かどうかを判別します。

引数
なし

戻り値
リモートサイトが定義されているかどうか、および、サーバータイプがローカルまたはネットワークの場合にはドライブがマウントされているかどうかを示すブール値。

site.removeLink()

対応バージョン
Dreamweaver 3

説明
選択したリンクを、サイトマップでその上にあるドキュメントから削除します。

引数
なし

戻り値
なし

イネーブラ
詳細については、439 ページの site.canRemoveLink() を参照してください。

site.renameSelection()

対応バージョン
Dreamweaver 3

説明
ユーザーがファイル名を変更できるように、選択したファイルの名前をテキストフィールドに変換します。複数のファイルが選択されている場合、この関数は最後に選択したファイルに対して動作します。

引数
なし

戻り値
なし

site.runValidation()

対応バージョン
Dreamweaver MX

説明
サイト全体またはハイライト表示されている項目だけに対してバリデータを実行します。

引数
selection

- *selection* 引数は、ハイライト表示されている項目だけをバリデータでチェックするためのパラメータです。これを指定しない場合は、現在のサイト全体がチェックされます。

戻り値
なし

site.saveAsImage()

対応バージョン
Dreamweaver 3

説明
ユーザーがサイトマップをイメージとして保存できるように、[新規保存] ダイアログボックスを開きます。

引数

fileType

- *fileType* 引数には、保存するイメージのタイプを指定します。有効な値は、Windows では "bmp" と "png"、Macintosh では "pict" と "jpeg" です。この引数を指定しない場合や、使用中のプラットフォームで無効な値を指定すると、Windows では "bmp"、Macintosh では "pict" がデフォルト値として使用されます。

戻り値

なし

site.selectAll()

対応バージョン

Dreamweaver 3

説明

アクティブなビューにあるすべてのファイル (サイトマップまたはサイトファイルのいずれか) を選択します。

引数

なし

戻り値

なし

site.selectHomePage()

対応バージョン

Dreamweaver 3

説明

ユーザーが新規のホームページを選択できるように、[ファイルを開く] ダイアログボックスを開きます。

注意 : この関数は、サイトマップビューのファイルだけに使用できます。

引数

なし

戻り値

なし

site.selectNewer()

対応バージョン

Dreamweaver 3

説明

[サイト] パネルの指定した側にある、すべての新しいファイルを選択します。

引数

whichSide

- *whichSide* 引数には、"local" か "remote" のいずれかを指定する必要があります。

戻り値
なし

イネーブラ
詳細については、439 ページの site.canSelectNewer() を参照してください。

site.serverActivity()

対応バージョン
Dreamweaver 8

説明
この関数は、現在 Dreamweaver がサーバーと交信しているかどうかを判別します。Dreamweaver が一度に実行できるサーバー操作は 1 つだけです。このため、この関数を使用して、サーバーとの対話が必要な機能を無効にするかどうかを判断できます。

引数
なし

戻り値
現在 Dreamweaver がサーバーと交信しているかどうかを示すブール値。

例
次の例では、サーバー操作がなく、Dreamweaver で現在のサイトが指定されている場合に、"menus.xml" ファイルからメニュー項目を表示します。

```
<menuitem name="Remove Connection Scripts" enabled="!site.serverActivity() &&
site.getCurrentSite() != ''" command="alert(MMDB.removeConnectionScripts())"
id="SiteOptionsSiteMenu_RemoveConnectionScripts" />
```

site.setAsHomePage()

対応バージョン
Dreamweaver 3

説明
サイトファイルビューで選択されているファイルを、このサイトのホームページに指定します。

引数
なし

戻り値
なし

site.setCloakingEnabled()

対応バージョン
Dreamweaver MX

説明
現在のサイトのクロークを有効にする必要があるかどうかを指定します。

引数

enable

- *enable* 引数には、クロークを有効にする必要があるかどうかを示すブール値を指定します。値 `true` を指定すると、現在のサイトのクロークが有効になります。値 `false` を指定すると、現在のサイトのクロークが無効になります。

戻り値

なし

site.setConnectionState()

対応バージョン

Dreamweaver 3

説明

現在のサイトの接続状態を設定します。

引数

bConnected

- *bConnected* 引数には、現在のサイトとの接続がある (`true`) か、ない (`false`) かを示すブール値を指定します。

戻り値

なし

site.setCurrentSite()

対応バージョン

Dreamweaver 3

説明

指定したサイトを [サイト] パネルのローカルペインで開きます。

引数

whichSite

- *whichSite* 引数には、定義されたサイトの名前を指定します。この名前は、[サイト] パネルの [現在のサイト] リストまたは [サイトの編集] ダイアログボックスに表示される名前と同じものです。

戻り値

なし

例

たとえば、avocado8、dreamcentral、および testsite という 3 つのサイトが定義されている場合に `site.setCurrentSite("dreamcentral");` を呼び出すと、dreamcentral が現在のサイトに指定されます。

site.setFocus()

対応バージョン

Dreamweaver 3

説明

[サイト] パネルの指定したペインにフォーカスを与えます。指定したペインが表示されていない場合は、ペインを表示してからフォーカスを与えます。

引数

whichPane

- *whichPane* 引数には、"local"、"remote"、または "site map" のいずれかのストリングを指定する必要があります。

戻り値

なし

site.setLayout()

対応バージョン

Dreamweaver 3

説明

[サイト定義] ダイアログボックスの [サイトマップレイアウト] 領域を開きます。

引数

なし

戻り値

なし

イネーブラ

詳細については、439 ページの site.canSetLayout() を参照してください。

site.setLinkVisibility()

対応バージョン

Dreamweaver 3

説明

現在のリンクの表示と非表示を切り替えます。

引数

bShow

- *bShow* 引数には、現在のリンクの非表示指定を削除するかどうかを示すブール値を指定します。

戻り値

なし

site.setSelection()

対応バージョン

Dreamweaver 3

説明

[サイト] パネルのアクティブなペインにあるファイルまたはフォルダを選択します。

引数

arrayOfURLs

- *arrayOfURLs* 引数には、現在のサイトのファイルまたはフォルダへのパスを file:// URL 形式で表すストリングの配列を指定します。

注意 : フォルダパスを指定する場合、最後のスラッシュ (/) は省略します。

戻り値

なし

site.siteRelativeToLocalPath()

対応バージョン

Dreamweaver 8

説明

サイト相対 URI リファレンスをローカルファイルパスに変換します。

引数

siteRelativeURI

- siteRelativeURI 属性 (必須) には、サイト相対 URI を含むストリングを指定します。

戻り値

ローカルコンピュータ上のローカルファイルへのパスを指定するストリング。

例

次の例では、`var filePath = site.siteRelativeToLocalPath("/myWebApp/myFile.xlml");`

[サイト定義] ダイアログボックスの [ローカル情報] で指定した HTTP アドレスおよびサイトマッピングに基づいて、"C:¥Inetpub¥wwwroot¥siteA¥myFile.xml" を返します。

site.synchronize()

対応バージョン

Dreamweaver 3

説明

[同期] ダイアログボックスを開きます。

引数

なし

戻り値

なし

イネーブラ

詳細については、440 ページの site.canSynchronize() を参照してください。

site.uncloak()

対応バージョン
Dreamweaver MX

説明
[サイト] パネルまたは指定フォルダ内の現在の選択範囲からクロークを解除します。

引数
siteOrURL

- *siteOrURL* 引数には、以下の値のいずれかを含める必要があります。
  - キーワード "site"。これは unCloak() 関数が [サイト] パネル内の選択範囲に対して動作することを示します。
  - 特定フォルダの URL。これは、unCloak() 関数が指定したフォルダとそのすべてのコンテンツに対して動作することを示します。

戻り値
なし

イネーブラ
詳細については、440 ページの site.canUncloak() を参照してください。

site.uncloakAll()

対応バージョン
Dreamweaver MX

説明
現在のサイトのすべてのフォルダからクロークを解除し、[クローク] の [次で終わるファイルをクロークする] チェックボックスをオフにします。

引数
なし

戻り値
なし

イネーブラ
詳細については、440 ページの site.canUncloak() を参照してください。

site.undoCheckOut()

対応バージョン
Dreamweaver 3

説明
指定したファイルに関連付けられているロックファイルをローカルサイトとリモートサイトから削除し、指定したファイルのローカルコピーをリモートコピーに置換します。

引数

siteOrURL

- *siteOrURL* 引数には、この関数が [サイト] パネルの選択範囲に対して動作することを示すキーワード "site"、または 1 つのファイルの URL を指定する必要があります。

戻り値

なし

イネーブラ

詳細については、441 ページの site.canUndoCheckOut() を参照してください。

site.viewAsRoot()

対応バージョン

Dreamweaver 3

説明

選択したファイルを一時的にサイトマップの一番上に移動します。

引数

なし

戻り値

なし

イネーブラ

詳細については、441 ページの site.canViewAsRoot() を参照してください。

第 14 章 : ドキュメント

Adobe® Dreamweaver® CS3 のドキュメント関数は、ユーザーが作業しているドキュメントに影響を与える操作を行います。ドキュメント関数が実行するタスクには、テーブルからレイヤーへの変換、"Configuration/Commands" フォルダにあるコマンドの実行、ファイル URL の参照、スペルチェックやページプロパティの設定、相対 URL から絶対 URL への変換、現在選択されてるノードの取得、ストリングに対する URL エンコードの実行、ドキュメントに対するトランスレータの実行などがあります。

本章では関数をグループ分けして、それぞれを以下の項で説明します。

変換関数

変換関数を使用して、テーブルをレイヤーに、レイヤーをテーブルに、カスケーディングスタイルシート (CSS) を HTML マークアップに変換します。これらの関数のビヘイビアは、[ファイル] メニューまたは [修正] メニューにある変換コマンドのビヘイビアとまったく同じです。

dom.convertLayersToTable()

対応バージョン

Dreamweaver 3

説明

[レイヤーをテーブルに変換] ダイアログボックスを開きます。

引数

なし

戻り値

なし

イネーブラ

詳細については、409 ページの dom.canConvertLayersToTable() を参照してください。

dom.convertTablesToLayers()

対応バージョン
Dreamweaver 3

説明
[テーブルをレイヤーに変換] ダイアログボックスを開きます。

引数
なし

戻り値
なし

イネーブラ
詳細については、409 ページの dom.canConvertTablesToLayers() を参照してください。

コマンド関数

コマンド関数を使用すると、"Configuration/Commands" フォルダのファイルを最大限に活用できます。コマンド関数は、[コマンド] メニューを管理する場合、および他のタイプの拡張機能ファイルからコマンドを呼び出す場合に使用します。

dreamweaver.editCommandList()

対応バージョン
Dreamweaver 3

説明
[コマンドリストの編集] ダイアログボックスを開きます。

引数
なし

戻り値
なし

dreamweaver.popupCommand() (非推奨)

対応バージョン
Dreamweaver 2 (3 では 212 ページの dreamweaver.runCommand() に置き換えられるため非推奨)

説明
指定したコマンドを実行します。ユーザーにとっては、メニューからダイアログボックスが関連付けられているコマンドを選択した場合にそのダイアログボックスが表示されるのと同じ結果になります。この関数によって、別の拡張機能ファイルからコマンドを呼び出すことができます。このとき、ユーザーがダイアログボックスを閉じるまで、他の編集操作を行うことはできません。

注意 : この関数は、*`objectTag()`* 関数の内部、コマンドファイルのスクリプト、またはプロパティインスペクタファイルから呼び出すことができます。

引数

commandFile

- *commandFile* 引数には "Configuration¥Commands" フォルダ内のコマンドファイルの名前を指定します。たとえば、"Format Table.htm" と指定します。

戻り値

なし

dreamweaver.runCommand()

対応バージョン

Dreamweaver 3

説明

指定されたコマンドを実行します。コマンドがメニューから選択された場合と同じように機能します。コマンドにダイアログボックスが関連付けられている場合は、ダイアログボックスが表示されます。ユーザーがそのダイアログボックスを閉じるまで、他の編集を行うことはできません。この関数によって、別の拡張機能ファイルからコマンドを呼び出すことができます。

注意 : この関数は、*objectTag()* 関数の内部、コマンドファイルのスクリプト、またはプロパティインスペクタファイルから呼び出すことができます。

引数

commandFile、*{commandArg1}*、*{commandArg2}*、*...{commandArgN}*

- *commandFile* 引数には、"Configuration¥Commands" フォルダにあるファイルの名前を指定します。
- *commandArg1*、*commandArg2* などの他の引数 (オプション) は、*commandFile* 引数の `receiveArguments()` 関数に渡されます。

戻り値

なし

例

テーブルのカスタムプロパティインスペクタを作成して、ユーザーがインスペクタのボタンをクリックすると [テーブルのフォーマット] コマンドを実行できるようにするには、ボタンの `onClick` イベントハンドラから次の関数を呼び出します。

```
function callFormatTable(){
    dreamweaver.runCommand('Format Table.htm');
}
```

ファイル操作関数

ファイル操作関数は、ドキュメント (XML および XHTML を含む) を開く、作成、保存などの操作を行います。また、既存の HTML ドキュメントの XHTML への変換や、外部ファイルへの CSS の書き出しを行います。これらの関数は、ファイルまたはフォルダの参照、テンプレートに基づくファイルの作成、ドキュメントを閉じる操作、および最近開いたファイルについての情報の取得などのタスクを実行します。

dom.cleanupXHTML()

対応バージョン

Dreamweaver MX

説明

この関数は、convertToXHTML() 関数に似ていますが、既存の XHTML ドキュメントをクリーンアップします。この関数は、ドキュメント内の選択範囲に対して実行できます。XHTML ドキュメント全体またはドキュメントの現在選択されている部分のシンタックスをクリーンアップするには、cleanupXHTML() 関数を実行します。

引数

bWholeDoc

- *bWholeDoc* 引数には、ブール値を指定します。値が true の場合、cleanupXHTML() 関数はドキュメント全体をクリーンアップします。それ以外の場合、この関数は選択範囲のみをクリーンアップします。

戻り値

以下のエレメントの個数を表す 6 つの整数から成る配列。

- Dreamweaver によって修正された XHTML エラー
- id 属性がなく、修正できない map エレメント
- type 属性がなく、修正できない script エレメント
- type 属性がなく、修正できない style エレメント
- alt 属性がなく、修正できない img エレメント
- alt 属性がなく、修正できない area エレメント

dom.convertToXHTML()

対応バージョン

Dreamweaver MX

説明

HTML の解析を実行して DOM ツリーを作成し、XHTML に必要で欠落しているアイテムを挿入し、ツリーをクリーンアップし、最後に、そのツリーをクリーンな XHTML として記述します。convertToXHTML() 関数によって、必要に応じて DOM ツリーに追加される、HTML に存在しないディレクティブ、宣言、エレメント、および属性には、以下のアイテムが含まれます。

- XML ディレクティブ
- doctype 宣言
- html エレメント内の xmlns 属性
- head セクション
- title エレメント
- body セクション

dom.convertToXHTML() 関数は、変換時に、純粋な HTML タグと属性を小文字に変換し、HTML タグと属性を正しい XHTML シンタックスで記述し、可能であれば、欠落している HTML 属性を追加します。この関数は、[環境設定] ダイアログボックスの設定に従って、サードパーティタグと属性を処理します。

ドキュメントがテンプレートの場合、dom.convertToXHTML() 関数はユーザーに警告を発し、変換を実行しません。

引数

なし

戻り値

以下のアイテムの個数を表す 6 つの整数から成る配列。

- Dreamweaver によって修正された XHTML エラー
- id 属性がなく、修正できない map エレメント
- type 属性がなく、修正できない script エレメント
- type 属性がなく、修正できない style エレメント
- alt 属性がなく、修正できない img エレメント
- alt 属性がなく、修正できない area エレメント

例

通常の使用では、最初に拡張機能によって dreamweaver.openDocument() 関数または dreamweaver.getDocumentDOM() 関数が呼び出され、ドキュメントへの参照が取得されます。次に、この拡張機能によって dom.getIsXHTMLDocument() 関数が呼び出され、ドキュメントが既に XHTML 形式になっているかどうかが判別されます。XHTML 形式でなければ、dom.convertToXHTML() 関数が呼び出され、ドキュメントが XHTML に変換されます。次に、dreamweaver.saveDocument() 関数が呼び出され、変換されたファイルが新しいファイル名で保存されます。

dom.getIsXHTMLDocument()

対応バージョン

Dreamweaver MX

説明

ドキュメント (具体的には <!DOCTYPE> 宣言) をチェックして、XHTML かどうかを確認します。

引数

なし

戻り値

ドキュメントが XHTML であれば true、それ以外の場合は false。

dreamweaver.browseForFileURL()

対応バージョン

Dreamweaver 1、2、3、4 (2、3、および 4 で機能強化)

説明

指定したラベルをタイトルバーに付けて、指定したタイプのダイアログボックスを開きます。

引数

openSelectOrSave、{titleBarLabel}、{bShowPreviewPane}、{bSupressSiteRootWarnings}、{arrayOfExtensions}

- *openSelectOrSave* 引数には、ダイアログボックスのタイプを示すストリングを "open"、"select"、または "save" のように指定します。
- *titleBarLabel* 引数は、Dreamweaver 2 から追加されました。これには、ダイアログボックスのタイトルバーに表示するラベルを指定します。この引数を指定しない場合は、オペレーティングシステム側で用意されたデフォルトのラベルが使用されます。

- *bShowPreviewPane* 引数は、Dreamweaver 2 から追加されました。これには、ダイアログボックスにイメージプレビューペインを表示するかどうかを示すブール値を指定します。この引数の値が `true` の場合は、ダイアログボックスでイメージファイルのフィルタリングが行われます。省略した場合のデフォルト値は `false` です。
- *bSupressSiteRootWarnings* 引数は、Dreamweaver 3 から追加されました。これには、選択したファイルがサイトルートの外にあるという警告を非表示にするかどうかを示すブール値を指定します。この引数を指定しない場合のデフォルト値は `false` です。
- *arrayOfExtensions* 引数は、Dreamweaver 4 から追加されました。これには、ダイアログボックスの下部にある [ファイルの種類] リストメニューのコンテンツのデフォルト値を、ストリングの配列で指定します。正しいシンタックスは、`menuEntryText|.xxx[;.yyy;.zzz]|CCCC|` です。ここで、*menuEntryText* は、表示するファイルタイプの名前を表します。拡張子は、*.xxx[;.yyy;.zzz]* または *CCCC* を指定できます。*.xxx* には、ファイルタイプのファイル拡張子を指定し (オプションで *.yyy* および *.zzz* を使用して複数のファイル拡張子を指定し)、CCCC には、Macintosh で使用するための 4 文字のファイルタイプ定数を指定します。

戻り値

ファイル名を file:// URL 形式で表記したストリング。

dreamweaver.browseForFolderURL()

対応バージョン

Dreamweaver 3

説明

指定されたラベルをタイトルバーに付けて [フォルダの選択] ダイアログボックスを開きます。

引数

{titleBarLabel}、*{directoryToStartIn}*

- *titleBarLabel* 引数には、ダイアログボックスのタイトルバーに表示するラベルを指定します。*titleBarLabel* 引数を省略したときのデフォルト値は、[フォルダの選択] です。
- *directoryToStartIn* 引数には、選択の開始点となるフォルダへのパスを file:// URL 形式で指定します。

戻り値

フォルダ名を file:// URL 形式で表記したストリング。

例

次のコードは、フォルダの URL を返します。

```
return dreamweaver.browseForFolderURL('Select a Folder',¬
dreamweaver.getSiteRoot());
```

dreamweaver.closeDocument()

対応バージョン

Dreamweaver 2

説明

指定したドキュメントを閉じます。

引数

documentObject

- *documentObject* 引数には、ドキュメントの DOM ツリーのルートにあるオブジェクト (`dreamweaver.getDocumentDOM()` 関数で返される値) を指定します。*documentObject* 引数がアクティブなドキュメントを参照する場合、この関数を呼び出すスクリプトの実行が完了するまでドキュメントウィンドウが閉じないことがあります。

戻り値

なし

dreamweaver.createDocument()

対応バージョン

Dreamweaver 2、Dreamweaver 4 (Dreamweaver 4 で機能強化)

説明

この関数に渡す引数に応じて、新規ドキュメントが同じウィンドウまたは新規ウィンドウに表示されます。新規ドキュメントがアクティブなドキュメントになります。

注意 : この関数は、"*menus.xml*" ファイル、コマンド、またはプロパティインスペクタファイルからのみ呼び出すことができます。ビヘイビアアクションやオブジェクトからこの関数を呼び出そうとすると、*Dreamweaver* はエラーメッセージを表示します。

引数

{bOpenInSameWindow}、*{type}*

- *bOpenInSameWindow* 引数には、新規ドキュメントを現在のウィンドウで開くかどうかを示すブール値を指定します。*bOpenInSameWindow* 引数を `false` 値に指定するか、この引数を指定しない場合、新規ドキュメントは別のウィンドウで開きます。この引数は、Windows のみで有効です。
- *type* 引数には、作成するドキュメントのタイプを指定します。ドキュメントタイプは、Dreamweaver の "Configuration¥DocumentTypes¥MMDocumentTypes.xml" ファイルで、`documenttype` タグの `id` 属性として定義されます。*type* 引数には、たとえば `"HTML"`、`"ASP-JS"`、`"ASP-VB"`、`"ColdFusion"`、`"CFC"`、`"JSP"`、`"ASP.NET_VB"` を指定します。指定できるすべてのタイプのリストについては、"MMDocumentTypes.xml" ファイルを参照してください。*type* を指定しない場合のデフォルト値は、`"HTML"` です。

注意 : "*MMDocumentTypes*" ファイルは、独自のドキュメントタイプを追加して拡張することができます。ドキュメントタイプ拡張の詳細については、『*Dreamweaver* 拡張ガイド』を参照してください。

戻り値

新規作成されたドキュメントのドキュメントオブジェクト。これは `dreamweaver.getDocumentDOM()` 関数の戻り値と同じです。

dreamweaver.createXHTMLDocument()

対応バージョン

Dreamweaver MX

説明

この関数に渡す引数に応じて、新規 XHTML ドキュメントが同じウィンドウまたは新規ウィンドウに表示されます。新規ドキュメントがアクティブなドキュメントになります。これは、`dreamweaver.createDocument()` 関数に似ています。

Dreamweaver は、新規 XHTML ドキュメントの作成時に、"Configuration/Templates" フォルダにある "default.xhtml" というファイルを読み取ります。次に、このファイルの内容を使用して、以下のスケルトン宣言が含まれる出力ファイルを作成します。

```
<?xml version="1.0">
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=" />
</head>

<body bgcolor="#FFFFFF" text="#000000">

</body>
</html>
```

デフォルトのドキュメントタイプ定義 (DTD) 宣言は、`Strict` ではなく、`XHTML 1.0 Transitional` です。ユーザーがフレームセットをドキュメントに追加すると、DTD は `XHTML 1.0 Frameset` に切り替えられます。`Content-Type` は `text/html` です。`charset` は、default.xhtml ファイルでは意図的に省略されていますが、ユーザーが新規ドキュメントを表示する前に値が設定されます。`?xml` ディレクティブは、ドキュメントで使用する文字エンコードが UTF-8 または UTF-16 の場合は必要ありませんが、このディレクティブがあれば、古いバージョンのブラウザで表示できる場合があります。ただし、このディレクティブは XHTML ドキュメント内に置く必要があるため、新規ドキュメントと変換されたドキュメントのどちらの場合も、デフォルトでこのディレクティブが使用されます。ユーザーは、手作業でこのディレクティブを削除できます。`?xml` ディレクティブには、エンコード属性があります。これは、`Content-Type` 属性の中の `charset` に一致します。

引数

{bOpenInSameWindow}

- *bOpenInSameWindow* 引数には、新規ドキュメントを現在のウィンドウで開くかどうかを示すブール値を指定します。この値が `false` であるか、または省略されている場合、新規ドキュメントは別のウィンドウで開きます。この引数は、Windows のみで有効です。

戻り値

新規作成されたドキュメントのドキュメントオブジェクト。これは、`dreamweaver.getDocumentDOM()` 関数が返す値と同じです。

dreamweaver.createXMLDocument()

対応バージョン

Dreamweaver MX

説明

新規 XML ファイルを作成して開きます。このファイルには、XML ディレクティブのみがあります。

引数

なし

戻り値

新規 XML ファイルの DOM。

例

次の例では、XML ディレクティブのみが入っている新規ドキュメントを作成します。

```
var theDOM = dreamweaver.createXMLDocument("document");
```

dreamweaver.exportCSS() (非推奨)

対応バージョン
Dreamweaver 3

説明
[CSS ファイルとして書き出し] ダイアログボックスを開きます。

引数
なし

戻り値
なし

イネーブラ
詳細については、418 ページの dreamweaver.canExportCSS() (非推奨) を参照してください。

dreamweaver.exportEditableRegionsAsXML() (非推奨)

対応バージョン
Dreamweaver 3 (MX では非推奨)

説明
この関数は、[編集可能領域を XML として書き出し] ダイアログボックスを開きます。

引数
なし

戻り値
なし

dreamweaver.exportTemplateDataAsXML()

対応バージョン
Dreamweaver MX

説明
現在のドキュメントを指定ファイルに XML として書き出します。この関数は、フォーカスが置かれているドキュメントで実行します。ドキュメントはテンプレートである必要があります。ファイル名の引数を省略した場合は、書き出し先ファイルのストリングを入力するためのダイアログボックスが Dreamweaver MX によって表示されます。

引数
{filePath}

- *filePath* 引数 (オプション) には、Dreamweaver のテンプレート書き出し先のファイル名を表すストリングを指定します。*filePath* 引数には、URL ファイルストリングとして指定します。たとえば、`"file:///c|/temp/mydata.txt"` とします。

戻り値
なし

イネーブラ

詳細については、418 ページの dreamweaver.canExportTemplateDataAsXML() を参照してください。

例

```
if(dreamweaver.canExportTemplateDataAsXML())
{
    dreamweaver.exportTemplateDataAsXML("file:///c|/dw_temps/mytemplate.txt")
}
```

dreamweaver.getDocumentDOM()

対応バージョン

Dreamweaver 2

説明

指定されたドキュメントに対するオブジェクトツリーへのアクセスを提供します。呼び出し元にオブジェクトツリーが返された後、呼び出し元は受け取ったツリーを編集してドキュメントのコンテンツを変更できます。

引数

{sourceDoc}

- *sourceDoc* 引数には、"document"、"parent"、"parent.frames[*number*]"、"parent.frames['*frameName*']"、または URL のいずれかを指定する必要があります。sourceDoc を省略した場合のデフォルト値は "document" です。引数に指定するこれらの値には、次の意味があります。
  - document 値は、フォーカスが置かれ、現在の選択部分が含まれているドキュメントを示します。
  - parent 値は、親フレームセットを示します (現在選択されているドキュメントがフレーム内にある場合)。
  - parent.frames[*number*] および parent.frames['*frameName*'] 値は、現在のドキュメントを含むフレームセット内の特定のフレームにあるドキュメントを示します。
  - この引数が相対 URL の場合、この URL の基準は拡張機能ファイルです。

注意 : この引数が "*document*" の場合、呼び出し元の関数は、*applyBehavior()* 関数、*deleteBehavior()* 関数、*objectTag()* 関数、あるいはドキュメントを編集できるコマンドまたはプロパティインスペクタファイル内の関数であることが必要です。

戻り値

ツリーのルートにある JavaScript ドキュメントオブジェクト。

例

次の例では、dreamweaver.getDocumentDOM() 関数を使用して現在のドキュメントにアクセスします。

```
var theDOM = dreamweaver.getDocumentDOM("document");
```

次の例では、現在のドキュメント DOM により選択部分を識別し、それを他のドキュメントの末尾にペーストします。

```
var currentDOM = dreamweaver.getDocumentDOM('document');
currentDOM.setSelection(100,200);
currentDOM.clipCopy();
var otherDOM = dreamweaver.openDocument(dreamweaver.¬
getSiteRoot() + "html/foo.htm");
otherDOM.endOfDocument();
otherDOM.clipPaste();
```

注意 : *DOM* メソッドは通常、開いているドキュメントに対してのみ動作するため、*openDocument()* 引数を使用します。開いていないドキュメントに対して関数を実行すると、*Dreamweaver* のエラーが発生します。*DOM* メソッドが、アクティブなドキュメントまたは閉じているドキュメントに対してのみ動作する場合は、そのことがメソッドの説明に記載されています。

dreamweaver.getNewDocumentDOM()

対応バージョン

Dreamweaver MX (Dreamweaver 8 では `documentType` 引数を追加)

説明

新しい空のドキュメントに対する編集可能ツリーへのアクセスを提供します。この関数は、`getDocumetDOM()` 関数と同じように機能します。ただし、既存のドキュメントではなく新規ドキュメントをポイントし、ドキュメントを開きません。

引数

`{documentType}`

- *`documentType`* 引数はストリングです。その値は、"DocumentTypes.xml" ファイルで指定されるドキュメントタイプである必要があります。

戻り値

新しい空のドキュメントへのポインタ。

例

次のコードは、新しい空のドキュメントの DOM を返します。

```
var theDOM = dreamweaver.getNewDocumentDOM();
```

dreamweaver.getRecentFileList()

対応バージョン

Dreamweaver 3

説明

[ファイル] メニューの一番下にある [最近使ったファイル] リストに含まれるすべてのファイルのリストを取得します。

引数

なし

戻り値

最近アクセスしたファイルへのパスを表すストリングを含む配列。各パスは、file:// URL 形式で表されます。最近アクセスしたファイルがない場合、戻り値はありません。

dreamweaver.importXMLIntoTemplate()

対応バージョン

Dreamweaver 3

説明

XML テキストファイルを現在のテンプレートドキュメントに読み込みます。この関数は、フォーカスが置かれているドキュメントで実行します。ドキュメントはテンプレートである必要があります。ファイル名の引数を省略した場合は、読み込みファイルのストリングを入力するためのダイアログボックスが Dreamweaver によって表示されます。

引数

`{filePath}`

- *`filePath`* 引数 (オプション) には、Dreamweaver のテンプレート読み込み先のファイル名を表すストリングを指定します。*`filePath`* 引数には、URL ファイルストリングとして指定します。たとえば、`"file:///c|/temp/mydata.txt"` とします。

戻り値

なし

dreamweaver.newDocument()

対応バージョン

Dreamweaver MX

説明

現在のサイトでドキュメントを開き、[新規ドキュメント] ダイアログボックスを表示します。

引数

{bopenWithCurSiteAndShowDialog}

- *bopenWithCurSiteAndShowDialog* 引数 (オプション) の値は、`true` または `false` です。現在のサイトでドキュメントを開き、[新規ドキュメント] ダイアログボックスを表示するには `true`、それ以外の場合は `false` を指定します。

戻り値

なし

dreamweaver.newFromTemplate()

対応バージョン

Dreamweaver 3

説明

指定したテンプレートに基づいて、新規ドキュメントを作成します。引数を指定しない場合、[テンプレートの選択] ダイアログボックスが表示されます。

引数

{templateURL}、*bMaintain*

- *templateURL* 引数には、現在のサイトのテンプレートへのパスを file:// URL 形式で指定します。
- *bMaintain* 引数には、元のテンプレートへのリンクを維持するかどうかを示す `true` または `false` のブール値を指定します。

戻り値

なし

dreamweaver.openDocument()

対応バージョン

Dreamweaver 2

説明

新規 Dreamweaver ウィンドウで編集するためにドキュメントを開き、フォーカスを与えます。ユーザーにとっては、[ファイル]-[開く] を選択してファイルを選択する場合と同じ結果になります。指定したファイルが既に開いている場合、そのドキュメントを含むウィンドウが前面に表示されます。指定したファイルを含むウィンドウが現在選択されているドキュメントになります。Dreamweaver 2 では、チェックイン / チェックアウトが使用可能になっている場合、まずファイルをチェックアウトしてから開きます。Dreamweaver 3 以降のバージョンでこのビヘイビアを取得するには、222 ページの dreamweaver.openDocumentFromSite() を使用する必要があります。

注意 : [ビヘイビア] アクションまたはオブジェクトファイルからこの関数を呼び出すと、エラーが発生します。

引数

fileName

- *fileName* 引数には、開くファイルの名前を URL で指定します。相対 URL を指定する場合は、この関数を呼び出したスクリプトが含まれるファイルを基準とする URL を指定します。

戻り値

指定されたファイルのドキュメントオブジェクト。これは、dreamweaver.getDocumentDOM() 関数が返す値と同じです。

dreamweaver.openDocumentFromSite()

対応バージョン

Dreamweaver 3

説明

新規 Dreamweaver ウィンドウで編集するためにドキュメントを開き、フォーカスを与えます。ユーザーにとっては、[サイト] パネルでファイルをダブルクリックした場合と同じ結果になります。指定したファイルが既に開いている場合、そのドキュメントを含むウィンドウが前面に表示されます。指定したファイルを含むウィンドウが現在選択されているドキュメントになります。

注意 : [ビヘイビア] アクションまたはオブジェクトファイルからこの関数を呼び出すことはできません。呼び出すと、エラーが発生します。

引数

fileName

- *fileName* 引数には、開くファイルを URL で指定します。相対 URL を指定する場合は、この関数を呼び出したスクリプトが含まれるファイルを基準とする URL を指定します。

戻り値

指定されたファイルのドキュメントオブジェクト。これは、dreamweaver.getDocumentDOM() 関数が返す値と同じです。

dreamweaver.openInFrame()

対応バージョン

Dreamweaver 3

説明

[フレーム内に開く] ダイアログボックスを開きます。ユーザーがドキュメントを選択すると、そのドキュメントがアクティブフレーム内に開きます。

引数

なし

戻り値

なし

イネーブラ

詳細については、419 ページの dreamweaver.canOpenInFrame() を参照してください。

dreamweaver.releaseDocument()

対応バージョン
Dreamweaver 2

説明
以前に参照したドキュメントを明示的にメモリから解放します。

`dreamweaver.getObjectTags()`、`dreamweaver.getObjectRefs()`、`dreamweaver.getDocumentPath()`、または `dreamweaver.getDocumentDOM()` 関数によって参照したドキュメントは、その呼び出しを含むスクリプトの実行が完了すると自動的に解放されます。スクリプトが開くドキュメントの数が多い場合には、スクリプトが完了する前にこの関数を使用してドキュメントを明示的に解放し、メモリ不足を防ぎます。

注意 : この関数が意味を持つのは、*URL* によって参照された、現在フレームやドキュメントウィンドウ内で開いていない、および、拡張機能ファイルではないドキュメントに対してのみです。拡張機能ファイルは起動時にメモリにロードされ、*Dreamweaver* を終了するまで解放されません。

引数
`documentObject`

- *`documentObject`* 引数には、ドキュメントの DOM ツリーのルートにあるオブジェクト (`dreamweaver.getDocumentDOM()` 関数が返す値) を指定します。

戻り値
なし

dreamweaver.revertDocument()

対応バージョン
Dreamweaver 3

説明
指定したドキュメントを前回保存したバージョンに戻します。

引数
`documentObject`

- *`documentObject`* 引数には、ドキュメントの DOM ツリーのルートにあるオブジェクト (`dreamweaver.getDocumentDOM()` 関数が返す値) を指定します。

戻り値
なし

イネーブラ
詳細については、421 ページの dreamweaver.canRevertDocument() を参照してください。

dreamweaver.saveAll()

対応バージョン
Dreamweaver 3

説明
開いているすべてのドキュメントを保存します。今までに保存されたことがないドキュメントの場合は、[新規保存] ダイアログボックスを開きます。

引数
なし

戻り値
なし

イネーブラ
詳細については、421 ページの dreamweaver.canSaveAll() を参照してください。

dreamweaver.saveDocument()

対応バージョン
Dreamweaver 2

説明
指定したファイルをローカルコンピュータ上に保存します。

注意 : *Dreamweaver 2* では、ファイルが読み込み専用の場合は *Dreamweaver* がそのチェックアウトを試行します。チェックアウトを試行した後でもドキュメントが読み込み専用である場合や、ドキュメントを作成できない場合には、エラーメッセージが表示されます。

引数
documentObject、*{fileURL}*

- *documentObject* 引数には、ドキュメントの DOM ツリーのルートにあるオブジェクト (dreamweaver.getDocumentDOM() 関数が返す値) を指定します。
- *fileURL* 引数 (オプション) には、ローカルコンピュータ上の保存場所を表す URL を指定します。相対 URL を指定する場合は、拡張機能ファイルを基準とする URL を使用します。Dreamweaver 2 では、この引数の指定は必須です。Dreamweaver 4 では、*fileURL* 引数を省略した場合、保存済みのファイルは現在の場所に保存され、新規ファイルの場合は [保存] ダイアログボックスが表示されます。

戻り値
成功 (true) または失敗 (false) を示すブール値。

イネーブラ
詳細については、421 ページの dreamweaver.canSaveDocument() を参照してください。

dreamweaver.saveDocumentAs()

対応バージョン
Dreamweaver 3

説明
[新規保存] ダイアログボックスを開きます。

引数
documentObject

- *documentObject* 引数には、ドキュメントの DOM ツリーのルートにあるオブジェクト (dreamweaver.getDocumentDOM() 関数が返す値) を指定します。

戻り値
なし

dreamweaver.saveDocumentAsTemplate()

対応バージョン
Dreamweaver 3

説明
[テンプレートとして保存] ダイアログボックスを開きます。

引数
documentObject、*{fileName}*

- *documentObject* 引数には、ドキュメントの DOM ツリーのルートにあるオブジェクト (dreamweaver.getDocumentDOM() が返す値) を指定します。
- *fileName* 引数には、開くファイルの名前を絶対 URL で指定します。

戻り値
なし

イネーブラ
詳細については、422 ページの dreamweaver.canSaveDocumentAsTemplate() を参照してください。

dreamweaver.saveFrameset()

対応バージョン
Dreamweaver 3

説明
指定したフレームセットを保存します。今までに保存されたことがないフレームセットの場合は、[新規保存] ダイアログボックスを開きます。

引数
documentObject

- *documentObject* 引数には、ドキュメントの DOM ツリーのルートにあるオブジェクト (dreamweaver.getDocumentDOM() 関数が返す値) を指定します。

戻り値
なし

イネーブラ
詳細については、422 ページの dreamweaver.canSaveFrameset() を参照してください。

dreamweaver.saveFramesetAs()

対応バージョン
Dreamweaver 3

説明
指定した DOM を含むフレームセットファイルを保存するための [新規保存] ダイアログボックスを開きます。

引数

documentObject

- *documentObject* 引数には、ドキュメントの DOM ツリーのルートにあるオブジェクト (dreamweaver.getDocumentDOM() 関数が返す値) を指定します。

戻り値

なし

イネーブラ

詳細については、422 ページの dreamweaver.canSaveFramesetAs() を参照してください。

グローバルドキュメント関数

グローバルドキュメント関数は、ドキュメント全体に対して動作します。これらの関数は、スペルチェック、ターゲットブラウザチェック、ページプロパティの設定、ドキュメント内のエレメントに対する正しいオブジェクト参照の判別などを行います。

dom.checkSpelling()

対応バージョン

Dreamweaver 3

説明

必要に応じて [スペルチェック] ダイアログボックスを開いてドキュメントのスペルチェックを行い、スペルチェックが完了したらユーザーに通知します。

引数

なし

戻り値

なし

dom.checkTargetBrowsers()

対応バージョン

Dreamweaver 3

説明

ドキュメントに対してターゲットブラウザチェックを実行します。フォルダやファイルグループに対してターゲットブラウザチェックを実行するには、185 ページの site.checkTargetBrowsers() を参照してください。

引数

なし

戻り値

なし

dom.getParseMode()

対応バージョン
Dreamweaver MX 2004

説明
ドキュメントの現在の解析モードを取得します。解析モードは、ドキュメントの検証方法と、ドキュメントがメインのドキュメントウィンドウに HTML として表示されるかどうかを検証する方法を制御します。

引数
なし

戻り値
現在の解析モードを示すストリング。`"html"`、`"xml"`、`"css"`、または `"text"` が返されます。

dom.hideInfoMessagePopup()

対応バージョン
Dreamweaver MX 2004

説明
ドキュメントウィンドウにツールヒントのようなメッセージが表示されている場合に、それを非表示にします。

引数
なし

戻り値
なし

参照項目
228 ページの dom.showInfoMessagePopup().

dom.runValidation()

対応バージョン
Dreamweaver MX、Dreamweaver MX 2004 (Dreamweaver MX 2004 ではオプション引数追加)

説明
指定された 1 つのドキュメントに対してバリデータを実行します。この関数は 202 ページの site.runValidation() に似ています。バリデータは、ドキュメントの doctype (HTML 4.0 や HTML 3.2 など) に指定される言語、およびサーバーモデル (ColdFusion や ASP など) によって指定される言語と、ドキュメントが一致しているかどうかをチェックします。ドキュメントに doctype がない場合、バリデータは [環境設定] ダイアログボックスの [バリデータ] セクションに指定された言語設定を使用します。

引数
{controlString}、*{bOpenResultsWindow}*、*{bShowInfoMessage}*

- *controlString* 引数は、空白のストリング、`"xml"`、`"auto-explicit"`、または `"auto-implicit"` の 4 つの値のいずれかを指定できるオプションのストリングです。
  - 引数が空白のストリングである場合、バリデータはデフォルトを使用して検証を実行します。引数が `"xml"` の場合、バリデータはドキュメントを XML として検証します。

- 引数が "auto-explicit" または "auto-implicit" の場合、Dreamweaver は自動検証 (インライン検証) を実行します。自動検証では、バリデータの [結果] ウィンドウを開くのではなく、コードビューでエラーにアンダーラインを表示します。詳細については、394 ページの dom.source.getValidationErrorsForOffset() および 389 ページの dom.getAutoValidationCount() を参照してください。
- *controlString* 引数が "auto-explicit" の場合、Dreamweaver は検証を実行する前に、保存していないドキュメントを保存するようにユーザーに指示します。
- *controlString* 引数が "auto-implicit" の場合、検証は失敗します。このとき、現在のドキュメントが保存されていないことをユーザーに通知しません。

注意 : 自動検証 (*controlString* の *"auto-explicit"* 値または *"auto-implicit"* 値で定義) は、現在、ブラウザ互換性チェックでのみ有効です。

- *bOpenResultsWindow* 引数は、オプションのブール値です。バリデータの [結果] ウィンドウを開くには true、それ以外の場合は false を指定します。デフォルト値は true です。
- *bShowInfoMessage* 引数は、*controlString* 引数に "auto-explicit" または "auto-implicit" を定義したときにのみ使用します。*bShowInfoMessage* 引数はブール値です。true を指定すると、見つかったエラーの数を示す情報メッセージがツールバー項目 DW_ValidatorErrors の下に示され、false を指定すると何も表示されません。デフォルト値は false です。

戻り値

バリデータの [結果] ウィンドウオブジェクト。

例

次の例では、ユーザーが [ファイル]-[ページのチェック]-[マークアップのバリデート] (または [バリデータ] パネルの [現在のドキュメントをバリデート]) を選択したときに、通常の検証を実行します。

```
dw.getDocumentDOM().runValidation('');
```

次の例では、保存していないドキュメントを保存するようにユーザーに指示し、自動検証を実行します。さらに、バリデータの [結果] ウィンドウを開かずに、エラーの総数をドキュメントツールバーの DW_ValidatorErrors のボタンに表示します。

```
dw.getDocumentDOM().runValidation('auto-explicit', false, true);
```

次の例では、保存していないドキュメントの保存をユーザーに指示しません。ドキュメントが保存されていない場合、バリデータは開始しません。ドキュメントが保存されている場合、Dreamweaver は自動検証を実行します。ただし、バリデータの [結果] ウィンドウは開かれず、見つかったエラーの総数がドキュメントツールバーに表示されることはありません。

```
dw.getDocumentDOM().runValidation('auto-implicit', false);
```

dom.showInfoMessagePopup()

対応バージョン

Dreamweaver MX 2004

説明

ドキュメントウィンドウ内、またはツールバー項目の下に、ツールヒントのようなメッセージを表示します。

引数

location、*message*、*timeout*

- *location* 引数はストリングです。ツールバー項目を示すストリング、空白のストリング、または "top"、"topright"、"right"、"bottomright"、"bottom"、"bottomleft"、"left"、"topleft" のいずれかのキーワードを指定します。ツールヒントは、指定した辺や角に合わせて配置され、中央に整列されます。空白のストリングを指定すると、ドキュメント内で中央に整列されます。ツールバー項目を指定するには、"toolbar:*toolbarID*:*itemID*" を使用します。ここで、ツールバー ID およびツールバー項目 ID は、"toolbars.xml" ファイル内の ID と同じです。

- *message* 引数には、メッセージを含むストリングを指定します。
- *timeout* 引数には、メッセージを表示する時間をミリ秒単位で指定します。デフォルト値は 0 です。値が 0 の場合、メッセージは消去されません。ユーザーがメッセージをクリックするか、またはドキュメントを切り替えた場合、もしくは指定時間が経過した場合は、Dreamweaver によってメッセージが自動的に消去されます。

戻り値

なし

例

次の例では、2 つのツールヒントメッセージを表示します。最初のコード行では、ドキュメントの中央に "This message is in the center" というメッセージが表示されます。showInfoMessagePopup() の 2 番目の呼び出しにより、DW_Toolbar_Main という ID のツールバーにある DW_SetTitle という ID の [タイトル] テキストボックスに対して、"Don't forget the title for the Window" というツールヒントメッセージが表示されます。

```
dw.getDocumentDOM.showInfoMessagePopup('', 'This message is in the center', 5000);
dw.getDocumentDOM.showInfoMessagePopup('toolbar:DW_Toolbar_Main:DW_SetTitle', 'Don't
    forget the title for the window', 5000);
```

参照項目

227 ページの dom.hideInfoMessagePopup().

dom.showPagePropertiesDialog()

対応バージョン

Dreamweaver 3

説明

[ページプロパティ] ダイアログボックスを開きます。

引数

なし

戻り値

なし

dreamweaver.doURLDecoding()

対応バージョン

Dreamweaver MX

説明

Dreamweaver 内部の URL デコードメカニズムを使用して、URL ストリングに含まれる特殊文字および記号をデコードします。たとえば、%20 をスペースに、&quot を " にデコードします。

引数

inStr

- inStr 引数には、デコードするストリングを指定します。

戻り値

デコード後の URL が格納されたストリング。

例

次の例では、dw.doURLDecoding() を呼び出して引数内の特殊文字をデコードし、結果のストリングを outstr に保存します。

```
outStr = dreamweaver.doURLDecoding("http://maps.yahoo.com/py/ddResults.py?Pyt= ¬
Tmap&tarname=&tardesc=&newname=&newdesc=&newHash=&newTHash=&newSts=&newTSts=&tlt=&tln= ¬
&slt=&sln=&newFL=Use+Address+Below&newaddr=2000+Shamrock+Rd&newcsz=Metroo+Park%2C+CA& ¬
newcountry=us&newTFL=Use+Address+Below&newtaddr=500+El+Camino&newtcsz=Santa+Clara%2C+CA& ¬
newtcountry=us&Submit=Get+Directions")
```

dreamweaver.getElementRef()

対応バージョン

Dreamweaver 2

説明

DOM ツリー内の特定のタグオブジェクトに対する Netscape Navigator または Internet Explorer のオブジェクト参照を取得します。

引数

NSorIE、*tagObject*

- *NSorIE* 引数には、"NS 4.0" または "IE 4.0" のいずれかを指定する必要があります。Netscape Navigator 4.0 と Internet Explorer 4.0 では、DOM およびネストされた参照の規則が異なります。この引数には、どちらのブラウザに対して有効な参照を返すかを指定します。
- *tagObject* 引数には、DOM ツリー内のタグオブジェクトを指定します。

戻り値

このオブジェクトに対する有効な JavaScript 参照を表すストリング。たとえば、document.layers['myLayer'] を返します。ストリングには、次の条件があります。

- Dreamweaver は、Internet Explorer 用には、A、AREA、APPLET、EMBED、DIV、SPAN、INPUT、SELECT、OPTION、TEXTAREA、OBJECT、および IMG の各タグへの正しい参照を返します。
- Dreamweaver は Netscape Navigator 用には、A、AREA、APPLET、EMBED、LAYER、ILAYER、SELECT、OPTION、TEXTAREA、OBJECT、と IMG タグ、および絶対位置指定された DIV と SPAN タグへの正しい参照を返します。絶対位置指定されていない DIV タグと SPAN タグについては、Dreamweaver は "cannot reference <*tag*>" というストリングを返します。
- Dreamweaver は、名前のないオブジェクトについては参照を返しません。オブジェクトに NAME 属性も ID 属性も含まれていない場合は、"unnamed <*tag*>" というストリングが返されます。ブラウザが名前による参照方法をサポートしない場合、Dreamweaver はインデックスを使用してオブジェクトを参照します (たとえば、document.myform.applets[3])。
- 名前が付けられたオブジェクトが名前のないフォームやレイヤーに含まれていても、そのオブジェクトに対する参照は返されます (たとえば、document.forms[2].myCheckbox)。

dreamweaver.getObjectRefs() (非推奨)

対応バージョン

Dreamweaver 1 (3 では非推奨)

説明

この関数は、指定したドキュメントで指定したタグのインスタンス、またはタグが指定されていない場合は、ドキュメント内のすべてのタグを検索し、それらのタグへの参照をブラウザ固有の形式で作成します。この関数は、getElementsByTagName() を呼び出した後、nodelist 中の各タグに対して dreamweaver.getElementRef() を呼び出すのと同じ操作を行います。

引数

NSorIE、*sourceDoc*、*{tag1}*、*{tag2}*、*...{tagN}*

- *NSorIE* 引数には、"NS 4.0" または "IE 4.0" のいずれかを指定する必要があります。Netscape Navigator 4.0 と Internet Explorer 4.0 では、DOM およびネストされた参照の規則が異なります。この引数には、どちらのブラウザに対して有効な参照を返すかを指定します。
- *sourceDoc* 引数には、"document"、"parent"、"parent.frames[*number*]"、"parent.frames['*frameName*']"、または URL のいずれかを指定する必要があります。document 値は、フォーカスが置かれ、現在の選択部分が含まれているドキュメントを示します。parent 値は、現在選択されているドキュメントがフレーム内にある場合は親フレームセットを示します。parent.frames[*number*] および parent.frames['*frameName*'] は、現在のドキュメントを含むフレームセットに入っている、特定フレーム内のドキュメントを示します。この引数が相対 URL の場合、この URL の基準は拡張機能ファイルです。
- 3 番目以降の引数を指定する場合、これらにはタグの名前 (たとえば "IMG"、"FORM"、"HR" など) を指定します。

戻り値

ストリングの配列。各配列は、指定したブラウザに適した形式の有効な JavaScript 参照 (たとえば "document.myLayer.document.myImage") で、指定したドキュメント内に存在する、要求されたタグタイプの名前付きインスタンスを参照します。

- Dreamweaver は Internet Explorer 用には、A、AREA、APPLET、EMBED、DIV、SPAN、INPUT、SELECT、OPTION、TEXTAREA、OBJECT、および IMG の各タグへの正しい参照を返します。
- Dreamweaver は Netscape Navigator 用には、A、AREA、APPLET、EMBED、LAYER、ILAYER、SELECT、OPTION、TEXTAREA、OBJECT、と IMG タグ、および絶対位置指定された DIV と SPAN タグへの正しい参照を返します。絶対位置指定されていない DIV タグと SPAN タグについては、Dreamweaver は "cannot reference <*tag*>" というストリングを返します。
- Dreamweaver は、名前のないオブジェクトについては参照を返しません。オブジェクトに NAME 属性も ID 属性も含まれていない場合は、"unnamed <*tag*>" というストリングが返されます。ブラウザが名前による参照方法をサポートしない場合、Dreamweaver はインデックスを使用してオブジェクトを参照します (たとえば、document.myform.applets[3])。
- Dreamweaver では、名前のないフォームやレイヤーに含まれていても、それ自体が名前付きのオブジェクトに対する参照は返されます。たとえば、document.forms[2].myCheckbox となります。

同じ引数を含むリストを getObjectTags() に渡すと、これら 2 つの関数は、長さが同じでコンテンツが相互に対応する配列を返します。

dreamweaver.getObjectTags() (非推奨)

対応バージョン

Dreamweaver 1 (3 では非推奨)

説明

この関数は、指定したドキュメントで、指定したタグのインスタンス、またはタグが指定されていない場合は、ドキュメント内のすべてのタグを検索します。この関数は、getElementsByTagName() を呼び出した後で nodelist 内の各エレメントに対して outerHTML を呼び出すのと同じ操作を行います。

引数

sourceDoc、*{tag1}*、*{tag2}*、*...{tagN}*

- *sourceDoc* 引数には、"document"、"parent"、"parent.frames[*number*]"、"parent.frames['*frameName*']"、または URL のいずれかを指定する必要があります。document 値は、フォーカスが置かれ、現在の選択部分が含まれているドキュメントを示します。parent 値は、現在選択されているドキュメントがフレーム内にある場合は親フレームセットを示します。parent.frames[*number*] および parent.frames['*frameName*'] は、現在のドキュメントを含むフレームセットに入っている、特定フレーム内のドキュメントを示します。この引数が相対 URL の場合、この URL の基準は拡張機能ファイルです。
- 2 番目以降の引数を指定する場合、これらには、タグの名前 (たとえば "IMG"、"FORM"、"HR" など) を指定します。

戻り値

ストリングの配列。各配列は、指定したドキュメント内の要求されたタグタイプのインスタンスのソースコードです。

- *tag* 引数のいずれかに LAYER を指定した場合、この関数はすべての LAYER タグと ILAYER タグ、およびすべての絶対位置指定された DIV タグと SPAN タグを返します。
- *tag* 引数のいずれかに INPUT を指定した場合、この関数はすべてのフォームエレメントを返します。特定のタイプのフォームエレメントを取得するには、INPUT/*TYPE* と指定します。*TYPE* は、button、text、radio、checkbox、password、textarea、select、hidden、reset、または submit のいずれかです。

同じ引数を含むリストを getObjectRefs() に渡すと、これら 2 つの関数は同じ長さの配列を返します。

例

dreamweaver.getObjectTags("document", "IMG") は、アクティブなドキュメントのコンテンツによって、次のような項目を含む配列を返します。

- "<IMG SRC="/images/dot.gif" WIDTH="10" HEIGHT="10" NAME="bullet">"
- "<IMG SRC="header.gif" WIDTH="400" HEIGHT="32" NAME="header">"
- "<IMG SRC="971208_nj.jpg" WIDTH="119" HEIGHT="119" NAME="headshot">"

dreamweaver.getPreferenceInt()

対応バージョン

Dreamweaver MX

説明

拡張機能に対する整数の環境設定値を取得します。

引数

section、*key*、*default_value*

- *section* 引数には、対象のエントリが含まれている環境設定セクションを表すストリングを指定します。
- *key* 引数には、設定する値のエントリを表すストリングを指定します。
- *default_value* 引数には、エントリが見つからない場合に Dreamweaver が返すデフォルト値を指定します。この値は、0 ～ 65,535 の符号なし整数、または -32,768 ～ 32,767 の符号付き整数である必要があります。

戻り値

指定されたセクションの指定されたエントリの整数値。エントリが見つからない場合はデフォルト値。指定されたエントリの値が整数ではない場合は、0 を返します。

例

次の例では、[環境設定] の [My Extension] セクションにある [Snap Distance] 設定の整数値を返します。[My Extension] セクションまたは [Snap Distance] エントリがない場合、関数は指定されたデフォルト値 0 を返します。

```
var snapDist; // エントリが見つからない場合はデフォルト値
snapDist = dreamweaver.getPreferenceInt("My Extension", "Snap Distance", 0);
```

dreamweaver.getPreferenceString()

対応バージョン

Dreamweaver MX

注意 : サイトの環境設定値にアクセスするには、バージョン *7.0.1* が必要です。サイト情報にアクセスする前に、*dw.appVersion* で正しいバージョンを確認してください。

説明

拡張機能に対するストリングの環境設定値を取得します。

引数

section、*key*、*default_value*

- *section* 引数には、対象のエントリが含まれている環境設定セクションを表すストリングを指定します。
- *key* 引数には、取得する値を示すストリングを指定します。
- *default_value* 引数には、エントリが見つからない場合に Dreamweaver が返すデフォルトのストリング値を指定します。

戻り値

要求された環境設定ストリング。ストリングが見つからない場合はデフォルト値。

例

次の例では、[環境設定] の [My Extension] セクションにある [Text Editor] 設定のストリング値を返します。[My Extension] セクションまたは [Text Editor] エントリがない場合、関数は変数 `txtEditor` に指定されたデフォルト値を返します。

```
var txtEditor = getExternalTextEditor(); // テキストエディタのデフォルト値を設定します。
txtEditor = dreamweaver.getPreferenceString("My Extension", "Text Editor", txtEditor);
```

dreamweaver.setPreferenceInt()

対応バージョン

Dreamweaver MX

説明

拡張機能に対する整数の環境設定値を設定します。この設定は、Dreamweaver が動作していないときは Dreamweaver の環境設定に保存されています。

引数

section、*key*、*new_value*

- *section* 引数には、オプションを設定する環境設定カテゴリを表すストリングを指定します。指定したカテゴリが存在しない場合は、Dreamweaver によって作成されます。
- *key* 引数には、関数で設定するカテゴリオプションを表すストリングを指定します。指定したオプションが存在しない場合は、Dreamweaver によって作成されます。
- *new_value* 引数には、カテゴリオプションの値を表す整数を指定します。

戻り値

成功した場合は `true`、失敗した場合は `false`。

例

次の例では、[環境設定] の [My Extension] カテゴリにある [Snap Distance] エントリに、`snapDist` 変数の値を設定します。

```
var snapDist = getSnapDistance();
if(snapDist > 0)
{
    dreamweaver.setPreferenceInt("My Extension", "Snap Distance", snapDist);
}
```

dreamweaver.setPreferenceString()

対応バージョン

Dreamweaver MX

注意 : サイトの環境設定値にアクセスするには、バージョン *7.0.1* が必要です。サイト情報にアクセスする前に、*`dw.appVersion`* で正しいバージョンを確認してください。

説明

拡張機能に対するストリングの環境設定値を書き込みます。この設定は、Dreamweaver が動作していないときは Dreamweaver の環境設定に保存されています。

引数

`section`、*`key`*、*`new_value`*

- *`section`* 引数には、オプションを設定する環境設定カテゴリを表すストリングを指定します。指定したカテゴリが存在しない場合は、Dreamweaver によって作成されます。
- *`key`* 引数には、関数で設定するカテゴリオプションを表すストリングを指定します。指定したカテゴリオプションが存在しない場合は、Dreamweaver によって作成されます。
- *`new_value`* 引数には、カテゴリオプションの値を表すストリングを指定します。

戻り値

成功した場合は `true`、失敗した場合は `false`。

例

```
var txtEditor = getExternalTextEditor();
dreamweaver.setPreferenceString("My Extension", "Text Editor", txtEditor);
```

dreamweaver.showTargetBrowsersDialog()

対応バージョン

Dreamweaver MX 2004

説明

[ターゲットブラウザ] ダイアログボックスを開きます。[ターゲットブラウザ] ダイアログボックスでは、現在のページとブラウザとの互換性に問題がないかどうかをターゲットブラウザチェック機能でチェックするときに使用するブラウザバージョンを指定できます。

引数

なし

戻り値

なし

パス関数

パス関数は、ユーザーのハードディスクにあるさまざまなファイルやフォルダのパスを取得し、操作します。これらの関数は、現在のドキュメントが保存されているサイトのルートへのパスの判別、および相対パスから絶対 URL への変換などを行います。

dreamweaver.getConfigurationPath()

対応バージョン

Dreamweaver 2

説明

Dreamweaver の "Configuration" フォルダへのパスを file:// URL 形式で取得します。

Dreamweaver がマルチユーザープラットフォームで "Configuration" フォルダにアクセスする方法の詳細については、『Dreamweaver 拡張ガイド』の「C レベル拡張機能」を参照してください。

引数

なし

戻り値

アプリケーションの設定ファイルへのパス。

例

次の関数を使用すると、Dreamweaver アプリケーションフォルダ内の "Configuration" フォルダに保存されている他の拡張機能ファイルを参照できます。

```
var sortCmd = dreamweaver.getConfigurationPath() + ¬
"/Commands/Sort Table.htm"
var sortDOM = dreamweaver.getDocumentDOM(sortCmd);
```

dreamweaver.getDocumentPath()

対応バージョン

Dreamweaver 1.2

説明

指定したドキュメントへのパスを file:// URL 形式で取得します。この関数は、`dreamweaver.getDocumentDOM()` を呼び出して、戻り値の `URL` プロパティを読み取った場合と同じ操作を実行します。

引数

sourceDoc

- *sourceDoc* 引数には、"document"、"parent"、"parent.frames[*number*]"、または "parent.frames['*frameName*']" のいずれかの値を指定する必要があります。"document" 値は、フォーカスが置かれ、現在の選択部分が含まれているドキュメントを示します。"parent" 値は、現在選択されているドキュメントがフレーム内にある場合は親フレームセットを示します。"parent.frames[*number*]" 値および "parent.frames['*frameName*']" 値は、現在のドキュメントを含むフレームセットに入っている、特定フレーム内のドキュメントを示します。

戻り値

ファイルが保存されている場合は、指定したドキュメントの URL を含むストリング。ファイルが保存されていない場合は、空白のストリング。

dreamweaver.getSiteRoot()

対応バージョン
Dreamweaver 1.2

説明
現在選択されているドキュメントの関連サイトのローカルルートフォルダ ([サイト定義] ダイアログボックスで指定) を、file:// URL 形式で取得します。

引数
なし

戻り値
サイトにファイルが保存されている場合は、サイトのローカルルートフォルダの URL を含むストリング。ファイルがサイトに関連付けられていない場合は、空白のストリング。

dreamweaver.getTempFolderPath()

対応バージョン
Dreamweaver MX

説明
一時的に使用するファイルを保存できる一時フォルダへのフルパスを取得します。この関数は、Dreamweaver の "Configuration" フォルダ内で "Temp" フォルダを検索します。システムが複数のユーザーをサポートしている場合は、ユーザーの "Configuration" フォルダで検索が実行されます。"Temp" フォルダがない場合は、関数で作成されます。一時的でない共有ファイルは、"Configuration¥Shared" フォルダに格納する必要があります。

引数
なし

戻り値
file:// URL 形式で表されたフォルダへのフルパス。

例
次のコード行は、指定したファイルへのフルパスを返します。Dreamweaver の他の関数 (`dreamweaver.getSiteRoot()` など) と異なり、`dw.getTempFolderPath()` 関数が返すパスの最後にはスラッシュ (/) がありません。

```
var myTempfile = dw.getTempFolderPath() + "/myTempFile.txt";
```

dreamweaver.relativeToAbsoluteURL()

対応バージョン
Dreamweaver 2

説明
相対 URL および参照ポイント (現在のドキュメントへのパスまたはサイトルート) が指定されている場合、この関数はその相対 URL を file:// 形式の絶対 URL に変換します。

引数

docPath、*siteRoot*、*relURL*

- *docPath* 引数には、ユーザーのコンピュータ上のドキュメント (現在のドキュメントなど) へのパスを file:// URL 形式で指定します。*relURL* がルートからの相対 URL である場合は、空白のストリングを指定します。
- *siteRoot* 引数には、サイトルートへのパスを file:// URL 形式で指定します。*relURL* がドキュメントからの相対 URL である場合は、空白のストリングを指定します。
- *relURL* 引数には、変換する URL を指定します。

戻り値

絶対 URL を示すストリング。戻り値は、以下のように生成されます。

- *relURL* が絶対 URL である場合は、変換を行いません。戻り値は *relURL* と同じです。
- *relURL* がドキュメントからの相対 URL の場合、戻り値は *docPath* と *relURL* を組み合わせたパスです。
- *relURL* がルートからの相対 URL の場合、戻り値は *siteRoot* と *relURL* を組み合わせたパスです。

選択関数

選択関数は、開いているドキュメント内で選択範囲の取得および設定を行います。[サイト] パネルにおける選択範囲の取得または設定に関する詳細については、178 ページのサイト関数を参照してください。

dom.getSelectedNode()

対応バージョン

Dreamweaver 3

説明

選択したノードを取得します。この関数は、`dom.getSelection()` 関数を呼び出して、その戻り値を `dom.offsetsToNode()` 関数に渡す場合と同じ操作を行います。

引数

なし

戻り値

指定した範囲のすべての文字を含むタグ、テキスト、またはコメントオブジェクト。

dom.getSelection()

対応バージョン

Dreamweaver 3

説明

ドキュメントのソースコードにおける文字オフセットとして選択範囲を取得します。

引数

{bAllowMultiple}

- *bAllowMultiple* 引数 (オプション) には、複数のテーブルセル、イメージマップホットスポット、またはレイヤーが選択されている場合に、この関数が複数のオフセットを返す必要があるかどうかを示すブール値を指定します。

この引数を指定しない場合のデフォルト値は `false` です。

戻り値

単純な選択範囲の場合は、2 つの整数を含む配列を返します。最初の整数は、選択範囲の開始点の文字オフセットです。2 番目の整数は、選択範囲の終了点の文字オフセットです。これら 2 つの整数が同じ値である場合は、現在の選択範囲が挿入ポイントであることを示します。

複雑な選択範囲の場合 (複数のテーブルセル、複数のレイヤー、複数のイメージマップホットスポットなど) は、$2n$ 個の整数を含む配列を返します。*n* は選択した項目の数を表します。各ペアの最初の整数は、選択範囲の開始点 (TD、DIV、SPAN、LAYER、ILAYER、または MAP の開始タグを含む) の文字オフセットです。各ペアの 2 番目の整数は、選択範囲の終了点 (TD、DIV、SPAN、LAYER、ILAYER、または MAP の終了タグを含む) の文字オフセットです。複数のテーブル行が選択されている場合は、各行にある各セルのオフセットを返します。選択範囲に TR タグが含まれることはありません。

dom.nodeToOffsets()

対応バージョン

Dreamweaver 3

説明

DOM ツリー内の特定のノードの位置を、ドキュメントのソースコードにおける文字オフセットとして取得します。これは、ローカルドライブ上のすべてのドキュメントに使用できます。

引数

node

- *node* 引数には、dreamweaver.getDocumentDOM() 関数によって返されるツリー内のノードであるタグ、コメント、またはテキスト範囲を指定する必要があります。

戻り値

2 つの整数を含む配列。最初の整数は、タグ、テキスト、またはコメントの開始点を示す文字オフセットです。2 番目の整数は、HTML ドキュメントの先頭を基準とした、ノードの終了点を示す文字オフセットです。

例

次のコードは、現在のドキュメントにある最初のイメージオブジェクトを選択します。

```
var theDOM = dw.getDocumentDOM();
var theImg = theDOM.images[0];
var offsets = theDom.nodeToOffsets(theImg);
theDom.setSelection(offsets[0], offsets[1]);
```

dom.offsetsToNode()

対応バージョン

Dreamweaver 3

説明

指定した開始点と終了点の間にあるすべての文字を完全に含む、DOM ツリー内のオブジェクトを取得します。これは、ローカルドライブ上のすべてのドキュメントに使用できます。

引数

offsetBegin、*offsetEnd*

- *offsetBegin* 引数には、ドキュメントの先頭から DOM ツリーに含まれるオブジェクトである文字範囲の開始点までのオフセットを指定します。
- *offsetEnd* 引数には、ドキュメントの先頭から DOM ツリーに含まれるオブジェクトである文字範囲の終了点までのオフセットを指定します。

戻り値

指定した範囲のすべての文字を含むタグ、テキスト、またはコメントオブジェクト。

例

次のコードは、選択範囲がイメージである場合に警告を表示します。

```
var offsets = dom.getSelection();
var theSelection = dreamweaver.offsetsToNode(offsets[0], ¬
offsets[1]);
if (theSelection.nodeType == Node.ELEMENT_NODE && ¬
theSelection.tagName == 'IMG'){
    alert('The current selection is an image.');
}
```

dom.selectAll()

対応バージョン

Dreamweaver 3

説明

[すべて選択] と同じ操作を実行します。

注意 : ほとんどの場合、この関数はアクティブなドキュメント内のすべてのコンテンツを選択します。ただし、挿入ポイントがテーブル内にある場合などには、アクティブなドキュメントの一部だけを選択することがあります。ドキュメント全体を選択範囲にするには、*dom.setSelection()* を使用してください。

引数

なし

戻り値

なし

dom.setSelectedNode()

対応バージョン

Dreamweaver 3

説明

選択したノードを設定します。この関数は、dom.nodeToOffsets() 関数を呼び出して、その戻り値を dom.setSelection() 関数に渡す場合と同じ操作を行います。

引数

node、*{bSelectInside}*、*{bJumpToNode}*

- *node* 引数には、ドキュメント内のテキスト、コメント、またはエレメントノードを指定します。
- *bSelectInside* 引数 (オプション) には、このノードの innterHTML を選択するかどうかを示すブール値を指定します。この引数は、*node* がエレメントノードである場合にだけ指定します。指定しない場合のデフォルト値は false です。
- *bJumpToNode* 引数 (オプション) には、必要に応じて選択範囲を表示できるように、ドキュメントウィンドウをスクロールするかどうかを示すブール値を指定します。この引数を指定しない場合のデフォルト値は false です。

戻り値

なし

dom.setSelection()

対応バージョン
Dreamweaver 3

説明
ドキュメントで選択範囲を設定します。

引数
offsetBegin、*offsetEnd*

- これらの引数には、新規選択範囲の開始点と終了点を、ドキュメントのソースコードにおけるバイトオフセットとしてそれぞれ指定します。これら 2 つの整数が同じ値である場合、新規選択範囲が挿入ポイントであることを示します。新規選択範囲が有効な HTML 選択範囲ではない場合、最初の有効な HTML 選択範囲の文字が含まれるように拡張されます。たとえば、*offsetBegin* と *offsetEnd* によって定義される範囲が `<IMG SRC="myImage.gif">` のうち `SRC="myImage.gif"` という部分だけである場合、`IMG` タグ全体が含まれるようにこの選択範囲が拡張されます。

戻り値
なし

dreamweaver.getSelection() (非推奨)

対応バージョン
Dreamweaver 2 (3 では非推奨)。詳細については、237 ページの dom.getSelection() を参照してください。

説明
現在のドキュメント内の選択範囲を、ドキュメントのソースコードにおけるバイトオフセットとして取得します。

引数
なし

戻り値
2 つの整数を含む配列。最初の整数は選択範囲の開始点のバイトオフセットを、2 番目の整数は選択範囲の終了点のバイトオフセットをそれぞれ示します。これら 2 つの整数が同じ値である場合は、現在の選択範囲が挿入ポイントであることを示します。

dreamweaver.nodeExists()

対応バージョン
Dreamweaver 3

説明
指定したノードに対する参照が現在も存在するかどうかを判別します。拡張機能のプログラミング時には、ノードの親の `innerHTML` プロパティや `outerHTML` プロパティを設定するときなどに、ノードを参照してから、そのノードを削除することがあります。この関数を使用すると、ノードのプロパティやメソッドを参照する前に、そのノードが削除されていないことを確認することができます。参照されるノードは、現在のドキュメント内に存在する必要はありません。

引数
node

- *node* 引数には、チェックするノードを指定します。

戻り値

ノードが存在する場合は true、存在しない場合は false のブール値。

例

次の例では、現在のノードを取得し、そのノード内でテーブルを検索します。さらに、後から dw.nodeExists() を呼び出して元のノードがまだ存在するかどうかを確認します。

```
function applyFormatToSelectedTable(){

    // 現在の選択範囲を取得します。
    var selObj = dw.getDocumentDOM().getSelectedNode();
    alternateRows(dwscripts.findDOMObject("presetNames").selectedIndex,
        findTable());

// 元の選択範囲が存在する場合は、その選択範囲を復元します。
// 存在しない場合は、テーブルを選択します。
    var selArr;
    if (dw.nodeExists(selObj))
        selArr = dom.nodeToOffsets(selObj);
    else
        selArr = dom.nodeToOffsets(findTable());

    dom.setSelection(selArr[0],selArr[1]);
}
```

dreamweaver.nodeToOffsets() (非推奨)

対応バージョン

Dreamweaver 2 (3 では 238 ページの dom.nodeToOffsets() に置き換えられるため非推奨)

説明

DOM ツリー内の特定のノードの位置を、ドキュメントのソースコードにおけるバイトオフセットとして取得します。

引数

node

- *node* 引数には、dreamweaver.getDocumentDOM() 関数によって返されるツリー内のノードであるタグ、コメント、またはテキスト範囲を指定する必要があります。

戻り値

2 つの整数を含む配列。最初の整数はタグ、テキスト、またはコメントの開始点のバイトオフセットを、2 番目の整数はノードの終了点のバイトオフセットをそれぞれ示します。

dreamweaver.offsetsToNode() (非推奨)

対応バージョン

Dreamweaver 2 (3 では 238 ページの dom.offsetsToNode() に置き換えられるため非推奨)

説明

指定した開始点と終了点の間にあるすべての文字を完全に含む、DOM ツリー内のオブジェクトを取得します。

引数

offsetBegin、*offsetEnd*

- これらの引数には、文字範囲の開始点と終了点を、ドキュメントのソースコード内におけるバイトオフセットとしてそれぞれ指定します。

戻り値

指定した範囲のすべての文字を含むタグ、テキスト、またはコメントオブジェクト。

dreamweaver.selectAll()

対応バージョン

Dreamweaver 3

説明

アクティブなドキュメントウィンドウ、[サイト] パネル、または Macintosh の場合はダイアログボックスまたはフローティングパネル内のフォーカスが置かれているテキストフィールドで、[すべて選択] の操作を実行します。

注意 : この操作をアクティブなドキュメントで実行すると、通常はそのドキュメント内のすべてのコンテンツが選択されます。ただし、挿入ポイントがテーブル内にある場合などには、アクティブなドキュメントの一部だけが選択されることがあります。ドキュメント全体を選択範囲にするには、*dom.setSelection()* 関数を使用してください。

引数

なし

戻り値

なし

イネーブラ

詳細については、423 ページの dreamweaver.canSelectAll() を参照してください。

dreamweaver.setSelection() (非推奨)

対応バージョン

Dreamweaver 2 (3 では 240 ページの dom.setSelection() に置き換えられるため非推奨)

説明

現在のドキュメントで選択範囲を設定します。この関数は、現在のドキュメント内でのみ選択範囲を移動できます。別のドキュメントにフォーカスを移すことはできません。

引数

offsetBegin、*offsetEnd*

- これらの引数には、新規選択範囲の開始点と終了点を、ドキュメントのソースコードにおけるバイトオフセットとしてそれぞれ指定します。これら 2 つの整数が同じ値である場合、新規選択範囲が挿入ポイントであることを示します。新規選択範囲が有効な HTML 選択範囲ではない場合、最初の有効な HTML 選択範囲の文字が含まれるように拡張されます。たとえば、*offsetBegin* と *offsetEnd* によって定義される範囲が `<IMG SRC="myImage.gif">` のうち `SRC="myImage.gif"` という部分だけである場合、`IMG` タグ全体が含まれるようにこの選択範囲が拡張されます。

戻り値

なし

ストリング操作関数

ストリング操作関数は、ストリングについての情報の入手や、ストリングを Latin 1 エンコードとプラットフォームネイティブのエンコード間で変換する場合に使用します。

dreamweaver.doURLEncoding()

対応バージョン

Dreamweaver 1

説明

この関数は、ストリングを取得して、すべてのスペースと特殊文字を指定したエンティティに置換し、URL エンコードされたストリングを返します。

引数

stringToConvert

- *stringToConvert* 引数には、関数でエンコードする対象であるエンコード前の URL を含むストリングを指定します。

戻り値

URL エンコードされたストリング。

例

以下の例は、"My URL-encoded string" の URL.value を示しています。

```
var URL = dw.doURLEncoding(theURL.value);
returns "My%20URL-encoded%20string"
```

dreamweaver.getTokens()

対応バージョン

Dreamweaver 1

説明

ストリングを受け入れて複数のトークンに分割します。

引数

searchString、*separatorCharacters*

- *searchString* 引数には、トークンに分割するストリングを指定します。
- *separatorCharacters* 引数には、トークンに区切る場所を示す文字を指定します。トークンを区切る文字が引用符で囲まれている場合は無視されます。separatorCharacters に含まれるすべてのホワイトスペース文字 (タブなど) は、明示的に指定しなくても区切り文字として処理されます。2 つ以上連続しているホワイトスペースは、1 つの区切り文字として処理されます。

戻り値

トークンストリングの配列。

例

次のように dw.getTokens() 関数を呼び出すと、その後に示すトークンが返されます。

```
dreamweaver.getTokens('foo("my arg1", 34)', '(),')
```

- foo
- "my arg 1"
- 34

dreamweaver.latin1ToNative()

対応バージョン
Dreamweaver 2

説明
Latin 1 エンコードのストリングを、ユーザーが使用しているコンピュータのネイティブエンコードに変換します。この関数は、拡張機能ファイルの UI を別の言語で表示するために使用します。

注意 : *Windows* のエンコードは既に *Latin 1* が基準となっているため、この関数は *Windows* では何も効果はありません。

引数
stringToConvert

- *stringToConvert* 引数には、Latin 1 エンコードからネイティブエンコードに変換するストリングを指定します。

戻り値
変換後のストリング。

dreamweaver.nativeToLatin1()

対応バージョン
Dreamweaver 2

説明
ネイティブエンコードのストリングを、Latin 1 エンコードに変換します。

注意 : *Windows* のエンコードは既に *Latin 1* が基準となっているため、この関数は *Windows* では何も効果はありません。

引数
stringToConvert

- *stringToConvert* 引数には、ネイティブエンコードから Latin 1 エンコードに変換するストリングを指定します。

戻り値
変換後のストリング。

dreamweaver.scanSourceString()

対応バージョン
Dreamweaver UltraDev 1 (英語版)

説明
HTML のストリングをスキャンし、タグ、属性、ディレクティブ、およびテキストを検索します。検出されたタグ、属性、ディレクティブ、およびテキストスパンのそれぞれに対して、`scanSourceString()` 関数は指定されたコールバック関数を呼び出します。Dreamweaver は次のコールバック関数をサポートしています。

- `openTagBegin()`
- `openTagEnd()`
- `closeTagBegin()`
- `closeTagEnd()`
- `directive()`
- `attribute()`
- `text()`

Dreamweaver がこの 7 つのコールバック関数を呼び出すのは、以下の場合です。

1 Dreamweaver は、開始タグ (に対応する など) や、空のタグ (<img>、<hr> など) それぞれに対し、openTagBegin() を呼び出します。openTagBegin() 関数は、タグの名前 ("font" や "img" など) とドキュメントオフセットの 2 つの引数を受け取ります。ドキュメントオフセットは、ドキュメントの先頭からタグの開始点までのバイト数です。この関数は、スキャンを続行する必要のある場合は true を返し、停止する必要のある場合は false を返します。

2 Dreamweaver は openTagBegin() を実行した後、HTML 属性それぞれに対して attribute() を呼び出します。attribute() 関数は、2 つの引数を受け取ります。1 つは属性名 ("color"、"src" など) を含むストリングで、もう 1 つは属性値 ("#000000"、"foo.gif" など) を含むストリングです。attribute() 関数は、スキャンを続行する必要があるかどうかを示すブール値を返します。

3 Dreamweaver は、タグ内のすべての属性をスキャンした後、openTagEnd() を呼び出します。openTagEnd() 関数は、ドキュメントの開始タグの末尾までのバイト数を示す 1 つの引数 (ドキュメントオフセット) を受け取ります。また、この関数は、スキャンを続行する必要があるかどうかを示すブール値を返します。

4 Dreamweaver は、 などの終了タグそれぞれに対して、closeTagBegin() を呼び出します。この関数は、終了タグの名前 ("font" など) とドキュメントオフセットの 2 つの引数を受け取ります。ドキュメントオフセットは、ドキュメントの終了タグの先頭までのバイト数です。また、この関数は、スキャンを続行する必要があるかどうかを示すブール値を返します。

5 closeTagBegin() が値を返した後、Dreamweaver は closeTagEnd() 関数を呼び出します。closeTagEnd() 関数は、ドキュメントの終了タグの末尾までのバイト数を示す 1 つの引数 (ドキュメントオフセット) を受け取ります。また、この関数は、スキャンを続行する必要があるかどうかを示すブール値を返します。

6 Dreamweaver は、HTML コメント、ASP スクリプト、JSP スクリプト、または PHP スクリプトそれぞれに対して、directive() 関数を呼び出します。directive() 関数は、ディレクティブを含むストリングとドキュメントオフセットの 2 つの引数を受け取ります。ドキュメントオフセットは、ドキュメントの終了タグの末尾までのバイト数です。また、この関数は、スキャンを続行する必要があるかどうかを示すブール値を返します。

7 Dreamweaver は、ドキュメント内の各テキストのスパン、つまりタグとディレクティブを除くすべてに対して、text() 関数を呼び出します。テキストスパンには、ユーザーには表示されない <title> タグまたは <option> タグ内のテキストが含まれます。text() 関数は、テキストを含むストリングとドキュメントオフセットの 2 つの引数を受け取ります。ドキュメントオフセットは、ドキュメントの終了タグの末尾までのバイト数です。また、text() 関数は、スキャンを続行する必要があるかどうかを示すブール値を返します。

引数

HTMLstr、*parserCallbackObj*

- *HTMLstr* 引数には、コードを含むストリングを指定します。
- *parserCallbackObj* 引数は、openTagBegin()、openTagEnd()、closeTagBegin()、closeTagEnd()、directive()、attribute()、および text() のうち少なくとも 1 つのメソッドを含む JavaScript オブジェクトです。最高のパフォーマンスを得るために、*parserCallbackObj* には、C レベル拡張機能インターフェイスで定義した共有ライブラリを指定する必要があります。*parserCallbackObj* で必要なコールバック関数だけを定義することによっても、パフォーマンスを向上することができます。

戻り値

操作が正常に終了した場合は true、失敗した場合は false のブール値。

例

以下の一連の手順は、dreamweaver.scanSourceString() 関数の使用方法を示しています。

1 7 つあるコールバック関数のいずれか (複数可) に対して実装を作成します。

2 dreamweaver.scanSourceString() 関数を呼び出すスクリプトを記述します。

3 dreamweaver.scanSourceString() 関数は、HTML とポインタを含むストリングを、記述したコールバック関数に渡します。たとえば、HTML のストリングが "hello" だとします。

4 Dreamweaver はこのストリングを分析して、ストリングに font タグが含まれるかどうかを判別します。Dreamweaver は、次の順序でコールバック関数を呼び出します。

- `openTagBegin()` 関数
- `attribute()` 関数 (size 属性に対して)
- `openTagEnd()` 関数
- `text()` 関数 (`"hello"` ストリングに対して)
- `closeTagBegin()` 関数と `closeTagEnd()` 関数

トランスレート関数

トランスレート関数は、トランスレータを直接処理するか、またはトランスレートの結果を処理します。これらの関数を使用して、トランスレータに関する情報の取得、トランスレータの実行、およびロックされた領域内のコンテンツの編集を行うことができます。また、選択範囲のオフセットの取得や設定の際に、トランスレートされたソースを使用するように指定することもできます。

dom.runTranslator()

対応バージョン

Dreamweaver 3

説明

この関数は、指定したトランスレータを特定のドキュメントに対して実行します。この関数はアクティブなドキュメントのみに適用されます。

引数

`translatorName`

- *`translatorName`* 引数には、トランスレータの名前を [トランスレート] 環境設定に表示されるとおりに指定します。

戻り値

なし

dreamweaver.editLockedRegions()

対応バージョン

Dreamweaver 2

説明

ロックされた領域を、引数の値に応じて編集可能または編集不可に設定します。デフォルトでは、ロックされた領域は編集できません。この関数を使用して編集可能にする前に、ロックされている領域を編集しようとすると、警告音が鳴り、編集は許可されません。

注意 : ロックされた領域を編集すると、ライブラリ項目やテンプレートに予想外の影響が生じる可能性があります。この関数は、データトランスレータのコンテキストの範囲外では使用しないでください。

引数

`bAllowEdits`

- *`bAllowEdits`* 引数はブール値です。編集可能にするには `true`、それ以外の場合は `false` を指定します。この関数を呼び出したスクリプトの実行が完了すると、Dreamweaver ではロックされた領域の状態が自動的にデフォルト (編集不可) に戻されます。

戻り値

なし

dreamweaver.getTranslatorList()

対応バージョン

Dreamweaver 3

説明

この関数は、インストールされているトランスレータのリストを取得します。

引数

なし

戻り値

それぞれが [トランスレート] 環境設定に表示されるトランスレータの名前を表すストリングの配列。

dreamweaver.useTranslatedSource()

対応バージョン

Dreamweaver 2

説明

`dom.nodeToOffsets()` および `dom.getSelection()` によって返される値を指定します。これらの値は、`dom.offsetsToNode()` および `dom.setSelection()` によって使用されます。また、トランスレートされていないソースではなく、トランスレートされたソース (トランスレータを実行した後に DOM に格納される HTML) におけるオフセットである必要があります。

注意 : この関数は、プロパティインスペクタファイル内でのみ使用できます。

引数

`bUseTranslatedSource`

- *`bUseTranslatedSource`* 引数はブール値です。トランスレートされたソースにおけるオフセットを関数で使用するには `true`、トランスレートされていないソースを使用するには `false` を指定します。

この引数のデフォルト値は `false` です。引数に `false` を指定して `dw.useTranslatedSource()` を明示的に呼び出していない場合、呼び出し元のスクリプトの実行が完了すると、Dreamweaver ではその後の `dw.getSelection()`、`dw.setSelection()`、`dw.nodeToOffsets()`、および `dw.offsetsToNode()` の呼び出しに、トランスレートされていないソースが使用されます。

戻り値

なし

XSLT 関数

XSLT 関数は、XML ファイルを処理します。これらの関数は、スキーマツリーや XML ドキュメントへの参照などの XML ドキュメントに関する情報を取得し、現在の XSLT ドキュメントに関連付けられた XML ドキュメントを指定するようユーザーに指示します。

MMXSLT.getXML()

対応バージョン
Dreamweaver CS3

説明
XML ファイルの XML ソースを表すストリングを取得します。

引数
xmlSourceURI

- XML ファイルへの URI を表すストリングです。絶対パス (http または https)、サイトルート相対パス、またはドキュメント相対パスのいずれかで指定します。

戻り値
XML ファイルのコンテンツを含むストリング。

例
```
var xmlSource = MMXSLT.getXML(this.fileDataSetURL);
```

MMXSLT.getXMLSchema()

対応バージョン
Dreamweaver 8

説明
この関数は、指定した XML ファイルのスキーマツリーを返します。

引数
schemaURI、*{bRefresh}*

- *schemaURI* 引数 (必須) は、ローカルまたはリモートの XML ファイルへの参照であるストリングです。
- *bRefresh* 引数 (オプション) はブール値です。スキーマを強制的に更新するには `true`、XML スキーマキャッシュからスキーマのコピーを返すには `false` を指定します。デフォルト値は `false` です。

戻り値
XML スキーマツリーを含むストリング。

例
次の例では、"menus.xml" の XML スキーマキャッシュからスキーマツリーを取得します。

```
var theSchema = MMXSLT.getXMLSchema("file:///c:/Program Files/Adobe/¬
    Adobe Dreamweaver CS3/Configuration/Menus/menus.xml"");
```

MMXSLT.getXMLSourceURI()

対応バージョン
Dreamweaver 8

説明
この関数は、現在の XSLT ドキュメントに関連付けられた XML ソースドキュメントへの参照を取得します。

引数

xsltfileURI、*{bUseTempForRemote}*

- *xsltfileURI* 引数は、XML ファイルの場所を指すローカルファイル URI であるストリングです。
- *bUseTempForRemote* 引数 (オプション) はブール値です。true の場合は、元の XML ファイルがリモート (たとえば、http://myHost/rssfeed.xml) にあるときにダウンロードされている一時 XML ファイルへの参照 (たとえば、file:///C:/Documents and Settings/<ユーザー名>/Local Settings/Temporary Internet Files/Content.IE5/GTSLQ9KZ/rss[1].xml) を返します。false の場合は、絶対参照を返します。

戻り値

現在の XSLT ドキュメントに関連付けられた XML ソースドキュメントへの参照を含むストリング。XML ソースの参照がリモート参照である場合は、一時的な場所にダウンロードされているファイルへのパスを返します。

例

次の例では、"c:¥myxslt¥myxsltdocument.xsl" に関連付けられた XML ソースドキュメントへの参照を取得します。

```
var theXMLSource = MMXSLT.getXMLSourceURI("file:///c:/myxslt/myxsltdocument.xsl");
```

MMXSLT.launchXMLSourceDialog()

対応バージョン

Dreamweaver 8

説明

この関数は、現在の XSLT ドキュメントに関連付けられた XML ソースドキュメントを指定するようユーザーに指示します。ユーザーは、XML ドキュメントのローカル参照またはリモート参照のいずれかを選択できます。

引数

{xsltfileURI、*bUseTempForRemote*、*bAddSchemaReference}*

- *xsltfileURI* 引数はオプションです。XSL ファイルの場所を指すローカルファイル URI のストリングを指定します。この引数を指定しない場合のデフォルト値は、現在開いているドキュメントの URI です。
- *bUseTempForRemote* 引数 (オプション) はブール値です。true の場合は、元の XML ファイルがリモート (たとえば、http://myHost/rssfeed.xml) にあるときにダウンロードされている一時 XML ファイルへの参照 (たとえば、file:///C:/Documents and Settings/<ユーザー名>/Local Settings/Temporary Internet Files/Content.IE5/GTSLQ9KZ/rss[1].xml) を返します。false の場合は、絶対参照を返します。
- *bAddSchemaReference* 引数はオプションです。[XML ソースの指定] ダイアログボックスで指定した XML ソースの URI を指す参照が現在のドキュメントに追加されます。この引数を指定しない場合のデフォルト値は、現在開いているドキュメントの URI です。

戻り値

現在の XSLT ドキュメントに関連付けられた XML ソースドキュメントへの参照を含むストリング。XML ソースの参照がリモート参照である場合は、一時的な場所にダウンロードされているファイルへのパスを返します。

例

次の例では、値を指定せずに [XML ソースの指定] ダイアログボックスを開きます。

```
MMXSLT.launchXMLSourceDialog()
```

第 15 章 : ページコンテンツ

Adobe® Dreamweaver® CS3 のページコンテンツ関数は、Web ページのコンテンツに影響を与える操作を行います。ページコンテンツ関数が行う操作には、[アセット] パネルでのアセットの操作、ビヘイビアの追加、クリップボードからのエレメントのカットおよびペースト、テンプレートの適用、コードスニペットの挿入、Spry XML データセットの作成、Spry およびその他の Widget の機能強化された編集、Widget の挿入などがあります。本章では、どのブラウザでも正しく動作するページレイアウトの作成に役立つ、ブラウザ互換性チェック関数についても説明します。

本章では関数をグループ分けして、それぞれを以下の項で説明します。

アセットパネル関数

" アセットパネル " として API にプログラムされているアセットパネル関数を使用すると、[アセット] パネル内のエレメント (テンプレート、ライブラリ、イメージ、Adobe Shockwave コンテンツと Adobe Flash コンテンツ、URL、カラー、およびスクリプト) を管理および使用できます。

dreamweaver.assetPalette.addToFavoritesFromDocument()

対応バージョン

Dreamweaver 4

説明

ドキュメントウィンドウで選択されているエレメントを [お気に入り] リストに追加します。この関数は、イメージ、Shockwave ファイル、Flash ファイル、テキストフォントカラー、および URL のみを処理します。

引数

なし

戻り値

なし

dreamweaver.assetPalette.addToFavoritesFromSiteAssets()

対応バージョン

Dreamweaver 4

説明

[サイト] リストで選択されているエレメントを [お気に入り] リストに追加し、各項目に [お気に入り] リストでのニックネームを付けます。この関数では、[サイト] リストからのエレメントの削除は実行されません。

引数

なし

戻り値

なし

dreamweaver.assetPalette.addToFavoritesFromSiteWindow()

対応バージョン

Dreamweaver 4

説明

[サイト] パネルまたはサイトマップで選択されているエレメントを [お気に入り] リストに追加します。この関数は、イメージ、ムービー、スクリプト、Shockwave ファイル、Flash ファイル、および URL (サイトマップを使用した場合) のみを処理します。その他のフォルダまたはファイルが選択されている場合は、無視します。

引数

なし

戻り値

なし

dreamweaver.assetPalette.copyToSite()

対応バージョン

Dreamweaver 4

説明

選択されているエレメントを別のサイトにコピーし、そのサイトの [お気に入り] リストに追加します。エレメントがファイル (カラーや URL 以外) の場合は、実際のファイルがそのサイトにコピーされます。

引数

targetSite

- *targetSite* 引数は、`site.getSites()` を呼び出したときに返されるコピー先サイトの名前です。

戻り値

なし

dreamweaver.assetPalette.edit()

対応バージョン

Dreamweaver 4

説明

選択されているエレメントを、プライマリ外部エディタまたはカスタム編集コントロールを使用して編集します。カラーの場合、カラーピッカーが表示されます。URL の場合は、URL およびニックネームを入力するためのダイアログボックスが表示されます。この関数は、カラーと URL の [サイト] リストには使用できません。

引数
なし

戻り値
なし

イネーブラ
詳細については、416 ページの dreamweaver.assetPalette.canEdit() を参照してください。

dreamweaver.assetPalette.getSelectedCategory()

対応バージョン
Dreamweaver 4

説明
この関数は、現在選択されているカテゴリを返します。

引数
なし

戻り値
現在選択されているカテゴリ。カテゴリは、"templates"、"library"、"images"、"movies"、"shockwave"、"flash"、"scripts"、"colors"、または "urls" のいずれかです。

dreamweaver.assetPalette.getSelectedItems()

対応バージョン
Dreamweaver 4

説明
[アセット] パネルの [サイト] リストまたは [お気に入り] リストで選択されている項目の配列を返します。

引数
なし

戻り値
選択されている項目ごとに以下の 3 つのストリングが格納された配列。

- *name* ストリングは、[アセット] パネルに表示される名前 (ファイル名) またはニックネームです。
- value ストリングは、フルパス、完全な URL、またはカラー値です (選択されている項目による)。
- *type* ストリングには、"folder" またはカテゴリを指定します。カテゴリは、"templates"、"library"、"images"、"movies"、"shockwave"、"flash"、"scripts"、"colors"、または "urls" のいずれかです。

注意 : [アセット] パネルで何も選択されていない場合は、空白のストリングを *1* つ含む配列を返します。

例
カテゴリが URL で、"MyFolderName" というフォルダと、"MyFavoriteURL" という URL が両方とも [お気に入り] リストで選択されている場合は、次のリストを返します。

```
items[0] = "MyFolderName"
items[1] = "//path/FolderName"
items[2] = "folder"
items[3] = "MyFavoriteURL"
items[4] = "http://www.MyFavoriteURL.com"
items[5] = "urls"
```

dreamweaver.assetPalette.getSelectedView()

対応バージョン
Dreamweaver 4

説明
[アセット] パネルで現在表示されているリストを示します。

引数
なし

戻り値
`"site"` または `"favorites"` のいずれかの値を持つストリングを返します。

dreamweaver.assetPalette.insertOrApply()

対応バージョン
Dreamweaver 4

説明
選択されているエレメントを挿入するか、現在の選択範囲に適用します。テンプレート、カラー、および URL が選択範囲に適用され、さらに、URL とその他のエレメントが挿入ポイントの位置に挿入されます。ドキュメントが開いていない場合は、この関数は使用できません。

引数
なし

戻り値
なし

イネーブラ
詳細については、416 ページの dreamweaver.assetPalette.canInsertOrApply() を参照してください。

dreamweaver.assetPalette.locateInSite()

対応バージョン
Dreamweaver 4

説明
[サイト] パネルのローカルファイルペインで選択されたエレメントに関連付けられているファイルを選択します。この関数は、カラーおよび URL に対しては機能しません。[サイト] リストおよび [お気に入り] リストの両方で使用できます。[お気に入り] リストでフォルダが選択されている場合は、フォルダは無視されます。

引数
なし

戻り値
なし

dreamweaver.assetPalette.newAsset()

対応バージョン
Dreamweaver 4

説明
[お気に入り] リストで、現在のカテゴリの新規エレメントを作成します。ライブラリとテンプレートの場合、ユーザーがすぐに名前を付けることができる新規の空白のライブラリまたはテンプレートのファイルが作成されます。カラーの場合、カラーピッカーが表示されます。URL の場合は、URL およびニックネームを入力するためのダイアログボックスが表示されます。この関数は、イメージ、Shockwave ファイル、Flash ファイル、およびスクリプトには使用できません。

引数
なし

戻り値
なし

dreamweaver.assetPalette.newFolder()

対応バージョン
Dreamweaver 4

説明
現在のカテゴリの新規フォルダをデフォルトの名前 (untitled) で作成し、テキストボックスにデフォルト名を表示します。[お気に入り] リストでのみ使用できます。

引数
なし

戻り値
なし

dreamweaver.assetPalette.recreateLibraryFromDocument()

対応バージョン
Dreamweaver 4

説明
非推奨の `libraryPalette` 関数である `recreateLibraryFromDocument()` に代わるものです。現在のドキュメントで選択されているライブラリ項目のインスタンスに対して、Library Item (LBI) ファイルを作成します。この関数は、プロパティインスペクタで [再作成] をクリックした場合と同じ操作を実行します。

引数
なし

戻り値
なし

dreamweaver.assetPalette.refreshSiteAssets()

対応バージョン
Dreamweaver 4

説明
サイトをスキャンして、[サイト] リストに切り替え、リストの値を表示します。

引数
なし

戻り値
なし

dreamweaver.assetPalette.removeFromFavorites()

対応バージョン
Dreamweaver 4

説明
選択されているエレメントを [お気に入り] リストから削除します。この関数は、ディスク上の実際のファイルは削除しません。ただし、ライブラリまたはテンプレートの場合は例外で、ファイルを削除する前に確認のメッセージを表示します。[お気に入り] リストで、またはカテゴリがライブラリまたはテンプレートの場合にのみ機能します。

引数
なし

戻り値
なし

dreamweaver.assetPalette.renameNickname()

対応バージョン
Dreamweaver 4

説明
既存のニックネームをテキストボックスに表示して、フォルダ名またはファイルのニックネームを編集します。[お気に入り] リストで、またはカテゴリがライブラリまたはテンプレートの場合にのみ使用できます。

引数
なし

戻り値
なし

dreamweaver.assetPalette.setSelectedCategory()

対応バージョン
Dreamweaver 4

説明
切り替えて別のカテゴリを表示します。

引数
categoryType

- *categoryType* 引数には、カテゴリを指定します。カテゴリは、"templates"、"library"、"images"、"movies"、"shockwave"、"flash"、"scripts"、"colors"、または "urls" のいずれかです。

戻り値
なし

dreamweaver.assetPalette.setSelectedView()

対応バージョン
Dreamweaver 4

説明
[サイト] リストまたは [お気に入り] リストのいずれかに表示を切り替えます。

引数
viewType

- *viewType* 引数には、"site" または "favorites" のストリングを指定します。

戻り値
なし

dreamweaver.libraryPalette.deleteSelectedItem() (非推奨)

対応バージョン
Dreamweaver 3 (Dreamweaver 4 では 255 ページの dreamweaver.assetPalette.setSelectedCategory() の使用後に 255 ページの dreamweaver.assetPalette.removeFromFavorites() を呼び出すため非推奨)

説明
この関数は、選択したライブラリ項目を [ライブラリ] パネルから削除し、その関連する Dreamweaver LBI ファイルを現在のサイトのルートにある "Library" フォルダから削除します。削除された項目のインスタンスは、サイト上のページに残る場合があります。

引数
なし

戻り値
なし

dreamweaver.libraryPalette.getSelectedItem() (非推奨)

対応バージョン
Dreamweaver 3 (4 では 252 ページの dreamweaver.assetPalette.getSelectedItems() に置き換えられるため非推奨)

説明

この関数は、選択したライブラリ項目のパスを取得します。

引数

なし

戻り値

file:// URL 形式で表記された、ライブラリ項目のパスを含むストリング。

dreamweaver.libraryPalette.newFromDocument() (非推奨)

対応バージョン

Dreamweaver 3 (Dreamweaver 4 では 255 ページの dreamweaver.assetPalette.setSelectedCategory() の使用後に 254 ページの dreamweaver.assetPalette.newAsset() を呼び出すため非推奨)

説明

この関数は、現在のドキュメントの選択範囲に基づいて、新規のライブラリ項目を作成します。

引数

bReplaceCurrent

- *bReplaceCurrent* 引数には、選択範囲を新規作成したライブラリ項目のインスタンスに置換するかどうかを示すブール値を指定します。

戻り値

なし

dreamweaver.libraryPalette.recreateFromDocument() (非推奨)

対応バージョン

Dreamweaver 3 (Dreamweaver 4 では 254 ページの dreamweaver.assetPalette.recreateLibraryFromDocument() に置き換えられるため非推奨)

説明

この関数は、現在のドキュメントで選択されているライブラリ項目のインスタンスに対して、LBI ファイルを作成します。この関数は、プロパティインスペクタで [再作成] をクリックした場合と同じ操作を実行します。

引数

なし

戻り値

なし

dreamweaver.libraryPalette.renameSelectedItem() (非推奨)

対応バージョン

Dreamweaver 3 (Dreamweaver 4 では、引数の値に `"library"` を指定して 255 ページの dreamweaver.assetPalette.setSelectedCategory() を使用し、次に 255 ページの dreamweaver.assetPalette.renameNickname() を呼び出すため非推奨)

説明
この関数は、ユーザーが選択範囲の名前を変更できるように、選択したライブラリ項目の名前をテキストフィールドに変えます。

引数
なし

戻り値
なし

dreamweaver.referencePalette.getFontSize()

対応バージョン
Dreamweaver 4

説明
[リファレンス] パネルの表示領域の現在のフォントサイズを返します。

引数
なし

戻り値
`small`、`medium`、または `large` の相対フォントサイズ。

dreamweaver.referencePalette.setFontSize()

対応バージョン
Dreamweaver 4

説明
[リファレンス] パネルに表示されるフォントサイズを変更します。

引数
`fontSize`

- *`fontSize`* 引数は、`small`、`medium`、または `large` のいずれかの相対フォントサイズです。

戻り値
なし

dreamweaver.templatePalette.deleteSelectedTemplate() (非推奨)

対応バージョン
Dreamweaver 3 (Dreamweaver 4 では、引数の値に `"templates"` を指定して 255 ページの dreamweaver.assetPalette.setSelectedCategory() を使用し、次に 255 ページの dreamweaver.assetPalette.removeFromFavorites() を呼び出すため非推奨)

説明
この関数は、選択したテンプレートを "Templates" フォルダから削除します。

引数
なし

戻り値
なし

dreamweaver.templatePalette.getSelectedTemplate() (非推奨)

対応バージョン
Dreamweaver 3 (4 では 252 ページの dreamweaver.assetPalette.getSelectedItems() に置き換えられるため非推奨)

説明
この関数は、選択したテンプレートのパスを取得します。

引数
なし

戻り値
file:// URL 形式で表記された、テンプレートのパスを含むストリング。

dreamweaver.templatePalette.renameSelectedTemplate() (非推奨)

対応バージョン
Dreamweaver 3 (Dreamweaver 4 では、引数の値に `"templates"` を指定して 255 ページの dreamweaver.assetPalette.setSelectedCategory() を使用し、次に 255 ページの dreamweaver.assetPalette.renameNickname() を呼び出すため非推奨)

説明
この関数は、ユーザーが選択範囲の名前を変更できるように、選択したテンプレートの名前をテキストフィールドに変えます。

引数
なし

戻り値
なし

ビヘイビア関数

ビヘイビア関数を使用すると、オブジェクトに対するビヘイビアの追加と削除、オブジェクトに関連付けられているビヘイビアの検索、ビヘイビアが関連付けられているオブジェクトに関する情報の取得などを行うことができます。`dreamweaver.behaviorInspector` オブジェクトのメソッドは、現在のドキュメントの選択範囲ではなく、[ビヘイビア] パネルの選択範囲に対してのみ制御または動作します。

dom.addBehavior()

対応バージョン
Dreamweaver 3

説明

選択されているエレメントに、イベントとアクションのペアを新たに追加します。この関数はアクティブなドキュメントのみに適用されます。

引数

event、*action*、*{eventBasedIndex}*

- *event* 引数には、エレメントにビヘイビアを関連付ける JavaScript イベントハンドラを指定します。たとえば、onClick、onMouseOver、onLoad などです。
- *action* 引数には、[ビヘイビア] パネルを使用してアクションが追加された場合に applyBehavior() が返す関数呼び出しを指定します。たとえば、"MM_popupMsg('Hello World')" などです。
- *eventBasedIndex* 引数 (オプション) には、このアクションを追加する位置を指定します。*eventBasedIndex* 引数はゼロから開始するインデックスです。たとえば、指定されたイベントに既に 2 つのアクションが関連付けられている場合に *eventBasedIndex* で 1 を指定すると、このアクションは既存の 2 つのアクションの間に実行されます。この引数を省略すると、指定したイベントの既存のすべてのアクションの後にこのアクションが追加されます。

戻り値

なし

dom.getBehavior()

対応バージョン

Dreamweaver 3

説明

指定されたイベント内の、指定された位置のアクションを取得します。この関数は現在の選択範囲に対して動作し、アクティブなドキュメントにのみ使用できます。

引数

event、*{eventBasedIndex}*

- *event* 引数には、エレメントにアクションを関連付けるのに使用する JavaScript イベントハンドラを指定します。たとえば、onClick、onMouseOver、onLoad などです。
- *eventBasedIndex* 引数 (オプション) には、アクションを取得する位置を指定します。たとえば、指定したイベントに 2 つのアクションが関連付けられている場合は、0 が最初のアクション、1 が 2 番目のアクションを表します。この引数を省略すると、関数は指定したイベントのすべてのアクションを返します。

戻り値

関数呼び出しを表すストリング (たとえば、
"MM_swapImage('document.Image1','document.Image1','foo.gif','#933292969950')")。
eventBasedIndex が指定されていない場合は、ストリングの配列。

dom.reapplyBehaviors()

対応バージョン

Dreamweaver 3

説明

指定されたノード上のビヘイビア呼び出しに関連付けられている関数がドキュメントの HEAD セクションに存在することを確認し、存在しない場合はその関数を挿入します。

引数

elementNode

- *elementNode* 引数には、現在のドキュメント内のエレメントノードを指定します。この引数を省略した場合は、ドキュメント内のすべてのエレメントノードで単独ビヘイビア呼び出しがあるかどうかのチェックが行われます。

戻り値

なし

dom.removeBehavior()

対応バージョン

Dreamweaver 3

説明

指定されたイベント内の、指定された位置のアクションを削除します。この関数は現在の選択範囲に対して動作し、アクティブなドキュメントにのみ使用できます。

引数

event、*{eventBasedIndex}*

- *event* 引数には、エレメントにアクションを関連付けるのに使用する JavaScript イベントハンドラを指定します。たとえば、onClick、onMouseOver、onLoad などです。この引数を省略した場合は、エレメントのすべてのアクションが削除されます。
- *eventBasedIndex* 引数 (オプション) には、アクションを削除する位置を指定します。たとえば、指定したイベントに 2 つのアクションが関連付けられている場合は、0 が最初のアクション、1 が 2 番目のアクションを表します。この引数を省略した場合は、指定したイベントのすべてのアクションが削除されます。

戻り値

なし

dreamweaver.getBehaviorElement()

対応バージョン

Dreamweaver 2

説明

ビヘイビアを適用するタグに対応する DOM オブジェクトを取得します。この関数は、ビヘイビアアクションファイルの中でのみ使用できます。

引数

なし

戻り値

DOM オブジェクトまたは null 値。この関数は、次の状況下で null 値を返します。

- 現在のスクリプトが [ビヘイビア] パネルのコンテキスト内で実行されていない場合
- [ビヘイビア] パネルがタイムラインのビヘイビアを編集するために使用されている場合
- 現在実行中のスクリプトが dreamweaver.popupAction() により呼び出された場合
- [ビヘイビア] パネルでリンクラッパーにイベントが関連付けられているが、そのリンクラッパーがまだ作成されていない場合
- この関数がアクションファイルの外で使用されている場合

例

dreamweaver.getBehaviorElement() 関数は、262 ページの dreamweaver.getBehaviorTag() と同様に、選択したアクションが選択した HTML タグに適したものであるかどうかを判別するのに使用できますが、タグとその属性についてさらに詳細な情報を得られる点で異なります。次の例に示すように、ターゲットが別のフレームやウィンドウをターゲットとしないハイパーテキストリンク (A HREF) のみに適用できるアクションを作成する場合、[パラメータ] ダイアログボックスのユーザーインターフェイスを初期化する関数の一部として getBehaviorElement() 関数を使用できます。

```
function initializeUI(){
    var theTag = dreamweaver.getBehaviorElement();
    var CANBEAPPLIED = (theTag.tagName == "A" && ¬
    theTag.getAttribute("HREF") != null && ¬
    theTag.getAttribute("TARGET") == null);
    if (CANBEAPPLIED) {
        // アクション UI を表示します。
    } else {
        // ユーザーに対し役立つメッセージを表示します。
        // 明示的なターゲットのないハイパーリンクにのみ
        // このアクションを適用できることを知らせます。
    }
}
```

dreamweaver.getBehaviorEvent() (非推奨)

対応バージョン

Dreamweaver 1.2。Dreamweaver 2 からはアクションがイベントの前に選択されるため非推奨。

説明

この関数は、[ビヘイビア] アクションファイルで、このアクションを引き起こすイベントを取得します。

引数

なし

戻り値

イベントを表すストリング。これは、canAcceptBehavior() 関数に *event* 引数として渡されるのと同じストリングです。

dreamweaver.getBehaviorTag()

対応バージョン

Dreamweaver 1.2

説明

ビヘイビアを適用するタグのソースを取得します。この関数は、アクションファイルの中でのみ使用できます。

引数

なし

戻り値

タグのソースを表すストリング。これは、canAcceptBehavior() 関数に引数 (*HTMLelement*) として渡すのと同じストリングです。この関数をアクションファイルの外で使用すると、空白のストリングが返されます。

例

ハイパーテキストリンク (A HREF) だけに適用できるアクションを作成する場合は、次の例に示すように、[パラメータ] ダイアログボックスのユーザーインターフェイスを初期化する関数で getBehaviorTag() 関数を使用できます。

```
function initializeUI(){
    var theTag = dreamweaver.getBehaviorTag().toUpperCase();
    var CANBEAPPLIED = (theTag.indexOf('HREF') != -1));
    if (CANBEAPPLIED) {
        // アクション UI を表示します。
    } else {
        // ユーザーに対し役立つメッセージを表示します。
        // ハイパーリンクにのみこのアクションを
        // 適用できることを知らせます。
    }
}
```

dreamweaver.popupAction()

対応バージョン

Dreamweaver 2

説明

指定されたビヘイビアアクションの [パラメータ] ダイアログボックスを表示します。ユーザーから見ると、[ビヘイビア] パネルの [アクション] ポップアップメニューでアクションを選択したのと同じことになります。この関数を使用すると、アクション以外の拡張機能ファイルで、ユーザーのドキュメント内のオブジェクトにビヘイビアを関連付けることができます。このとき、ユーザーがダイアログボックスを閉じるまで、他の編集操作を行うことはできません。

注意 : この関数は、*objectTag()* 関数の内部、コマンドファイルのスクリプト、またはプロパティインスペクタファイルで呼び出すことができます。

引数

actionName、*{funcCall}*

- *actionName* 引数には、JavaScript ビヘイビアアクションが記述されている "Configuration/Behaviors/Actions" フォルダ内のファイルの名前を含むストリングを指定します。たとえば、`"Timeline/Play Timeline.htm"` と指定します。
- *funcCall* 引数 (オプション) には、`actionName` で指定したアクションの関数呼び出しを含むストリングを指定します。たとえば、`"MM_playTimeline(...)"` と指定します。この引数を指定する場合の値は、アクションファイル内の `applyBehavior()` 関数によって返されます。

戻り値

ビヘイビアアクションの関数呼び出し。ユーザーが [パラメータ] ダイアログボックスで [OK] をクリックすると、現在のドキュメントにビヘイビアが追加されます。その際、該当する関数がドキュメントの `HEAD` セクションに追加され、他にも `BODY` セクションの冒頭部に HTML が追加されるなどの編集が行われる場合があります。関数呼び出し (たとえば、`"MM_playTimeline(...)"`) はドキュメントに追加されるのではなく、この関数の戻り値となります。

dreamweaver.behaviorInspector.getBehaviorAt()

対応バージョン

Dreamweaver 3

説明

[ビヘイビア] パネルにおいて、指定された位置のイベントとアクションのペアを取得します。

引数

positionIndex

- *positionIndex* 引数には、[ビヘイビア] パネルでのアクションの位置を指定します。リストの最初のアクションの位置は 0 です。

戻り値

次の 2 つの項目から成る配列

- イベントハンドラ
- 関数呼び出しまたは JavaScript ステートメント

例

`positionIndex` はゼロから開始するインデックスです。したがって、[ビヘイビア] パネルにリストが表示されている場合に `dreamweaver.behaviorInspector.getBehaviorAt(2)` 関数を呼び出すと、`"onMouseOver"` および `"MM_changeProp('document.moon','document.moon','src','sun.gif','MG')"` の 2 つのストリングを含む配列が返されます。

dreamweaver.behaviorInspector.getBehaviorCount()

対応バージョン

Dreamweaver 3

説明

イベントハンドラを使用して、現在選択されているエレメントに関連付けられているアクションの数を数えます。

引数

なし

戻り値

エレメントに関連付けられているアクションの数を表す整数。この値は [ビヘイビア] パネルに表示されるアクションの数と同じで、Dreamweaver ビヘイビアアクションとカスタム JavaScript が含まれます。

例

`<A HREF="javascript:setCookie()" onClick="MM_popupMsg('A cookie has been set.');
parent.rightframe.location.href='aftercookie.html'">` というリンクを選択して `dreamweaver.behaviorInspector.getBehaviorCount()` を呼び出すと、2 が返されます。

dreamweaver.behaviorInspector.getSelectedBehavior()

対応バージョン

Dreamweaver 3

説明

選択されているアクションの、[ビヘイビア] パネルでの位置を取得します。

引数

なし

戻り値

選択されたアクションの [ビヘイビア] パネルでの位置を表す整数。アクションが選択されていない場合は、-1 となります。

例

次のように、[ビヘイビア] パネルの最初のアクションが選択されている場合は、`dreamweaver.behaviorInspector.getSelectedBehavior()` 関数を呼び出すと 0 が返されます。

dreamweaver.behaviorInspector.moveBehaviorDown()

対応バージョン

Dreamweaver 3

説明

イベントの範囲内で実行順序を変更して、ビヘイビアアクションの位置を下に移動します。

引数

positionIndex

- *positionIndex* 引数には、[ビヘイビア] パネルでのアクションの位置を指定します。リストの最初のアクションの位置は 0 です。

戻り値

なし

例

[ビヘイビア] パネルが次のように設定されている場合に `dreamweaver.behaviorInspector.moveBehaviorDown(2)` 関数を呼び出すと、`onMouseDown` イベントの [プリロードイメージ] アクションと [プロパティの変更] アクションの位置が入れ替わります。これ以外の位置で `dreamweaver.behaviorInspector.moveBehaviorDown()` 関数を呼び出しても、関数の効果はありません。これは、`onClick` と `onFocus` の各イベントにはビヘイビアが 1 つずつしか関連付けられておらず、位置が 3 のビヘイビアは既に `onMouseDown` グループの一番下にあるからです。

dreamweaver.behaviorInspector.moveBehaviorUp()

対応バージョン

Dreamweaver 3

説明

イベントの範囲内で実行順序を変更して、ビヘイビアの位置を上に移動します。

引数

positionIndex

- *positionIndex* 引数には、[ビヘイビア] パネルでのアクションの位置を指定します。リストの最初のアクションの位置は 0 です。

戻り値

なし

例

[ビヘイビア] パネルが次のように設定されている場合に `dreamweaver.behaviorInspector.moveBehaviorUp(3)` 関数を呼び出すと、`onMouseOver` イベントの [プリロードイメージ] アクションと [プロパティの変更] アクションの位置が入れ替わります。これ以外の位置で `dreamweaver.behaviorInspector.moveBehaviorUp()` 関数を呼び出しても、関数の効果はありません。これは、`onClick` と `onFocus` の各イベントにはビヘイビアが 1 つずつしか関連付けられておらず、位置が 2 のビヘイビアは既に `onMouseDown` グループの一番上にあるからです。

dreamweaver.behaviorInspector.setSelectedBehavior()

対応バージョン

Dreamweaver 3

説明

[ビヘイビア] パネルの、指定された位置にあるアクションを選択します。

引数

positionIndex

- *positionIndex* 引数には、[ビヘイビア] パネルでのアクションの位置を指定します。リストの最初のアクションの位置は 0 です。すべてのアクションの選択を解除するには、*positionIndex* に -1 を指定します。アクションが存在しない位置を指定したときの結果は、-1 を指定したときと同じです。

戻り値

なし

例

[ビヘイビア] パネルが次のように設定されているときに `dreamweaver.behaviorInspector.setSelection(2)` 関数を呼び出すと、`onMouseDown` イベントに関連付けられた [プロパティの変更] アクションが選択されます。

クリップボード関数

クリップボード関数は、カット、コピー、ペーストに関連する関数です。Macintosh の場合、ダイアログボックスやフローティングパネルのテキストフィールドに適用できるクリップボード関数もあります。テキストフィールドに適用できる関数は、`dreamweaver` オブジェクトのメソッドおよび DOM オブジェクトのメソッドとして実装されています。`dreamweaver` バージョンのクリップボード関数は、アクティブなウィンドウ (現在のドキュメントウィンドウ、コードインスペクタ、または [サイト] パネル) の選択範囲に対して動作します。Macintosh の場合、この関数は、テキストボックスが現在のフィールドである場合はその中の選択範囲に対しても動作します。一方、同じ関数の DOM バージョンは、常に指定されたドキュメント内の選択範囲に対して動作します。

dom.clipCopy()

対応バージョン

Dreamweaver 3

説明

選択範囲を定義するすべての HTML マークアップを含めて選択範囲をクリップボードにコピーします。

引数

なし

戻り値

なし

dom.clipCopyText()

対応バージョン

Dreamweaver 3

説明

選択されたテキストを、HTML マークアップを除いてクリップボードにコピーします。

引数

なし

戻り値
なし

イネーブラ
詳細については、408 ページの dom.canClipCopyText() を参照してください。

dom.clipCut()

対応バージョン
Dreamweaver 3

説明
選択範囲を定義するすべての HTML マークアップを含めて選択範囲を削除し、クリップボードにコピーします。

引数
なし

戻り値
なし

dom.clipPaste()

対応バージョン
Dreamweaver 3

説明
クリップボードのコンテンツを現在のドキュメントの現在の挿入ポイントの位置にペーストするか、クリップボードのコンテンツで現在の選択範囲を置き換えます。クリップボードのコンテンツに含まれている HTML は、HTML として解釈されます。

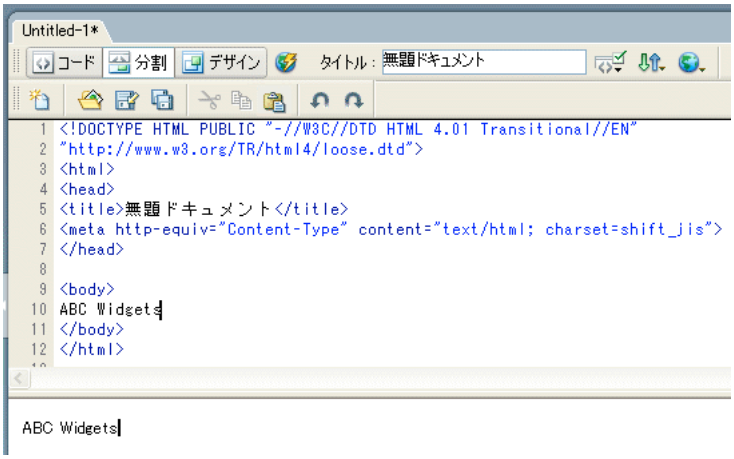
引数
なし

戻り値
なし

イネーブラ
詳細については、408 ページの dom.canClipPaste() を参照してください。

例

クリップボードのコンテンツが ABC Widgets の場合、dw.getDocumentDOM().clipPaste() を呼び出したときの結果は次のとおりです。



dom.clipPasteText() (非推奨)

対応バージョン

Dreamweaver 3。Dreamweaver 8 では非推奨です。代わりに dom.clipPaste("text") 関数を使用してください。

説明

クリップボードのコンテンツを現在のドキュメントの挿入ポイントの位置にペーストするか、クリップボードのコンテンツで現在の選択範囲を置き換えます。クリップボードのコンテンツに含まれている改行は、BR タグに置き換えられます。クリップボードのコンテンツに含まれている HTML は解釈されず、山カッコは < と > としてペーストされます。

引数

なし

戻り値

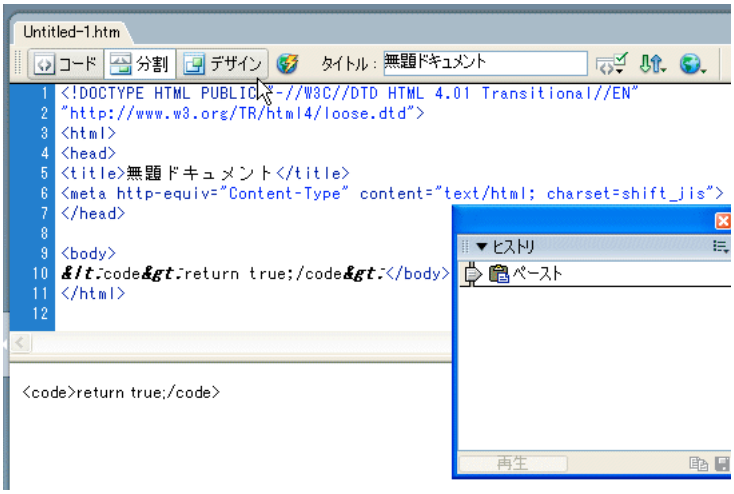
なし

イネーブラ

詳細については、408 ページの dom.canClipPasteText() を参照してください。

例

クリップボードのコンテンツが <code>return true;</code> の場合、dw.getDocumentDOM().clipPasteText() を呼び出したときの結果は次のとおりです。



dreamweaver.clipCopy()

対応バージョン

Dreamweaver 3

説明

アクティブなドキュメントウィンドウ、ダイアログボックス、フローティングパネル、または [サイト] パネルの現在の選択範囲をクリップボードにコピーします。

引数

なし

戻り値

なし

イネーブラ

詳細については、417 ページの dreamweaver.canClipCopy() を参照してください。

dreamweaver.clipCut()

対応バージョン

Dreamweaver 3

説明

アクティブなドキュメントウィンドウ、ダイアログボックス、フローティングパネル、または [サイト] パネルの選択範囲を削除し、クリップボードにコピーします。

引数

なし

戻り値
なし

イネーブラ
詳細については、417 ページの dreamweaver.canClipCut() を参照してください。

dreamweaver.clipPaste()

対応バージョン
Dreamweaver 3 (Dreamweaver 8 で *strPasteOption* 引数を追加)

説明
クリップボードのコンテンツを現在のドキュメント、ダイアログボックス、フローティングパネル、または [サイト] パネルにペーストします。

引数
{strPasteOption}

- *strPasteOption* 引数 (オプション) には、実行するペーストのタイプを指定します。指定可能な値には、"text"、"structured"、"basicFormat"、および "fullFormat" があります。

戻り値
なし

イネーブラ
詳細については、417 ページの dreamweaver.canClipPaste() を参照してください。

例
次の例では、クリップボードのコンテンツをテキストとしてペーストします。

```
dw.clipPaste("text");
```

dreamweaver.getClipboardText()

対応バージョン
Dreamweaver 3

説明
クリップボードに保存されているすべてのテキストを取得します。

引数
{bAsText}

- *bAsText* はオプションのブール値です。クリップボードのコンテンツをテキストとして取得するかどうかを指定します。*bAsText* が true 値の場合は、テキストとして取得されます。*bAsText* が false 値の場合、コンテンツはフォーマットを維持します。引数を省略した場合のデフォルト値は false です。

戻り値
クリップボードのコンテンツがテキスト (HTML も可) の場合はクリップボードのコンテンツが格納されたストリング。それ以外の場合は戻り値はありません。

例
dreamweaver.getClipboardText() が "text bold text" を返す場合、
dreamweaver.getClipboardText(true) は "text bold text" を返します。

ライブラリ関数とテンプレート関数

ライブラリ関数とテンプレート関数は、ドキュメントとライブラリ項目間、またはドキュメントとテンプレート間のリンクの作成、更新、設定解除など、ライブラリ項目やテンプレートに関連する操作を制御します。dreamweaver.libraryPalette オブジェクトのメソッドは、現在のドキュメントの選択範囲ではなく、[アセット] パネルのライブラリ項目の選択範囲に対して制御および作用します。これと同様に、dreamweaver.templatePalette オブジェクトのメソッドは、[アセット] パネルのテンプレートオブジェクトの選択範囲に対して制御および作用します。

dom.applyTemplate()

対応バージョン

Dreamweaver 3

説明

現在のドキュメントにテンプレートを適用します。引数を指定しない場合、[テンプレートの選択] ダイアログボックスが表示されます。この関数は、フォーカスが置かれているドキュメントのみに適用されます。

引数

{templateURL}、*bMaintainLink*

- *templateURL* 引数には、現在のサイトのテンプレートへのパスを file:// URL 形式で指定します。
- *bMaintainLink* 引数には、元のテンプレートへのリンクを維持する (true) か、または維持しない (false) かを示すブール値を指定します。

戻り値

なし

イネーブラ

詳細については、407 ページの dom.canApplyTemplate() を参照してください。

dom.detachFromLibrary()

対応バージョン

Dreamweaver 3

説明

選択範囲の周りのロックタグを削除して、選択したライブラリ項目のインスタンスをその関連 LBI ファイルから切り離します。この関数は、プロパティインスペクタで [編集可能にする] をクリックしたのと同じ操作を実行します。

引数

なし

戻り値

なし

dom.detachFromTemplate()

対応バージョン

Dreamweaver 3

説明
現在のドキュメントをその関連テンプレートから切り離します。

引数
なし

戻り値
なし

dom.getAttachedTemplate()

対応バージョン
Dreamweaver 3

説明
ドキュメントに関連付けられているテンプレートのパスを取得します。

引数
なし

戻り値
file:// URL 形式で表記された、テンプレートのパスを含むストリング。

dom.getEditableRegionList()

対応バージョン
Dreamweaver 3

説明
ドキュメントの本文の編集可能領域すべてのリストを取得します。

引数
なし

戻り値
エレメントノードの配列。

例
274 ページの dom.getSelectedEditableRegion().

dom.getIsLibraryDocument()

対応バージョン
Dreamweaver 3

説明
ドキュメントがライブラリ項目であるかどうかを判別します。

引数
なし

戻り値

ドキュメントが LBI ファイルかどうかを示すブール値。

dom.getIsTemplateDocument()

対応バージョン

Dreamweaver 3

説明

ドキュメントがテンプレートであるかどうかを判別します。

引数

なし

戻り値

ドキュメントが DWT ファイルかどうかを示すブール値。

dom.getSelectedEditableRegion()

対応バージョン

Dreamweaver 3

説明

選択範囲または挿入ポイントが編集可能領域の中にある場合、この関数はドキュメントの本文にある他のすべての編集可能領域におけるこの編集可能領域の位置を取得します。

引数

なし

戻り値

`dom.getEditbableRegionList()` 関数が返す配列のインデックス。詳細については、273 ページの dom.getEditableRegionList() を参照してください。

例

以下のコードは、選択した編集可能領域のコンテンツが表示されたダイアログボックスを開きます。

```
var theDOM = dw.getDocumentDOM();
var edRegs = theDOM.getEditableRegionList();
var selReg = theDOM.getSelectedEditableRegion();
alert(edRegs[selReg].innerHTML);
```

dom.insertLibraryItem()

対応バージョン

Dreamweaver 3

説明

ライブラリ項目のインスタンスをドキュメントに挿入します。

引数

`libraryItemURL`

- *`libraryItemURL`* 引数には、LBI ファイルへのパスを file:// URL 形式で指定します。

戻り値
なし

dom.markSelectionAsEditable()

対応バージョン
Dreamweaver 3

説明
[新規編集可能領域] ダイアログボックスを表示します。ユーザーが [新規領域] をクリックすると、この選択範囲が編集可能領域として指定され、すべてのテキストがそのまま維持されます。

引数
なし

戻り値
なし

イネーブラ
詳細については、412 ページの dom.canMarkSelectionAsEditable() を参照してください。

dom.newEditableRegion()

対応バージョン
Dreamweaver 3

説明
[新規編集可能領域] ダイアログボックスを表示します。ユーザーが [新規領域] をクリックすると、Dreamweaver がドキュメントの挿入ポイントの位置に、波カッコ ({}) で囲まれた領域の名前を挿入します。

引数
なし

戻り値
なし

イネーブラ
詳細については、412 ページの dom.canMakeNewEditableRegion() を参照してください。

dom.removeEditableRegion()

対応バージョン
Dreamweaver 3

説明
ドキュメントから編集可能領域を削除します。編集可能領域内のコンテンツはすべて維持されます。削除されるのは編集可能領域のマーカーだけです。

引数
なし

戻り値

なし

イネーブラ

詳細については、413 ページの dom.canRemoveEditableRegion() を参照してください。

dom.updateCurrentPage()

対応バージョン

Dreamweaver 3

説明

ドキュメントのライブラリ項目、テンプレート、またはその両方を更新します。この関数はアクティブなドキュメントのみに適用されます。

引数

{typeOfUpdate}

- オプションの *typeOfUpdate* 引数には、"library"、"template"、"both" のいずれかを指定する必要があります。引数を省略した場合のデフォルト値は "both" です。

戻り値

なし

dreamweaver.updatePages()

対応バージョン

Dreamweaver 3

説明

[サイト全体を更新] ダイアログボックスを開いて指定したオプションを選択します。

引数

{typeOfUpdate}

- オプションの *typeOfUpdate* 引数を指定する場合は、"library"、"template"、"both" のいずれかを指定する必要があります。この引数を指定しない場合、"both" に初期設定されます。

戻り値

なし

スニペットパネル関数

Dreamweaver を使用して Web アプリケーションを開発するとき、再利用可能なコードブロックを [スニペット] パネルで編集および保存できます。このコードブロックは、必要に応じて取得できます。

[スニペット] パネルは、各コードスニペットを "Configuration/Snippets" フォルダ内の CSN ファイルに保存します。Dreamweaver に付属するスニペットは、以下のフォルダにあります。

- Accessible
- Comments
- Content_tables
- Filelist.txt
- Footers
- Form_elements
- Headers
- Javascript
- Meta
- Navigation
- Text

スニペットのファイルは XML ドキュメントです。したがって、次の例のように XML ディレクティブでエンコードを指定できます。

```
<?XML version="1.0" encoding="utf-8">
```

次に、スニペットファイルの例を示します。

```
<snippet name="Detect Flash" description="VBscript to check for Flash ActiveX control"
preview="code" factory="true" type="wrap" >
    <insertText location="beforeSelection">
        <![CDATA[ ------- code --------- ]]>
    </insertText>
    <insertText location="afterSelection">
        <![CDATA[ ------- code --------- ]]>
    </insertText>
</snippet>
```

CSN ファイルの snippet タグには、以下の属性があります。

属性	説明
name	スニペットの名前。
description	スニペットの説明。
preview	"code" を指定するとプレビュー領域にスニペットが表示され、"design" を指定するとスニペットを HTML で表したものがプレビュー領域に表示されます。
type	ユーザーが選択した範囲をスニペットで囲む場合は "wrap"、選択範囲の前にスニペットを挿入する場合は "block"。

以下のメソッドを使用して、[スニペット] パネルの機能を拡張機能に追加します。

dreamweaver.snippetPalette.getCurrentSnippetPath()

対応バージョン

Dreamweaver MX 2004

説明

[スニペット] パネルで現在選択されているスニペットへのパスを返します。

引数
なし

戻り値
[スニペット] パネルで選択されているスニペットへのパスを、"Snippets" フォルダへの相対で返します。スニペットが選択されていない場合は、空白のストリングを返します。

dreamweaver.snippetPalette.newFolder()

対応バージョン
Dreamweaver MX

説明
新規フォルダを作成してデフォルトの名前 *untitled* を付け、デフォルト名の周りにテキストボックスを表示します。

引数
なし

戻り値
なし

dreamweaver.snippetPalette.newSnippet()

対応バージョン
Dreamweaver MX

説明
[スニペットの追加] ダイアログボックスを開き、これにフォーカスを与えます。

引数
なし

戻り値
なし

dreamweaver.snippetPalette.editSnippet()

対応バージョン
Dreamweaver MX

説明
[スニペットの編集] ダイアログボックスを開いてこれにフォーカスを与え、選択されたエレメントを編集できるようにします。

引数
なし

戻り値
なし

イネーブラ

詳細については、432 ページの dreamweaver.snippetpalette.canEditSnippet() を参照してください。

dreamweaver.snippetPalette.insert()

対応バージョン

Dreamweaver MX

説明

[スニペット] パネルから選択されたスニペットを、現在の選択範囲に適用します。

引数

なし

戻り値

なし

イネーブラ

詳細については、432 ページの dreamweaver.snippetpalette.canInsert() を参照してください。

dreamweaver.snippetPalette.insertSnippet()

対応バージョン

Dreamweaver MX

説明

指定されたスニペットを、現在の選択範囲に挿入します。

引数

path

- スニペットへのパスを、"Snippets" フォルダへの相対で指定するストリング。

戻り値

ブール値。

イネーブラ

詳細については、432 ページの dreamweaver.snippetpalette.canInsert() を参照してください。

例

次のように `dw.snippetPalette.insertSnippet()` 関数を呼び出すと、現在のドキュメントの挿入ポイントに、引数に指定した場所にあるコードスニペットが挿入されます。

```
dw.snippetPalette.insertSnippet('Text\\Different_Link_Color.csn');
```

dreamweaver.snippetPalette.rename()

対応バージョン

Dreamweaver MX

説明

選択されたフォルダ名またはファイルニックネームの周りでテキストボックスをアクティブにし、選択されたエレメントを編集できるようにします。

引数

なし

戻り値

なし

dreamweaver.snippetPalette.remove()

対応バージョン

Dreamweaver MX

説明

[スニペット] パネルで選択されたエレメントまたはフォルダを削除し、ディスクからファイルを削除します。

戻り値

なし

Spry Widget 編集関数

Dreamweaver CS3 には、Spry およびその他の動的 Widget 用に機能強化された編集関数があります。

element.getTranslatedAttribute()

対応バージョン

Dreamweaver CS3

説明

この関数は、トランスレートされた属性に対して動作する点を除いて、W3C の `getAttribute()` 関数と同じです。`element.getTranslatedAttribute()` 関数は、名前で属性値を取得します。

引数

`name`

- *`name`* 引数は、取得する属性の名前を指定する DOMString です。

戻り値

属性の名前を DOMString として返します。属性に指定された値またはデフォルト値がない場合、この関数は空白のストリングを返します。

element.removeTranslatedAttribute()

対応バージョン

Dreamweaver CS3

説明

この関数は、トランスレートされた属性に対して動作する点を除いて、W3C の `removeAttribute()` 関数と同じです。`element.removeTranslatedAttribute()` 関数は、名前で属性を削除します。デフォルト値がある属性の場合、そのデフォルト値と対応する名前空間 URI、ローカル名、および接頭辞 (該当する場合) を格納した状態で表示されます。

引数

name

- *name* 引数は、削除する属性の名前を指定する DOMString です。

戻り値

なし

element.setTranslatedAttribute()

対応バージョン

Dreamweaver CS3

説明

この関数は、トランスレートされた属性に対して動作する点を除いて、W3C の `setAttribute()` 関数と同じです。`element.setTranslatedAttribute()` 関数は、値を指定した新しい属性を追加します。指定した名前の属性がエレメントに既に存在する場合、その属性の値は、*value* 引数で指定した値に変更されます。

value は単純なストリングです。設定されるときには解析されません。そのため、ストリング内のシンタックスは単純なテキストとして処理されます。記述の際に実装により適切にエスケープされるようにする必要があります。

エンティティ参照として認識されることを想定したシンタックスを含む属性値を割り当てるには、`Attr` ノードに加え `Text` ノードと `EntityReference` ノードを作成し、必要なサブツリーを作成して、`setAttributeNode` を使用して属性の値として割り当てます。

引数

name、*value*

- *name* 引数は、作成または変更する属性の名前を指定する DOMString です。
- *value* 引数は、属性に設定する値を指定する DOMString です。

戻り値

なし

element.translatedClassName

対応バージョン

Dreamweaver CS3

説明

この関数は、トランスレートされた `className` 属性に対して動作する点を除いて、`element.ClassName()` 関数と同じです。

element.translatedStyle

対応バージョン

Dreamweaver CS3

説明

この関数は、トランスレートされたスタイル属性に対して動作する点を除いて、element.style() 関数と同じです。

例

```
var div1 = dom.getElementById("div1");
div1.translatedStyle.display = "none";
```

Spry Widget 挿入関数

Dreamweaver には、Spry Widget の挿入を容易にする以下の関数があります。

dom.addJavaScript()

対応バージョン

Dreamweaver CS3

説明

この関数では、Dreamweaver で head または body のいずれかに JavaScript ブロックを挿入するように指定します。挿入先が body 内の場合、JavaScript ブロックは </body> タグの直前に挿入されます。ドキュメントのその位置に既に JavaScript ブロックがある場合、新しい <script> タグは挿入されませんが、<script> のコンテンツに "code" が付加されます。

引数

code、*insideHead*

- *code* は、ページに挿入される JavaScript コードを格納したストリングです。
- *insideHead* は、head または body のどちらに JavaScript ブロックを挿入するかを示すブール値です。デフォルトは true で、この値ではコードが head セクションに挿入されます。false の場合、コードは body の </body> タグの直前に挿入されます。この引数はオプションです。

戻り値

なし

例

```
function objectTag()
{
.
.
.
var dom = dw.getDocumentDOM();
var id = dwscripts.getUniqueId("accordion");
var code = "new Accordion('" + id + "',250,{duration:200,step:20})";
dom.addJavaScript(code, false);

return retVal;
}
```

dom.copyAssets()

対応バージョン

Dreamweaver CS3

説明

この API を使用すると、拡張機能の作成者は外部依存ファイルをユーザーのサイトにコピーして、必要なファイルリファレンスをページの head に追加できます。

引数

assetArray

JavaScript オブジェクトの配列。各 JavaScript オブジェクトには、"srcURL"、"destURL"、"referenceType"、"useDefaultFolder"、および "useRelativeSrc" フィールドがあります。

- "srcURL" は、Dreamweaver 設定フォルダの相対パスです。拡張機能の作成者が提供するアセットが参照されます。この項の "*useRelativeSrc*" の説明を参照してください。
- "destURL" は、Dreamweaver サイトのデフォルトアセットフォルダの相対パスです。アセットのコピー先が参照されます。デフォルトでは、サイト内に "Assets" フォルダが作成され、このフォルダがデフォルトアセットフォルダとして使用されます。ユーザーは、Dreamweaver サイトを定義する際に、デフォルトアセットフォルダを変更できます。この項の "useDefault" の説明を参照してください。
- "referenceType" は、拡張機能の作成者がファイルリファレンスを head に挿入する場合に必要なフィールドです。"referenceType" の有効な値は、次のとおりです。
  - "link" を指定すると、外部 CSS ファイルへの LINK タグが挿入されます。
  - "import" を指定すると、@import を指定した STYLE タグが挿入されます。
  - "javascript" を指定すると、type=text/javascript を指定した SCRIPT タグが挿入されます。
  - "vbscript" を指定すると、type=text/vbscript を指定した SCRIPT タグが挿入されます。
  - "" を指定すると、head セクションに参照は挿入されません。
- "useDefault" は、"*destURL*" に設定された値の解釈方法を示すブール値です。デフォルトは、true です。true の場合、"destURL" は、サイトの Spry アセットフォルダの相対パスとして処理されます。false の場合、"*destURL*" はサイトルートの相対パスになります。
- "useRelativeSrc" は、"*destURL*" に設定された値の解釈方法を示すブール値です。デフォルト値は false です。false の場合、"*referenceType*" を指定すると、"*srcURL*" は絶対パスとして挿入されます。

戻り値

なし

例

```
function objectTag()
{
.
.
.
var dom = dw.getDocumentDOM();
var assetList = new Array();
var assetInfo = new AssetInfo("Objects/Ajax/Accordion.css", "Objects/Ajax/Accordion.css",
    "Accordion.css", "link");
assetList.push(assetInfo);
assetInfo = new AssetInfo("Objects/Ajax/Accordion.js", "Accordion.js", "javascript");
assetList.push(assetInfo);
assetInfo = new AssetInfo("Objects/Ajax/Images", "Images", "");
assetList.push(assetInfo);
dom.copyAssets(assetList);
return retVal;
}
```

dom.getDefaultAssetFolder()

対応バージョン
Dreamweaver CS3

説明
ドキュメントのデフォルトアセットフォルダを取得します。

引数
なし

戻り値
デフォルトアセットフォルダ名を指定するストリング。

例
```
function objectTag()
{
.
.
.
var defaultAssetFolder = dom.getDefaultAssetFolder();
.
.
.
return retVal;
}
```

ブラウザ互換性チェック関数

以下の関数を使用すると、ブラウザレンダリングバグを引き起こす可能性のある HTML と CSS の組み合わせを簡単に検索することができます。詳細については、『Dreamweaver 拡張ガイド』の「ブラウザの互換性の問題をチェックする API」を参照してください。コマンドなど、他のタイプの拡張機能で使用することもできます。

注意 : これらの関数で返される値は、デザインビューで現在有効になっているスタイルを表します。ブラウザ互換性チェックの一環として、これらの関数を問題のあるファイルで使用すると、ファイルがターゲットブラウザでどのように解釈されるかに基づいて、スタイルが自動的にフィルタ処理されます。たとえば、*Star HTML* を使用して定義したスタイルは、ターゲットブラウザが *Internet Explorer 6.0* 以前の場合、考慮対象となります。ただし、ブラウザ互換性チェックのスコープ外でこれらの関数を使用する場合、フィルタ処理は実行されません。

elem.getComputedStyleProp()

対応バージョン
Dreamweaver CS3

説明
指定したエレメントのレンダリングに使用される、指定した CSS プロパティの値を取得します。カスケードでのプロパティの指定順序は無視されます。ブラウザとは異なり、値に “px” を指定していない場合でも、長さはピクセル単位でレポートされます。

引数

propName、*pseudoElt*

- *propName* - CSS プロパティの名前。ハイフンは使用せず、インターキャプス形式で記述してください。たとえば、"font-size" は "fontSize" と指定します。
- *pseudoElt* - CSS 擬似エレメント。擬似エレメントがない場合は null を指定します。

戻り値

プロパティの計算値を格納したストリング。

注意 : 数値もストリングとして返されます。これらの値を計算で使用するには、*parseInt()* または *parseFloat()* で値を数値に変換します。

例

```
var dom = dw.getDocumentDOM();
var myDiv = dom.getElementsByTagName('myDiv');
var float = myDiv.getComputedStyleProp("float");
if (float == "left")
    alert("This div is floated left.");
```

window.getDeclaredStyle()

対応バージョン

Dreamweaver CS3

説明

指定したエレメントに対して宣言されている CSS スタイルを取得します。getComputedStyle() 関数では、明示的に宣言されていないスタイルは場合によって結果が異なりますが、getDeclaredStyle() 関数は、ピクセル単位の計算値ではなく、20% や .8em など、スタイルシートに宣言された実際の長さの値を返します。bGetInherited が false (デフォルト) の場合は、エレメントに直接適用されるスタイルのみが取得され、親から継承されたスタイルは取得されません。

引数

elt、*pseudoElt*、*psuedoClassList*、*bGetInherited*

- *elt* - スタイル情報が必要なドキュメント内のノード。
- *pseudoElt* - CSS 擬似エレメント。擬似エレメントがない場合は null を指定します。
- *psuedoClassList* - スペースで区切られた疑似クラスのリストで構成される、オプションのストリング。
- *bGetInherited* - 祖先から継承したスタイルを取得するかどうかを示すオプションのブール値。デフォルトは false です。

戻り値

名前でアクセスできるスタイルプロパティを含む読み取り専用オブジェクト。

例

```
var dom = dw.getDocumentDOM();
var myDiv = dom.getElementById('myDiv');
var props = window.getDeclaredStyle(myDiv);
var marleft = "";
var units = "";
if (typeof(props.marginLeft) != "undefined"){
    marleft = props.marginLeft;
    units = marleft.replace(/\d+/,""); // 数字を削除し、単位を残します。
    alert(units); // %、px、pt、em などを表示します。
}
else
alert("no margin-left property has been set for myDiv.");
```

dom.getMinDisplayWidth()

対応バージョン

Dreamweaver CS3

説明

ブロックレベルのコンテナの内容をすべてを表示するために必要な最小幅を取得します。

注意 : *dom.minDisplayWidth()* 関数が返す値より小さい値が CSS を使用して指定されている場合、コンテナの実際の幅が内容に比べて小さい可能性があります。

引数

container

- *container* は、必要な最小幅を取得するコンテナエレメントです。

戻り値

指定したコンテナの最小表示幅をピクセル単位で表す整数。または、エレメントがコンテナでない場合や最小幅を決定できない場合は -1。

例

```
var dom = dw.getDocumentDOM();
var myDiv = dom.getElementById('myDiv');
var props = window.getComputedStyle(myDiv);
var minW = dom.getMinDisplayWidth(myDiv);
var setW = props.width;

if (minW > setW)
alert("Depending on the browser, your content will either be \n" +
       "clipped, or the container will expand beyond its set width.");
```

dom.getBlockElements()
elem.getBlockElements()

対応バージョン

Dreamweaver CS3

説明

固有のまたは指定された表示値として 'block' を持つ子孫を、ドキュメント (またはエレメント) から検索します。

引数

なし

戻り値

エレメントノードの配列。

例

```
[...]
var blocks = DOM.getBlockElements();
var dProps = null, children = null;
for (var i=0; i < blocks.length; i++){
    // 宣言されているスタイルを取得します。幅または高さが
    // 明示的に指定されているかどうかを確認できます。
    dProps = window.getDeclaredStyle(blocks[i]);
    // ブロックに子があり、border-left、padding-bottom は含まれているが、
    // 幅も高さも定義されていない場合
    if (blocks[i].hasChildNodes() && |
        issueUtils.hasBorder(blocks[i],null,"left") &&
        (parseFloat(blocks[i].getComputedStyleProp("padding-bottom")) > 0) &&
        typeof(dProps.width) == "undefined" && typeof(dProps.height) == "undefined"){
        children = blocks[i].getBlockElements();
        var hasLayout = false;
        // ブロックレベルの子をループ処理して、幅または高さが
        // 定義されているかどうかをチェックします。外側のブロックの子に
        // 幅または高さが定義されていれば、バグは発生しません。

        for (var j=0; j < children.length; j++){
            dProps = window.getDeclaredStyle(children[j]);
            if (typeof(dProps.width) != "undefined" || typeof(dProps.height) !=
            "undefined"){
            hasLayout = true;
                break;
            }
        }
    [...]
    }
}
[...]
```

dom.getInlineElements()
elem.getInlineElements()

対応バージョン

Dreamweaver CS3

説明

固有のまたは指定された表示値として `'inline'` を持つ子孫を、ドキュメント (またはエレメント) から検索します。

引数

なし

戻り値

エレメントノードの配列。

例

```
[...]
var DOM = dw.getDocumentDOM();
var inEls = DOM.body.getInlineElements();
var next = null, prev = null, parent = null;
var props = null;

// すべてのインラインエレメントで置換エレメントを検索します。
// 置換エレメントが検出されない場合、処理は行いません。
for (var i=0; i < inEls.length; i++){
    if (inEls[i].tagName == 'IMG' ||
        inEls[i].tagName == 'INPUT' ||
        inEls[i].tagName == 'TEXTAREA' ||
        inEls[i].tagName == 'SELECT' ||
        inEls[i].tagName == 'OBJECT'){
        // 何らかの処理
    }
}
[...]
```

dom.getHeaderElements()
elem.getHeaderElements()

対応バージョン

Dreamweaver CS3

説明

H1 ～ H6 のヘッダータグをドキュメント (またはエレメント) から検索します。

引数

なし

戻り値

エレメントノードの配列。

例

```
var DOM = dw.getDocumentDOM();
var headers = DOM.getHeaderElements();

for (var i=0; i < headers.length; i++){
    alert(headers[i].tagName);
}
```

dom.getListElements()
elem.getListElements()

対応バージョン

Dreamweaver CS3

説明

番号リスト、リスト、および定義リストをドキュメント (またはエレメント) から検索します。

引数
なし

戻り値
エレメントノードの配列。

例
```
[...]
var DOM = dw.getDocumentDOM();
// ドキュメント内のすべての UL、OL、DL の各要素を取得します。
var lists = DOM.getListElements();
var props = null;
for (var i=0; i < lists.length; i++){
    props = window.getDeclaredStyle(lists[i]);
    if ((props.cssFloat == "left" || props.cssFloat == "right") && props.overflow == "auto"){
        // 何らかの処理
    }
}
[...]
```

elem.isBlockElement()

対応バージョン
Dreamweaver CS3

説明
エレメントが、固有のまたは指定された表示値として 'block' を持つかどうかをチェックします。

引数
なし

戻り値
オブジェクトがブロックレベルのエレメントかどうかを示すブール値。

例
```
[...]
    var DOM = dw.getDocumentDOM();
    var blocks = DOM.body.getBlockElements();
    var next = null;
    for (var i=0; i < blocks.length; i++){
        // next は blocks[i] の直後のノード
        next = blocks[i].nextSibling;
        // next が null ではなく、エレメントノードで、ブロックエレメントの場合、
        // "2 つの連続したブロックエレメントの 2 番目 " と判断できます。
        if (next && (next.nodeType == 1) && next.isBlockElement()){
        // 何らかの処理
    }
}
[...]
```

elem.isInlineElement()

対応バージョン
Dreamweaver CS3

説明

エレメントが、固有のまたは指定された表示値として 'inline' を持つかどうかをチェックします。

引数

なし

戻り値

オブジェクトがインラインエレメントかどうかを示すブール値。

例

```
[...]
var DOM = dw.getDocumentDOM();
var floats = issueUtils.getFloats(DOM.body);
var next = null;
for (var i=0; i < floats.length; i++){
    next = floats[i].nextSibling;
    // フロートの nextSibling がテキストノードまたはインラインエレメントの場合
    if (next && (next.nodeType == Node.TEXT_NODE ||
        (next.nodeType == Node.ELEMENT_NODE && next.isInlineElement()))){
    // 何らかの処理
    }
}
[...]
```

elem.isHeaderElement()

対応バージョン

Dreamweaver CS3

説明

エレメントが、h1、h2、h3、h4、h5、h6 のいずれかのタグかどうかを確認します。

引数

なし

戻り値

オブジェクトがヘッダーエレメントかどうかを示すブール値。

例

```
[...]
var DOM = dw.getDocumentDOM();
var floats = issueUtils.getFloats(DOM.body);
var prev = null;
// ドキュメントの先頭のフロートは影響を受けません。
// そのため 1 から開始します。
for (var i=1; i < floats.length; i++){
    prev = floats[i].previousSibling;
    // フロートの前のエレメントがヘッダーの場合
    if (prev && prev.isHeaderElement()){
    // 何らかの処理
    }
}
[...]
```

elem.isListElement()

対応バージョン

Dreamweaver CS3

説明

エレメントが ul、ol、dl のいずれかのタグかどうかを確認します。

引数

なし

戻り値

オブジェクトがリストエレメントかどうかを示すブール値。

例

```
[...]
var DOM = dw.getDocumentDOM();
var floats = issueUtils.getFloats(DOM.body);
var prev = null, children = null;
for (var i=0; i < floats.length; i++){
    children = floats[i].childNodes;

    for (var k=0; k < children.length; k++){
                    if (children[k].isListElement()){
                // 何らかの処理
        }
    }
}
[...]
```

第 16 章 : 動的ドキュメント

Adobe® Dreamweaver® CS3 の動的ドキュメント関数は、Web サーバーページに関連する操作を行います。動的ドキュメント関数が行う操作には、[コンポーネント] パネルで選択されているノードのプロパティの取得、ユーザーのドキュメントに含まれるデータソースすべてのリストの取得、デザインビューでの動的コンテンツの表示、ドキュメントへのサーバービヘイビアの適用、現在定義されているサーバーモデルすべての名前の取得などがあります。

本章では関数をグループ分けして、それぞれを以下の項で説明します。

サーバーコンポーネント関数

サーバーコンポーネント関数を使用すると、[コンポーネント] パネルに表示されるサーバーコンポーネントツリーコントロールの、現在選択されているノードにアクセスできます。これらの関数を使用して、コンポーネントツリーの表示を更新することもできます。

dreamweaver.serverComponents.getSelectedNode()

対応バージョン
Dreamweaver MX

説明
サーバーコンポーネントツリーコントロールで現在選択されている `ComponentRec` プロパティを返します。

引数
なし

戻り値
`ComponentRec` プロパティ。

dreamweaver.serverComponents.refresh()

対応バージョン
Dreamweaver MX

説明
コンポーネントツリーの表示を更新します。

引数
なし

戻り値
なし

データソース関数

データソースファイルは、"Configuration/DataSources" フォルダに格納されています。各サーバーモデルには、ASP.Net/C#、ASP.Net/VisualBasic、ASP/JavaScript、ASP/VBScript、ColdFusion、JSP、および PHP/MySQL といった独自のフォルダが用意されています。各サーバーモデルのサブフォルダには、個々のサーバーモデルのデータソースに関連付けられている HTML および EDML ファイルが格納されています。

Dreamweaver におけるデータソースの使用の詳細については、『Dreamweaver 拡張ガイド』の「データソース」を参照してください。

dreamweaver.dbi.getDataSources

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

"Configuration/DataSources" フォルダの各ファイルに対して `findDynamicSources()` 関数を呼び出します。この関数を使用して、ユーザーのドキュメント内のすべてのデータソースのリストを生成できます。この関数は、"Configuration/DataSources" フォルダ内のすべてのファイルに対して、反復処理を行います。各ファイルの `findDynamicSources()` 関数を呼び出し、返されるすべての配列を連結して、データソースの連結された配列を返します。

引数

なし

戻り値

ユーザーのドキュメントにあるすべてのデータソースの連結されたリストを含む配列。配列の各要素はオブジェクトです。各オブジェクトには以下のプロパティがあります。

- `title` プロパティには、各親ノードのアイコンの右に表示されるラベルストリングを指定します。`title` プロパティの定義は必須です。
- `imageFile` プロパティには、[動的データ] ダイアログボックス、[動的テキスト] ダイアログボックス、または [バインディング] パネルで親ノードを表すアイコン (GIF イメージ) を含むファイルのパスを指定します。`imageFile` プロパティの定義は必須です。
- `allowDelete` プロパティはオプションです。このプロパティが `false` 値に設定されている場合、ユーザーが [バインディング] パネルでこのノードをクリックすると、[-] ボタンが使用不可になります。このプロパティを `true` 値に設定すると、[-] ボタンは使用可能になります。このプロパティを定義しない場合、[-] ボタンは、プロパティが `true` 値に設定された場合と同様に、ユーザーが項目をクリックすると使用可能になります。
- `dataSource` プロパティには、`findDynamicSources()` 関数が定義されているファイルの簡単な名前を指定します。たとえば、"Configuration/DataSources/ASP_Js" フォルダの "Session.htm" ファイルに定義されている `findDynamicSources()` 関数は、`dataSource` プロパティを `session.htm` に設定します。このプロパティの定義は必須です。
- `name` プロパティには、データソースの `dataSource` が存在する場合は、そのデータソースに関連付けられているサーバービヘイビアの名前を指定します。`name` プロパティの定義は必須ですが、サーバービヘイビアがセッション変数などのデータソースに関連付けられていない場合は、空白のストリング (`""`) を指定することもできます。

dw.dbi.setExpanded()

対応バージョン

Dreamweaver CS3

説明

ノードを展開または縮小に設定します。

引数

data-source-node-name、*expanded*

- *data-source-node-name* には、展開または縮小するデータソース名を表すストリングを設定します。
- *expanded* には、データソースノードを展開するか、縮小するかを示すブール値を指定します。

戻り値

なし

例

```
dw.dbi.setExpanded(dsName, true);          // データソースノードを展開します。
```

Extension Data Manager 関数

この項で説明する API は、EDM (Extension Data Manager) の構成要素です。これらの関数を呼び出して、グループファイルや構成要素ファイルに含まれるデータにプログラムからアクセスし、操作することができます。EDM には、次の機能があります。

- グループファイルと構成要素ファイルに対して、すべての EDML ファイルに対する入出力を実行します。
- EDM はサーバーモデルフィルタとして機能し、現在のサーバーモデルに対するすべてのデータ要求を実行します。

dreamweaver.getExtDataValue()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

この関数は、指定されたノードに、EDML ファイルからフィールド値を取得します。

引数

qualifier(s)

- *qualifier(s)* 引数は、カンマで区切ったノード修飾子のリストです。長さは必要な情報のレベルによって異なり、グループ名または構成要素名、サブブロック (存在する場合)、およびフィールド名が含まれます。

戻り値

フィールド値。値が指定されていない場合、Dreamweaver はデフォルト値を使用します。

例

次の例では、構成要素 recordset_main の insertText タグに対する location 属性値を取得しています。

```
dw.getExtDataValue("recordset_main", "insertText", "location");
```

dreamweaver.getExtDataArray()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

この関数は、指定されたノードに、EDML ファイルから値の配列を取得します。

引数

qualifier(s)

- *qualifier(s)* 引数は、カンマで区切ったノード修飾子の可変長リストで、グループ名または構成要素名、サブブロック (存在する場合)、およびフィールド名が含まれます。

戻り値

子ノード名の配列。

dreamweaver.getExtParticipants()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

この関数は、EDML グループファイルまたは構成要素ファイルから、構成要素のリストを取得します。

引数

value、*qualifier(s)*

- *value* 引数はプロパティ値です。無視する場合は空白にします。たとえば、`dreamweaver.getExtParticipants("", "participant");` のように指定します。
- *qualifier(s)* 引数は、必要なプロパティのノード修飾子をカンマで区切った可変長リストです。

戻り値

プロパティが指定された場合はそのプロパティを持ち、さらに値が指定された場合はプロパティがその値と一致する構成要素名の配列。

dreamweaver.getExtGroups()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

EDML グループファイルから、サーバービヘイビアの名前に相当するグループ名を取得します。

引数

value、*qualifier(s)*

- *value* 引数はプロパティ値です。無視する場合は空白にします。
- *qualifier(s)* 引数は、必要なプロパティのノード修飾子をカンマで区切った可変長リストです。

戻り値

プロパティが指定された場合はそのプロパティを持ち、さらに値が指定された場合はプロパティがその値と一致するグループ名の配列。

dreamweaver.refreshExtData()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

すべての拡張データファイルをリロードします。


サーバービヘイビア *EDML* ファイルに対する編集が、*Dreamweaver* を再起動せずにリロードされるようにする便利なコマンドを、この関数から作成することができます。

引数

なし

戻り値

リロードされたデータ。

ライブデータ関数

以下のライブデータ関数を使用すると、メニュー機能と同様の操作を実行できます。

- `showLiveDataDialog()` 関数は、[表示]-[ライブデータの設定] メニュー項目を選択する場合に使用します。
- `setLiveDataMode()` 関数は、[表示]-[ライブデータ] および [表示]-[ライブデータの更新] メニュー項目を選択する場合に使用します。
- `getLiveDataMode()` 関数は、ライブデータモードがアクティブかどうかを判別します。

上記以外のライブデータ関数は、トランスレータ API の `liveDataTranslateMarkup()` 関数を実装する場合に使用できます。

dreamweaver.getLiveDataInitTags()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

現在アクティブなドキュメントの初期化タグを返します。初期化タグは、ユーザーが [ライブデータの設定] ダイアログボックスに入力する HTML タグです。この関数は通常、トランスレータの `liveDataTranslateMarkup()` 関数から呼び出されるので、トランスレータは初期化タグを `liveDataTranslate()` 関数に渡すことができます。

引数

なし

戻り値

初期化タグを含むストリング。

dreamweaver.getLiveDataMode()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

[ライブデータ] ウィンドウが現在表示されているかどうかを判別します。

引数

なし

戻り値

[ライブデータ] ウィンドウが表示されている場合は `true`、それ以外の場合は false を示すブール値。

dreamweaver.getLiveDataParameters ()

対応バージョン

Dreamweaver MX

説明

ライブデータの設定として指定される URL パラメータを取得します。

ライブデータモードを使用すると、アプリケーションサーバーでトランスレートされて返されたものと同じ状態の Web ページを、デザインの段階で表示することができます。デザインビューに表示する動的コンテンツを生成すると、ライブデータと共にページレイアウトを表示して、必要に応じて調整することができます。

ライブデータを表示するには、ドキュメントで参照しているすべての URL パラメータについて、ライブデータ設定を入力しておく必要があります。これにより、シミュレーションで定義されていないパラメータに関するエラーが、Web サーバーから返されるのを防ぐことができます。

URL パラメータは、名前と値のペアとして入力します。たとえば、ドキュメントで、サーバースクリプトに含まれる URL 変数の `ID` および `Name` を参照する場合は、ライブデータを表示する前に、これらの URL パラメータを設定する必要があります。

Dreamweaver では、以下の方法でライブデータの設定を行うことができます。

- [表示] メニューからアクティブにできる [ライブデータの設定] ダイアログボックスを使用します。
- ツールバーの [ライブデータの表示] ボタンをクリックしたときに、ドキュメントの上部に表示される [URL] テキストフィールドを使用します。

`ID` および `Name` パラメータには、以下のペアを入力できます。

```
ID          22
Name         Samuel
```

[URL] に、次の例に示すようなパラメータが表示されます。

```
http://someURL?ID=22&Name=Samuel
```

この関数を使用すると、JavaScript でライブデータの設定を取得できます。

引数

なし

戻り値

現在のドキュメントの URL パラメータを含む配列。この配列には、偶数のパラメータストリングが含まれます。それぞれ 2 つのエレメントが、名前と値のペアから成る 1 つの URL パラメータを形成します。偶数のエレメントがパラメータ名で、奇数のエレメントが値です。たとえば、前の例では、`getLiveDataParameters()` を使用すると、`['ID,'22','Name','Samuel']` という `ID` および `Name` パラメータの配列が返されます。

例

次の例は、ライブデータの設定として指定されたパラメータを取得し、`paramsArray` に格納します。

```
var paramsArray = dreamweaver.getLiveDataParameters();
```

dreamweaver.liveDataTranslate()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

HTML ドキュメント全体をアプリケーションサーバーに送信し、サーバーにそのドキュメント内でスクリプトを実行するかどうかを問い合わせて、結果としての HTML ドキュメントを返します。この関数は、トランスレータの `liveDataTranslateMarkup()` 関数からのみ呼び出すことができます。その他の場合にこの関数を呼び出そうとすると、エラーになります。`dreamweaver.liveDataTranslate()` 関数は、以下の操作を実行します。

- アニメーション化したイメージを再生します。このイメージは、[ライブデータ] ウィンドウの右端近くに表示されます。
- ユーザーからの入力を待ちます。[停止] アイコンがクリックされたら、この関数はすぐに値を返します。
- 呼び出し元から単一ストリング引数を受け取ります。このストリングは通常、ユーザーのドキュメントのソースコード全体になります。これは、次の操作で使用するものと同じストリングです。
- ユーザーのドキュメントから取得した HTML ストリングを、ライブデータサーバー上に一時ファイルとして保存します。
- [ライブデータの設定] ダイアログボックスに指定したパラメータを使用して、HTTP 要求をライブデータサーバーに送信します。
- ライブデータサーバーから HTML 応答を受け取ります。
- ライブデータサーバーから一時ファイルを削除します。
- アニメーション化したイメージの再生を停止します。
- 呼び出し元に HTML 応答を返します。

引数

`string`

- 単一のストリング。これは通常、ユーザーの現在のドキュメントのソースコード全体になります。

戻り値

`httpReply` オブジェクト。このオブジェクトは、`MMHttp.getText()` 関数が返す値と同じです。ユーザーが [停止] アイコンをクリックした場合は、戻り値の `httpReply.statusCode` が 200 (OK の状態) になり、戻り値の `httpReply.data` は空白のストリングになります。`httpReply` オブジェクトの詳細については、13 ページの HTTP API を参照してください。

dreamweaver.setLiveDataError()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

トランスレータで `liveDataTranslateMarkup()` 関数の実行中にエラーが発生した場合に表示するエラーメッセージを指定します。Dreamweaver が `liveDataTranslate()` に渡したドキュメントにエラーが含まれる場合、サーバーは HTML でフォーマットされたエラーメッセージを返します。トランスレータ (`liveDataTranslate()` を呼び出したコード) は、サーバーからエラーメッセージが返されたと判断すると、`setLiveDataError()` を呼び出して、Dreamweaver にエラーメッセージを表示します。このメッセージは、`liveDataTranslateMarkup()` 関数の実行が完了した後に表示されます。エラーのダイアログボックスに説明が表示されます。`setLiveDataError()` 関数は、`liveDataTranslateMarkup()` 関数からのみ呼び出すことができます。

引数

`source`

- *`source`* 引数には、ソースコードを含むストリングを指定します。このソースコードは、エラーのダイアログボックスで解析およびレンダリングされます。

戻り値

なし

dreamweaver.setLiveDataMode()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

[ライブデータ] ウィンドウの表示 / 非表示を切り替えます。

引数

bIsVisible

- *bIsVisible* 引数には、[ライブデータ] ウィンドウを表示するかどうかを示すブール値を指定します。この関数に true を渡したときに、Dreamweaver に [ライブデータ] ウィンドウが表示されている場合は、[更新] ボタンをクリックしたときと同じ結果になります。

戻り値

なし

dreamweaver.setLiveDataParameters ()

対応バージョン

Dreamweaver MX

説明

ライブデータモードで使用できるように、ドキュメントで参照する URL パラメータを設定します。

ライブデータモードを使用すると、アプリケーションサーバーでトランスレートされて返されたものと同じ状態の Web ページを、デザインの段階で表示することができます。デザインビューに表示する動的コンテンツを生成すると、ライブデータと共にページレイアウトを表示して、必要に応じて調整することができます。

ライブデータを表示するには、ドキュメントで参照しているすべての URL パラメータについて、ライブデータ設定を入力しておく必要があります。これにより、シミュレーションで定義されていないパラメータに関するエラーが、Web サーバーから返されるのを防ぐことができます。

URL パラメータは、名前と値のペアとして入力します。たとえば、ドキュメントで、サーバースクリプトに含まれる URL 変数の ID および Name を参照する場合は、ライブデータを表示する前に、これらの URL パラメータを設定する必要があります。

この関数を使用すると、JavaScript でライブデータ値を設定できます。

引数

liveDataString

- *liveDataString* 引数には、設定する URL パラメータ (名前と値のペア) を含むストリングを指定します。

戻り値

なし

例

```
dreamweaver.setLiveDataParameters("ID=22&Name=Samuel")
```

dreamweaver.showLiveDataDialog()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

[ライブデータの設定] ダイアログボックスを表示します。

引数

なし

戻り値

なし

サーバービヘイビア関数

サーバービヘイビア関数を使用すると、[サーバービヘイビア] パネルを操作できます。このパネルは、[ウィンドウ]-[サーバービヘイビア] を選択すると表示されます。以下の関数を使用すると、ページ上のすべてのサーバービヘイビアを検索したり、新規ビヘイビアをドキュメントにプログラム上で追加することができます。また、既存のビヘイビアを修正することも可能です。

注意 : *dw.serverBehaviorInspector* を *dw.sbi* に省略できます。

dreamweaver.getParticipants()

対応バージョン

Dreamweaver UltraDev 4 (英語版)

説明

JavaScript 関数の `dreamweaver.getParticipants()` では、ユーザーのドキュメントから構成要素のリストが取得されます。Dreamweaver によって、すべてのビヘイビアの構成要素が検出された後、そのリストが保存されます。通常、この関数を `findServerBehaviors()` 関数 (詳細については、『Dreamweaver 拡張ガイド』の「サーバービヘイビア」を参照) と共に使用して、ユーザーのドキュメントからビヘイビアのインスタンスを探します。

引数

edmlFilename

- *edmlFilename* 引数は、ユーザーのドキュメントで検索する構成要素の名前を含む、グループファイルまたは構成要素ファイルの名前です。このストリングには、.edml 拡張子を付けずに、ファイル名を指定します。

戻り値

この関数では、指定された構成要素のインスタンス (グループファイルの場合は、グループに含まれる構成要素のインスタンス) のうち、ユーザーのドキュメントに検出されたものすべてが含まれる列が返されます。配列には、JavaScript オブジェクトが含まれており、ユーザーのドキュメントで検出された各構成要素のインスタンスごとに 1 つずつ配列の要素が対応しています。配列はドキュメントに検出された構成要素の順にソートされます。各 JavaScript オブジェクトには、以下のプロパティがあります。

- *participantNode* 引数は、ユーザーのドキュメント内の構成要素ノードへのポインタです。
- *participantName* プロパティは、.edml 拡張子を付けない、構成要素の EDML ファイルの名前です。
- *parameters* プロパティは、すべてのパラメータと値のペアを格納している JavaScript オブジェクトです。
- *matchRangeMin* プロパティは、ドキュメントの構成要素ノードから構成要素コンテンツの先頭までの文字のオフセットを定義します。
- *matchRangeMax* プロパティは、構成要素ノードの先頭から構成要素コンテンツの最後の文字までのオフセットを定義する構成要素を表す整数です。

dreamweaver.serverBehaviorInspector.getServerBehaviors()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

ページ上のすべてのビヘイビアのリストを取得します。Dreamweaver により、サーバービヘイビアの内部リストが古くなっている可能性があると判断された場合は、現在インストールされている各ビヘイビアに対して、`findServerBehaviors()` 関数を呼び出します。各関数は 1 つずつ配列を返します。Dreamweaver はすべての配列を 1 つの配列にマージし、ドキュメントに表示される各ビヘイビアの `selectedNode` オブジェクトの順序に基づいて、マージした配列をソートします。マージした配列は内部保存されます。`getServerBehaviors()` 関数は、マージした配列にポインタを返します。

引数

なし

戻り値

JavaScript オブジェクトの配列。`findServerBehaviors()` の呼び出しにより、配列内のオブジェクトが返されます。これらのオブジェクトは、[サーバービヘイビア] パネルに表示される順序でソートされます。

dreamweaver.popupServerBehavior()

対応バージョン

Dreamweaver UltraDev 1 (英語版)

説明

新規サーバービヘイビアをドキュメントに適用するか、既存のビヘイビアを変更します。ユーザーがビヘイビアにパラメータを指定する必要がある場合、ダイアログボックスが表示されます。

引数

{behaviorName または behaviorObject}

- *behaviorName* 引数 (オプション) には、ビヘイビアの名前、ファイルの title タグまたはファイル名を表すストリングを指定します。
- *behaviorObject* 引数 (オプション) には、ビヘイビアオブジェクトを指定します。

この引数を省略した場合は、現在選択されているサーバービヘイビアが実行されます。この引数にサーバービヘイビアの名前を指定した場合は、このビヘイビアがページに追加されます。この引数に `getServerBehaviors()` 関数が返す配列内のオブジェクトの 1 つを指定した場合は、ダイアログボックスが表示されるので、ユーザーはビヘイビアのパラメータを修正することができます。

戻り値

なし

サーバーモデル関数

Dreamweaver では、各ドキュメントにドキュメントタイプが関連付けられています。動的なドキュメントタイプには、ASP-JS、ColdFusion、PHP-MySQL などのサーバーモデルも関連付けられています。

サーバーモデルは、サーバーテクノロジー特有の機能を分類する場合に使用されます。ドキュメントに関連付けられたサーバーモデルに基づいて、さまざまなサーバービヘイビア、データソースなどが表示されます。

サーバーモデル関数を使用すると、現在定義されている一連のサーバーモデルを確認することができます。たとえば、現在のサーバーモデルの名前、言語、バージョンを確認したり、現在のサーバーモデルに UTF-8 などの特定の文字セットがサポートされているかどうかを確認したりできます。

注意 : *Dreamweaver* では、サーバーモデルを最初にロードするときに、そのモデルの *HTML* ファイル内の情報がすべて読み取られ、保存されます。そのため、拡張機能によって *dom.serverModel.getServerName()*、*dom.serverModel.getServerLanguage()*、*dom.serverModel.getServerVersion()* などの関数が呼び出されると、これらの関数から保存された値が返されます。

dom.serverModel.getAppURLPrefix()

対応バージョン

Dreamweaver MX

説明

テストサーバー上のサイトのルートフォルダの URL を返します。この URL は、[サイト定義] ダイアログボックスの [詳細設定] タブにある [テストサーバー] に指定されているものと同じです。

Dreamweaver は、テストサーバーとの通信に、HTTP を使用します (ブラウザと同じ方法)。その際、この URL を使用して、サイトのルートフォルダにアクセスします。

引数

なし

戻り値

ライブデータおよびデバッグの目的に使用されるアプリケーションサーバーの URL を含むストリング。

例

ユーザーがサイトを作成し、テストサーバーをローカルコンピュータに設定して、ルートフォルダに `"employeeapp"` を指定した場合、`dom.serverModel.getAppURLPrefix()` 関数を呼び出すと、次のストリングが返されます。

```
http://localhost/employeeapp/
```

dom.serverModel.getDelimiters()

対応バージョン

Dreamweaver MX

説明

JavaScript コードで各サーバーモデルのスクリプト区切りを取得できるようにします。これにより、サーバーモデルコードの管理を、ユーザーが記述したスクリプトコードの管理と切り離すことができます。

引数

なし

戻り値

オブジェクトの配列。各オブジェクトには、以下の 3 つのプロパティが含まれます。

- *startPattern* プロパティは、開始スクリプト区切りと一致する正規表現です。
- *endPattern* プロパティは、終了スクリプト区切りと一致する正規表現です。
- *participateInMerge* パターンは、リストされている区切りで囲まれたコンテンツが、ブロックのマージに含まれる (`true`) か、含まれない (`false`) かを指定するブール値です。

dom.serverModel.getDisplayName()

対応バージョン
Dreamweaver MX

説明
ユーザーインターフェイス (UI) に表示されるサーバーモデルの名前を取得します。

引数
なし

戻り値
サーバーモデルの名前を表すストリング。

dom.serverModel.getFolderName()

対応バージョン
Dreamweaver MX

説明
"Configuration" フォルダにあるこのサーバーモデルに使用されているフォルダの名前 ("ServerModels" サブフォルダなど) を取得します。

引数
なし

戻り値
フォルダ名を表すストリング。

dom.serverModel.getServerExtension() (非推奨)

対応バージョン
Dreamweaver UltraDeveloper 4 (Dreamweaver MX では非推奨)

説明
現在のサーバーモデルを使用するファイルのデフォルトのファイル拡張子を返します。デフォルトのファイル拡張子は、リストの先頭に入ります。ユーザードキュメントが現在選択されていない場合は、`serverModel` オブジェクトが現在選択されているサイトのサーバーモデルに設定されます。

引数
なし

戻り値
サポートされているファイル拡張子を表すストリング。

dom.serverModel.getServerIncludeUrlPatterns()

対応バージョン
Dreamweaver MX

説明

後に示すプロパティのリストを返します。これらのプロパティを使用すると、次の情報にアクセスできます。

- トランスレータ URL パターン
- ファイルリファレンス
- タイプ

引数

なし

戻り値

オブジェクトのリスト。searchPattern ごとに 1 つずつ返されます。各オブジェクトには、以下の 3 つのプロパティが含まれます。

プロパティ	説明
pattern	EDML ファイルの searchPattern フィールドに指定されている JavaScript の正規表現。正規表現は、2 つのスラッシュ (//) で区切られます。
fileRef	インクルードファイルリファレンスに対応する正規表現のサブマッチの 1 から始まるインデックス。
type	paramName からサフィックス _includeUrl を削除した後に残される部分。この type は、<MM:BeginLock> タグのタイプ属性に割り当てられます。"Configuration¥Translators" フォルダの "Server Model SSI.htm" の例を参照してください。

例

構成要素ファイルに含まれる以下のコードスニペットは、トランスレータ searchPatterns タグを示しています。

```
<searchPatterns whereToSearch="comment">
    <searchPattern paramNames=",ssi_comment_includeUrl">
        <![CDATA[/<!--\s*#include\s+(file|virtual)\s*=\s*"([^"]*)"\s*-->/i]]>
    </searchPattern>
</searchPatterns>
```

検索パターンには、2 つのサブマッチ (いずれもカッコに囲まれています) を指定する JavaScript の正規表現が含まれています。最初のサブマッチはテキストストリングの file または virtual を検索します。2 番目のサブマッチはファイルリファレンスです。

トランスレータ URL パターンにアクセスするには、次のようなコードを作成する必要があります。

```
var serverModel = dw.getDocumentDOM().serverModel;
var includeArray = new Array();
includeArray = serverModel.getServerIncludeUrlPatterns();
```

serverModel.getServerIncludeUrlPatterns() を呼び出すと、以下の 3 つのプロパティが返されます。

プロパティ	戻り値
pattern	/<!--\s*#include\s+(file|virtual)\s*=\s*"([^"]*)"\s*-->/i
fileRef	2
type	ssi_comment

dom.serverModel.getServerInfo()

対応バージョン

Dreamweaver MX

説明

現在のサーバーモデル特有の情報を返します。この情報は、"Configuration/ServerModels" フォルダにあるサーバーモデルの HTML 定義ファイルに定義されています。

HTML 定義ファイルは、情報を修正したり、変数値や関数を追加することができます。たとえば、`serverName`、`serverLanguage`、`serverVersion` プロパティなどを修正できます。`dom.serverModel.getServerInfo()` 関数は、サーバーモデルの作成者が定義ファイルに追加した情報を返します。

注意 : デフォルトのサーバーモデルファイルに定義されているその他の値は、内部使用に限定されています。

`serverName`、`serverLanguage`、および `serverVersion` プロパティは、以下の関数を使ってデベロッパーが直接アクセスすることのできる特別な情報です。

- `dom.serverModel.getServerName()`
- `dom.serverModel.getServerLanguage()`
- `dom.serverModel.getServerVersion()`

引数

なし

戻り値

現在のサーバーモデル特有の各種情報を含む JavaScript オブジェクト。

dom.serverModel.getServerLanguage() (非推奨)

対応バージョン

UltraDeveloper 1 (英語版)(Dreamweaver MX では非推奨)

説明

ドキュメントに関連付けられたサーバーモデルを判別し、その値を返します。サイトのサーバー言語は、[サイト定義] ダイアログボックスの [アプリケーションサーバー] タブの [スクリプト言語] 設定で選択された値です。戻り値を取得するために、この関数はサーバーモデル API の `getServerLanguage()` 関数を呼び出します。

注意 : [スクリプト言語] リストは、*Dreamweaver 4* 以前のバージョンでのみ表示されます。*Dreamweaver MX* 以降のバージョンでは、サポートされているスクリプト言語のリストが [サイト定義] ダイアログボックスに表示されません。また、*Dreamweaver MX* 以降のバージョンでは、*`dom.serverModel.getServerLanguage()`* 関数は、サーバーモデル *API* の *`getServerInfo()`* 関数の呼び出しによって返されたオブジェクトの *`serverLanguage`* プロパティを読み取ります。

引数

なし

戻り値

サポートされているスクリプト言語を含むストリング。

dom.serverModel.getServerName()

対応バージョン

Dreamweaver 1、Dreamweaver MX (Dreamweaver MX で機能強化)

説明

ドキュメントに関連付けられたサーバー名を取得し、その値を返します。サーバー名からサーバーテクノロジー (ASP.NET や JSP など) を特定することはできますが、同じサーバーテクノロジーのどの言語であるか (ASP.NET VB や ASP.NET C# など) を特定することはできません。返される値は、`ASP`、`ASP.NET`、`Cold Fusion`、`JSP` または `PHP` です。

ドキュメントに関連付けられたサーバーモデル名の取得方法については、303 ページの dom.serverModel.getDisplayName() または 303 ページの dom.serverModel.getFolderName() を参照してください。

注意 : *Dreamweaver MX* 以降のバージョンでは、*`dom.serverModel.getServerName()`* 関数は、サーバーモデル *API* の *`getServerInfo()`* 関数の呼び出しによって返されたオブジェクトの *`serverName`* プロパティを読み取ります。

引数

なし

戻り値

サーバー名を含むストリング。

dom.serverModel.getServerSupportsCharset()

対応バージョン

Dreamweaver MX

説明

ドキュメントに関連付けられているサーバーモデルが指定の文字セットをサポートしているかどうかを判別します。

注意 : この関数は *JavaScript* レイヤーから呼び出すこともできますが、*Dreamweaver* では、ユーザーが [ページプロパティ] ダイアログボックスでエンコードを変更した場合も、この関数が呼び出されます。サーバーモデルが新しい文字エンコードをサポートしていない場合、この関数は *`false`* を返します。また、変換を行うかどうかを尋ねる警告メッセージが表示されます。このような状況は、ユーザーが *ColdFusion 4.5* のドキュメントを *UTF-8* に変換しようとした場合などに発生します。これは、*ColdFusion* が *UTF-8* エンコードをサポートしていないためです。

引数

`metaCharSetString`

- *`metaCharSetString`* 引数は、特定の文字セットを指定するストリング値です。この値は、ドキュメントに関連付けられている `meta` タグの `"charset="` 属性の値と同じです。指定のサーバーモデルにサポートされている値は、そのサーバーモデルの HTML 定義ファイルに定義されています。このファイルは "Configuration¥ServerModels" フォルダにあります。

戻り値

サーバーモデルが指定の文字セットをサポートしている場合は `true`、それ以外の場合は `false` を示すブール値。

dom.serverModel.getServerVersion()

対応バージョン

UltraDeveloper 1 (英語版)、Dreamweaver MX (Dreamweaver MX で機能拡張)

説明

ドキュメントに関連付けられたサーバーモデルを判別し、その値を返します。各サーバーモデルには、サーバーモデル API の `getVersionArray()` 関数が含まれており、この関数は名前とバージョンのペアのテーブルを返します。

注意 : *Dreamweaver* では、*`dom.serverModel.getServerVersion()`* 関数は、サーバーモデル *API* の *`getServerInfo()`* 関数の呼び出しによって返されたオブジェクトの *`serverVersion`* プロパティを最初に読み取ります。このプロパティが存在しない場合、*`dom.serverModel.getServerVersion()`* は、*`getVersionArray()`* 関数からこれを読み取ります。

引数

name

- *name* 引数には 、サーバーモデルの名前を表すストリングを指定します。

戻り値

指定したサーバーモデルのバージョンを含むストリング。

dom.serverModel.testAppServer()

対応バージョン

Dreamweaver MX

説明

アプリケーションサーバーへの接続が可能かどうかをテストします。

引数

なし

戻り値

アプリケーションサーバーへの接続要求が成功したかどうかを示すブール値。

dreamweaver.getServerModels()

対応バージョン

Dreamweaver MX

説明

現在定義されているすべてのサーバーモデルの名前を取得します。取得される名前は、[サイト定義] ダイアログボックスの [サーバーモデル] テキストフィールドに表示されるものと同じです。

引数

なし

戻り値

ストリングの配列。各ストリングエレメントには、現在定義されているサーバーモデルの名前が含まれます。

第 17 章 : デザイン

Adobe® Dreamweaver® CS3 のデザイン関数は、ドキュメントの外観のデザインに関連する処理を実行します。デザイン関数が実行する処理には、指定した CSS (Cascading Style Sheet) スタイルの適用、選択したフレームの垂直分割または水平分割、選択したレイヤーまたはホットスポットの整列、選択したプラグイン項目の再生、レイアウトセルの作成、テーブルの行や列の操作などがあります。

本章では関数をグループ分けして、それぞれを以下の項で説明します。

CSS レイアウト関数

CSS 関数は、CSS スタイルの適用、解除、作成、および削除を行います。`dreamweaver.cssRuleTracker` オブジェクトのメソッドは、現在のドキュメントではなく、選択インスペクタの [CSS ルールトラッカー] パネルの選択範囲に対して制御または動作を実行します。`dreamweaver.cssStylePalette` オブジェクトのメソッドは、現在のドキュメントではなく、[CSS スタイル] パネルの選択範囲に対して制御または動作を実行します。

dom.applyLayout()

対応バージョン

Dreamweaver CS3

説明

ドキュメントに CSS ベースのレイアウトを適用します。ドキュメントの body が空で、レイアウトを適用できるページである必要があります。具体的には以下のとおりです。

- HTML、XHTML、ColdFusion、PHP などの HTML ベースのページ (CSS、XML、JavaScript などを除く)
- フレームセットまたはテンプレートインスタンスではないページ (テンプレート自体には適用可能)

引数

`layout-index`、*`CSS`*、*`cssFileName`*、*`preventOverwrite`*

- *`layout-index`* には、ゼロから開始する整数のインデックスで、使用するレイアウトを指定します。この引数は、レイアウトのリストへのインデックスです。リストは、対応する関数の `layoutNames` および `layoutDescriptions` を返すために使用されます。
- *`CSS`* には、CSS レイアウトを配置する場所を指定します。指定できる値は、以下のとおりです。
  - `"embed"` : CSS をドキュメントの head に埋め込みます。
  - `"link"` : *`cssFileName`* にリンクします。
  - `"create_and_link"` : CSS を *`cssFileName`* に書き込み、そのファイルにリンクします。

- "import"：*cssFileName* を読み込みます。
- "create_and_import"：CSS を *cssFileName* に書き込み、そのファイルを読み込みます。

- *cssFileName* には、必要に応じて、リンク、読み込み、または作成する CSS ファイルの名前を指定します。
- *preventOverwrite* には、以下のブール値を指定します。
  true：新しい CSS ファイルの作成時にファイルが既に存在する場合、作成は失敗します。
  false：ファイルが既に存在する場合、ファイルは上書きされます。

戻り値

ブール値。
true：レイアウトが正常に適用されています。
false：レイアウトが正常に適用されていません。

例

```
dw.getLayoutNames();
var theDOM = dw.getDocumentDOM();
alert (theDOM.canApplyLayout());
if (theDOM.canApplyLayout())
    theDOM.applyLayout(1, "embed");
else
    alert("can't apply layout to this doc");
```

dom.canApplyLayout()

対応バージョン

Dreamweaver CS3

説明

ドキュメントに CSS ベースのレイアウトを適用できるかどうかをチェックします。ドキュメントの body が空で、レイアウトを適用できるページであることをチェックします。具体的には以下のとおりです。

- HTML、XHTML、ColdFusion、PHP などの基本的に HTML ベースのページ (CSS、XML、JavaScript などを除く)
- フレームセットまたはテンプレートインスタンスではないページ (テンプレート自体には適用可能)

引数

なし

戻り値

ブール値。
true：レイアウトを適用できます。
false：レイアウトを適用できません。

dw.getFilesForLayout()

対応バージョン

Dreamweaver CS3

説明

指定したレイアウト用の設定ファイルのパスを取得します。

引数

layoutIndex

- *layoutIndex* には、ゼロから開始する整数のインデックスで、レイアウトを指定します。この引数は、レイアウトのリストへのインデックスです。リストは、対応する関数の layoutNames および layoutDescriptions を返すために使用されます。

戻り値

HTML ファイルおよびプレビューイメージファイルのフルパスを含むストリング配列 (null の場合もあり)。

dw.getLayoutNames()

対応バージョン

Dreamweaver CS3

説明

使用できる CSS ベースのレイアウトの名前を取得します。

引数

なし

戻り値

レイアウト名のストリング配列。

dw.getLayoutDescriptions()

対応バージョン

Dreamweaver CS3

説明

使用できる CSS ベースのレイアウトの説明を取得します。

引数

なし

戻り値

レイアウトの説明のストリング配列。

dom.applyCSSStyle()

対応バージョン

Dreamweaver 4

説明

指定されたエレメントに、指定されたスタイルを適用します。この関数はアクティブなドキュメントのみに適用されます。

引数

elementNode、*styleName*、*{classOrID}*、*{bForceNesting}*

- *elementNode* 引数には、DOM 内のエレメントノードを指定します。*elementNode* 引数に null 値または空白のストリング ("") を指定すると、この関数は現在の選択範囲に対して動作します。

- *styleName* 引数には、CSS スタイルの名前を指定します。
- *classOrID* 引数 (オプション) には、スタイルの適用時に使用する属性 ("class" または "id") を指定します。*elementNode* 引数が null 値または空白のストリングで、選択範囲がタグで囲まれていない場合は、SPAN タグを使用してスタイルが適用されます。選択範囲が挿入ポイントである場合、Dreamweaver はスタイルを適用するタグをヒューリスティクスに基づいて決定します。
- *bForceNesting* 引数 (オプション) には、ネストできるかどうかを示すブール値を指定します。*bForceNesting* フラグが指定されている場合、ドキュメントの既存のタグは修正されず、新規の SPAN タグが挿入されます。この引数を省略した場合のデフォルト値は false です。

戻り値

なし

例

以下のコードでは、選択範囲を SPAN タグで囲むか、選択範囲が既に囲まれている場合はそのタグに CLASS 属性を適用して、選択範囲に red というスタイルを適用します。

```
var theDOM = dreamweaver.getDocumentDOM('document');
theDOM.applyCSSStyle('','red');
```

dom.getElementView()

対応バージョン

Dreamweaver 8

説明

この関数は、ドキュメント内で現在選択されているエレメントのエレメントビューを取得します。現在選択されているエレメントの状態が "normal" の場合は、そのエレメントの一番目の祖先で、状態が "full" または "hidden" のいずれかであるものを検索します。

引数

なし

戻り値

選択されているエレメントの状態を示すストリング。以下の値があります。

- "hidden" : そのエレメントには、デザインビューでコンテンツが部分的または完全に非表示になる可能性がある CSS プロパティが存在することを示します。対象となる CSS プロパティには以下のものがあります。
  - overflow: hidden、overflow: scroll、または overflow: auto
  - display: none
- "full" : そのエレメントはデフォルトでは "hidden" ですが、setElementView("full") 関数で設定されているため、現在の表示は "full" であることを示します。
- "normal" : そのエレメントは "hidden" でも "full" でもないことを示します。

例

次の例では、選択したエレメントの状態が "hidden" の場合に、その状態を "full" に変更しています。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM && getElementView() == "hidden"){
    currentDOM.setElementView("full");
    }
```

dom.getShowDivBackgrounds()

対応バージョン
Dreamweaver 8

説明
この関数は、ビジュアルエイドの [レイアウトブロックの背景] の状態を取得します。

引数
なし

戻り値
ビジュアルエイドの [レイアウトブロックの背景] がオンの場合は `true`、それ以外の場合は `false` のブール値。

例
次の例では、[レイアウトブロックの背景] がオンかどうかをチェックし、オフの場合はオンにします。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.getShowDivBackgrounds() == false){
    currentDOM.setShowDivBackgrounds(true);
    }
```

dom.getShowDivBoxModel()

対応バージョン
Dreamweaver 8

説明
この関数は、ビジュアルエイドの [レイアウトブロックのボックスモデル] の状態を取得します。

引数
なし

戻り値
ビジュアルエイドの [レイアウトのブロックボックスモデル] がオンの場合は `true`、それ以外の場合は `false` のブール値。

例
次の例では、[レイアウトブロックのボックスモデル] がオンかどうかをチェックし、オフの場合はオンにします。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.getShowDivBoxModel() == false){
    currentDOM.setShowDivBoxModel(true);
    }
```

dom.getShowDivOutlines()

対応バージョン
Dreamweaver 8

説明
この関数は、ビジュアルエイドの [レイアウトブロックのアウトライン] の状態を取得します。

引数

なし

戻り値

ビジュアルエイドの [レイアウトブロックのアウトライン] がオンの場合は `true`、それ以外の場合は `false` のブール値。

例

次の例では、[レイアウトブロックのアウトライン] がオンかどうかをチェックし、オフの場合はオンにします。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.getShowDivOutlines() == false){
    currentDOM.setShowDivOutlines(true);
    }
```

dom.removeCSSStyle()

対応バージョン

Dreamweaver 3

説明

指定したエレメントから `CLASS` 属性または `ID` 属性を削除するか、指定したエレメントを完全に囲んでいる `SPAN` タグを削除します。この関数はアクティブなドキュメントのみに適用されます。

引数

elementNode、*{classOrID}*

- *elementNode* 引数には、DOM 内のエレメントノードを指定します。*elementNode* 引数に空白のストリング ("") を指定すると、この関数は現在の選択範囲に対して動作します。
- *classOrID* 引数 (オプション) は、削除する属性 (`"class"` または `"id"`) です。*classOrID* 引数を省略した場合のデフォルト値は `"class"` です。*elementNode* 引数に `CLASS` 属性が定義されていない場合は、*elementNode* 引数を囲んでいる `SPAN` タグが削除されます。

戻り値

なし

dom.resetAllElementViews()

対応バージョン

Dreamweaver 8

説明

この関数は、内部的に生成された CSS をすべて削除して、ドキュメント内のすべてのエレメントのエレメントビューを元のビューにリセットします。

引数

forceRefresh

- *forceRefresh* 引数 (オプション) は、削除する内部 CSS がない場合に、ドキュメント全体のレンダリングを更新するかどうかを指定するブール値です。`true` の値を指定すると更新されます。デフォルト値は `false` です。

戻り値

なし

例

次の例では、レンダリングを強制的には更新せずに、ドキュメント内のすべてのエレメントのエレメントビューをリセットします。

```
var currentDOM = dw.getDocumentDOM();
currentDOM.resetAllElementViews(false);
```

dom.setElementView()

対応バージョン

Dreamweaver 8

説明

この関数は、ドキュメント内で現在選択されているエレメントのエレメントビューを設定します。現在選択されているエレメントの状態が "normal" の場合は、そのエレメントの一番目の祖先で、状態が "full" または "hidden" であるものを検索します。

引数

view

- *view*引数 (必須) には、現在選択されているエレメントを "full" または "hidden" に設定するストリングを指定します。現在選択されているエレメントの状態が "normal" の場合は、そのエレメントの一番目の祖先で、状態が "full" または "hidden" のいずれかであるものを検索します。詳細については、311 ページの dom.getElementView() を参照してください。指定可能な値は次のとおりです。
  - "full" : エレメントを "full" ビューに配置する内部 CSS が削除され、エレメントが元の状態に戻されます。
  - "hidden" : 現在選択されているエレメントが "hidden" ビューにある場合、すべてのコンテンツを表示する CSS が生成され、その CSS が内部デザインタイムスタイルシートとして適用されます。

戻り値

なし

例

詳細については、311 ページの dom.getElementView() を参照してください。

dom.setShowDivBackgrounds()

対応バージョン

Dreamweaver 8

説明

この関数は、ビジュアルエイドの [レイアウトブロックの背景] のオンとオフを切り替えます。

引数

show

- *show*引数 (必須) は、ビジュアルエイドの [レイアウトブロックの背景] をオンにするかどうかを指定するブール値です。*show* を true に設定すると、ビジュアルエイドの [レイアウトブロックの背景] がオンになります。

戻り値

なし

例

詳細については、312 ページの dom.getShowDivBackgrounds() を参照してください。

dom.setShowDivBoxModel()

対応バージョン

Dreamweaver 8

説明

この関数は、ビジュアルエイドの [レイアウトブロックのボックスモデル] のオンとオフを切り替えます。

引数

show

- *show* 引数 (必須) は、ビジュアルエイドの [レイアウトブロックのボックスモデル] をオンにするかどうかを指定するブール値です。*show* を `true` に設定すると、ビジュアルエイドの [レイアウトブロックのボックスモデル] がオンになります。

戻り値

なし

例

詳細については、312 ページの dom.getShowDivBoxModel() を参照してください。

dom.setShowDivOutlines()

対応バージョン

Dreamweaver 8

説明

この関数は、ビジュアルエイドの [レイアウトブロックのアウトライン] のオンとオフを切り替えます。

引数

show

- *show* 引数 (必須) は、ビジュアルエイドの [レイアウトブロックのアウトライン] をオンにするかどうかを指定するブール値です。*show* を `true` に設定すると、ビジュアルエイドの [レイアウトブロックのアウトライン] がオンになります。

戻り値

なし

例

詳細については、312 ページの dom.getShowDivOutlines() を参照してください。

dreamweaver.cssRuleTracker.editSelectedRule()

対応バージョン

Dreamweaver MX 2004

説明

関連 CSS で現在選択されているルールをユーザーが編集するために使用されます。この関数は、関連 CSS で現在選択されているルールを表示し、必要に応じてプロパティグリッドとそこに含まれるフローターを表示します。

引数
なし

戻り値
なし

イネーブラ
詳細については、424 ページの dreamweaver.cssRuleTracker.canEditSelectedRule() を参照してください。

dreamweaver.cssRuleTracker.newRule()

対応バージョン
Dreamweaver MX 2004

説明
ユーザーが新しいルールを作成できるように、[新規 CSS スタイル] ダイアログボックスを開きます。

引数
なし

戻り値
なし

dreamweaver.cssStylePalette.applySelectedStyle()

対応バージョン
Dreamweaver MX

説明
[スタイル] パネルでの選択に応じて、選択されたスタイルを現在アクティブなドキュメントまたは添付されたスタイルシートに適用します。

引数
{ pane }

- *pane* 引数 (オプション) は、この関数を [スタイル] パネルのどのペインに適用するかを指定するストリングです。指定可能な値は、`"stylelist"` ([すべて] モードのスタイルのリスト)、`"cascade"` ([現在] モードの適用可能な関連ルールのリスト)、`"summary"` ([現在] モードの現在の選択範囲のプロパティのリスト)、`"ruleInspector"` ([現在] モードのプロパティの編集可能なリストまたはグリッド) です。デフォルト値は `"stylelist"` です。

戻り値
なし

イネーブラ
詳細については、424 ページの dreamweaver.cssStylePalette.canApplySelectedStyle() を参照してください。

dreamweaver.cssStylePalette.attachStyleSheet()

対応バージョン
Dreamweaver 4

説明

ダイアログボックスを表示します。このダイアログボックスを使用すると、ユーザーは、[スタイル] パネルでの選択に応じて、現在アクティブなドキュメント、または添付されたスタイルシートのいずれかにスタイルシートを添付することができます。

引数

なし

戻り値

なし

dreamweaver.cssStylePalette.deleteSelectedStyle()

対応バージョン

Dreamweaver 3

説明

[スタイル] パネルで現在選択されているスタイルをドキュメントから削除します。

引数

{ pane }

- *pane* 引数 (オプション) は、この関数を [スタイル] パネルのどのペインに適用するかを指定するストリングです。指定可能な値は、`"stylelist"` ([すべて] モードのスタイルのリスト)、`"cascade"` ([現在] モードの適用可能な関連ルールのリスト)、`"summary"` ([現在] モードの現在の選択範囲のプロパティのリスト)、`"ruleInspector"` ([現在] モードのプロパティの編集可能なリストまたはグリッド) です。デフォルト値は `"stylelist"` です。

戻り値

なし

イネーブラ

詳細については、425 ページの dreamweaver.cssStylePalette.canDeleteSelectedStyle() を参照してください。

dreamweaver.cssStylePalette.duplicateSelectedStyle()

対応バージョン

Dreamweaver 3

説明

[スタイル] パネルで現在選択されているスタイルを複製し、[スタイルの複製] ダイアログボックスを表示します。このダイアログボックスを使用すると、ユーザーは、新規スタイルの名前またはセレクタを指定することができます。

引数

{ pane }

- *pane* 引数 (オプション) は、この関数を [スタイル] パネルのどのペインに適用するかを指定するストリングです。指定可能な値は、`"stylelist"` ([すべて] モードのスタイルのリスト)、`"cascade"` ([現在] モードの適用可能な関連ルールのリスト)、`"summary"` ([現在] モードの現在の選択範囲のプロパティのリスト)、`"ruleInspector"` ([現在] モードのプロパティの編集可能なリストまたはグリッド) です。デフォルト値は `"stylelist"` です。

戻り値

なし

イネーブラ

詳細については、425 ページの dreamweaver.cssStylePalette.canDuplicateSelectedStyle() を参照してください。

dreamweaver.cssStylePalette.editSelectedStyle()

対応バージョン

Dreamweaver 3

説明

[スタイル] パネルで現在選択されているスタイルの [スタイル定義] ダイアログボックスを開きます。

引数

{ pane }

- *pane* 引数 (オプション) は、この関数を [スタイル] パネルのどのペインに適用するかを指定するストリングです。指定可能な値は、`"stylelist"` ([すべて] モードのスタイルのリスト)、`"cascade"` ([現在] モードの適用可能な関連ルールのリスト)、`"summary"` ([現在] モードの現在の選択範囲のプロパティのリスト)、`"ruleInspector"` ([現在] モードのプロパティの編集可能なリストまたはグリッド) です。デフォルト値は `"stylelist"` です。

戻り値

なし

イネーブラ

詳細については、426 ページの dreamweaver.cssStylePalette.canEditSelectedStyle() を参照してください。

dreamweaver.cssStylePalette.editSelectedStyleInCodeview()

対応バージョン

Dreamweaver 8

説明

この関数は、コードビューに切り替え、[スタイル] パネルで現在選択されているスタイルのコードにマウスポインタを移動します。

引数

{ pane }

- *pane* 引数 (オプション) は、この関数を [スタイル] パネルのどのペインに適用するかを指定するストリングです。指定可能な値は、`"stylelist"` ([すべて] モードのスタイルのリスト)、`"cascade"` ([現在] モードの適用可能な関連ルールのリスト)、`"summary"` ([現在] モードの現在の選択範囲のプロパティのリスト)、`"ruleInspector"` ([現在] モードのプロパティの編集可能なリストまたはグリッド) です。デフォルト値は `"stylelist"` です。

戻り値

なし

イネーブラ

詳細については、426 ページの dreamweaver.cssStylePalette.canEditSelectedStyleInCodeview() を参照してください。

dreamweaver.cssStylePalette.editStyleSheet()

対応バージョン
Dreamweaver 3

説明
[スタイルシートの編集] ダイアログボックスを開きます。

引数
なし

戻り値
なし

イネーブラ
詳細については、426 ページの dreamweaver.cssStylePalette.canEditStyleSheet() を参照してください。

dreamweaver.cssStylePalette.getDisplayStyles()

対応バージョン
Dreamweaver 8

説明
この関数は、CSS スタイルがレンダリング中であるかどうかを判別します。デフォルト値は `true` です。

引数
なし

戻り値
CSS スタイルがレンダリングされている場合は `true`、それ以外の場合は `false` のブール値。

例
```
var areStylesRendered = dw.cssStylePalette.getDisplayStyles();
```

dreamweaver.cssStylePalette.getMediaType()

対応バージョン
Dreamweaver MX 2004

説明
レンダリングのターゲットメディアタイプを取得します。メディアタイプのデフォルトは `"screen"` です。

引数
なし

戻り値
ターゲットメディアタイプを表すストリング値。

例
```
var mediaType = dw.cssStylePalette.getMediaType();
```

dreamweaver.cssStylePalette.getSelectedStyle()

対応バージョン

Dreamweaver 3、Dreamweaver MX (Dreamweaver MX では `fullSelector` を使用可能)

説明

[スタイル] パネルで現在選択されているスタイルの名前を取得します。

引数

`fullSelector`

- *`fullSelector`* 引数には、セレクタ名全体を返すか、クラス名のみを返すかを示すブール値を指定します。省略した場合は、クラス名のみが返されます。たとえば、`p.class1` は、スタイルを `class1` のすべての `p` タグに適用し、`class1` の `div` タグなどには適用しないことを示すセレクタです。*`fullSelector`* 引数を指定しないと、`dreamweaver.cssStylePalette.getSelectedStyle()` 関数からはセレクタのクラス名 `class1` のみが返されます。*`fullSelector`* 引数は、`class1` ではなく `p.class1` を返すように関数に指示します。

戻り値

`fullSelector` 引数が `true` である場合、関数はセレクタ名全体を返します。ただし、スタイルシートノードが選択されているときは、空白のストリングを返します。

`fullSelector` 引数が `false` である場合、または省略されている場合は、選択されたスタイルのクラス名を表すストリングを返します。選択されているスタイルにクラスがないか、スタイルシートノードが選択されている場合は、空白のストリングを返します。

例

`red` というスタイルが選択されている場合に `dw.cssStylePalette.getSelectedStyle()` 関数を呼び出すと、`"red"` が返されます。

dreamweaver.cssStylePalette.getSelectedTarget() (非推奨)

対応バージョン

Dreamweaver 3 (Dreamweaver MX では [スタイル] パネルの [適用先] メニューが廃止されているため非推奨)

説明

[スタイル] パネルの一番上にある [適用先] ポップアップメニューで選択されたエレメントを取得します。

引数

なし

戻り値

非推奨の関数。常に `null` 値が返されます。

dreamweaver.cssStylePalette.getStyles()

対応バージョン

Dreamweaver 3

説明

アクティブなドキュメントにあるすべてのクラススタイルのリストを取得します。引数を指定しない場合は、クラスセレクタ名だけが返されます。*bGetIDs* 引数が `true` である場合、ID セレクタ名だけが返されます。この引数の値にかかわらず、*bGetFullSelector* 引数が `true` である場合は、セレクタ名全体が返されます。

たとえば、次のコードを含む HTML ファイルがあるとします。

```
<style>
.test{ background:none };
p.foo{ background:none };
#bar {background:none };
div#hello p.world {background:none};
```

次の表に示す呼び出しを行うと、それぞれ「結果」列の値が返されます。

関数	結果
dw.cssStylePalette.getStyles()	foo、test、world
dw.cssStylePalette.getStyles(true)	bar、hello
dw.cssStylePalette.getStyles(false, true)	p.foo、.test、div#hello p.world
dw.cssStylePalette.getStyles(true, true)	#bar、div#hello p.world

引数

{bGetIDs、bGetFullSelector}

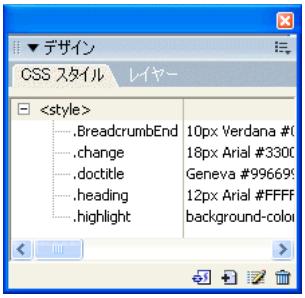
- *bGetIDs* 引数はオプションです。この引数にはブール値を指定します。`true` の場合は、ID セレクタ名 ("#" 以降の部分) だけを返します。デフォルト値は `false` です。
- *bGetFullSelector* 引数はオプションです。この引数にはブール値を指定します。`true` の場合は、名前だけではなく、セレクタストリング全体を返します。デフォルト値は `false` です。

戻り値

ドキュメントにあるすべてのクラススタイルの名前を表すストリングの配列。

例

[スタイル] パネルが次の図のように設定されている場合に `dreamweaver.cssStylePalette.getStyles()` 関数を呼び出すと、`"BreadcrumbEnd"`、`"change"`、`"doctitle"`、`"heading"`、および `"highlight"` というストリングを格納した配列が返されます。



dreamweaver.cssStylePalette.newStyle()

対応バージョン

Dreamweaver 3

説明

[新規スタイル] ダイアログボックスを開きます。

引数
なし

戻り値
なし

dreamweaver.cssStylePalette.renameSelectedStyle()

対応バージョン
Dreamweaver 3

説明
[スタイル] パネルで現在選択されているルールで使用されているクラス名と、選択したルールに含まれるそのクラス名のすべてのインスタンスの名前を変更します。

引数
{ pane }

- *pane* 引数 (オプション) は、この関数を [スタイル] パネルのどのペインに適用するかを指定するストリングです。指定可能な値は、`"stylelist"` ([すべて] モードのスタイルのリスト)、`"cascade"` ([現在] モードの適用可能な関連ルールのリスト)、`"summary"` ([現在] モードの現在の選択範囲のプロパティのリスト)、`"ruleInspector"` ([現在] モードのプロパティの編集可能なリストまたはグリッド) です。デフォルト値は `"stylelist"` です。

戻り値
なし

イネーブラ
詳細については、427 ページの dreamweaver.cssStylePalette.canRenameSelectedStyle() を参照してください。

dreamweaver.cssStylePalette.setDisplayStyles()

対応バージョン
Dreamweaver 8

説明
この関数は、CSS スタイルをレンダリングするように指定すると、開いているすべてのドキュメントのレンダリングを更新します。

引数
display

- *display* 引数はブール値です。CSS スタイルをレンダリングする場合は `true`、レンダリングしない場合は `false` を指定します。

戻り値
なし

例
次の例では、CSS スタイルのレンダリングをオフにします。

```
dw.cssStylePalette.setDisplayStyles(false);
```

dreamweaver.cssStylePalette.setMediaType()

対応バージョン
Dreamweaver MX 2004

説明
レンダリングのターゲットメディアタイプを設定します。開いているすべてのドキュメントの表示を更新します。

引数
mediaType

- *mediaType* 引数には、新しいターゲットメディアタイプを指定します。

戻り値
なし

例
```
dw.cssStylePalette.setMediaType("print");
```

dreamweaver.getBlockVisBoxModelColors()

対応バージョン
Dreamweaver 8

説明
この関数は、ビジュアルエイドの [レイアウトブロックのボックスモデル] がオンの場合に、選択されているブロックのボックスモデルのレンダリングに使用するカラーを取得します。

引数
なし

戻り値
2 つのストリングを含むストリングの配列。

- `marginColor` および `paddingColor` は RGB (赤、緑、青) カラーの 16 進数値で、#RRGGBB の形式で示されます。

例
次の例では、マージンと余白のカラーの値をチェックし、白でない場合は両方とも白に設定します。

```
var boxColors = dreamweaver.getBlockVisBoxModelColors();
if ((boxColors[0] != "#FFFFFF") || (boxColors[1] != "#FFFFFF")){
        currentDOM.setBlockVisBoxModelColors("#FFFFFF", "#FFFFFF");
        }
```

dreamweaver.getBlockVisOutlineProperties()

対応バージョン
Dreamweaver 8

説明
この関数は、ブロックを視覚化するビジュアルエイドのアウトラインのプロパティを取得します。

引数

forWhat

- *forWhat* 引数 (必須) には、ストリングを指定します。指定できる値は、"divs"、"selectedDiv"、または "layers" です。*forWhat* 引数が "divs" である場合、この関数は、すべてのレイアウトブロックのアウトラインを視覚化するビジュアルエイドに使用されるプロパティを返します。*forWhat* 引数が "selectedDiv" である場合、この関数は、選択されたレイアウトブロックのアウトラインを視覚化するビジュアルエイドに使用されるプロパティを返します。値が layers の場合は、レイヤーを示します。

戻り値

以下の 3 つのストリングから成る配列。

- color は RGB カラーの 16 進数値で、#RRGGBB の形式で示されます。
- width には、ピクセル単位の幅が示されます。
- style は、"SOLID"、"DOTTED"、"DASHED"、または "OUTSET" です。

例

次の例では、"divs" を指定した場合のアウトラインのプロパティを取得し、アウトラインのスタイルを "SOLID" にしています。

```
var outlineStyle = dw.getBlockVisOutlineProperties("divs");
if (outlineStyle[2] != "SOLID"){
    dw.setBlockVisOutlineProperties("divs", outlineStyle[0], outlineStyle[1], "SOLID");
    }
```

dreamweaver.getDivBackgroundColors()

対応バージョン

Dreamweaver 8

説明

この関数は、ビジュアルエイドの [レイアウトブロックの背景] に使用するカラーを取得します。

引数

なし

戻り値

16 個のカラーを含むストリングの配列。各カラーは、#RRGGBB 形式の RGB カラーの 16 進数値で表されます。

例

次の例では、ビジュアルエイドの [レイアウトブロックの背景] に使用される背景色を取得します。

```
var backgroundColors = dreamweaver.getDivBackgroundColors();
```

dreamweaver.setBlockVisOutlineProperties()

対応バージョン

Dreamweaver 8

説明

この関数は、ブロックを視覚化するビジュアルエイドのアウトラインのプロパティを設定します。

引数

forWhat、*color*、*width*、*{style}*

- *forWhat* 引数 (必須) は、指定されたカラーおよび幅の使用対象を指定するストリングです。指定できる値は、"divs"、"selectedDiv"、または "layers" です。値が "layers" の場合は、ビジュアルエイドの [レイアウトブロックのアウトライン] がオンのときに、すべてのレイヤーを表示する場合のアウトラインに指定したカラーと幅が使用されます。値が "divs" である場合、*color* 引数および *width* 引数は、すべての div およびその他のレイアウトブロックを表示する場合のアウトラインに使用されます。値が "selectedDiv" である場合、*color* 引数および *width* 引数は、選択されている div またはレイアウトブロックを表示する場合のアウトラインに使用されます。
- *color* 引数 (必須) は、RGB カラーを #RRGGBB の形式で指定する 16 進数値を含むストリングです。
- *width* 引数 (必須) は、アウトラインの幅をピクセル単位で指定する整数です。
- *style* 引数 (オプション) は、アウトラインのスタイルを指定するストリングです。指定できる値は、"SOLID"、"DOTTED"、"DASHED"、および "OUTSET" です。"OUTSET" の値は、レイヤーにのみ使用できます。この引数は、*forWhat* 引数の値が "selectedDiv" の場合は無視されます。

戻り値

なし

例

詳細については、323 ページの dreamweaver.getBlockVisOutlineProperties() を参照してください。

dreamweaver.setDivBackgroundColors()

対応バージョン

Dreamweaver 8

説明

この関数は、ビジュアルエイドの [レイアウトブロックの背景] に使用されるカラーを設定します。

引数

colors

- *colors* 引数 (必須) は、背景に使用するすべてのカラーを含むストリングの配列です。各カラーは、#RRGGBB の形式の 16 進数値で表します。この配列は 16 個のカラーを含んでいる必要があります。

戻り値

なし

例

次の例では、div の背景色として指定されているカラーの数が 16 個を超えていないことを確認し、超えている場合は、背景色に使用されているカラーをさまざまな濃淡の灰色に設定します。

```
var currentDOM = dw.getDocumentDOM();
var divColors = currentDOM.getDivBackgroundColors("divs");
var shadesOfGray = new Array["#000000", "#111111", "#222222", "#333333",¬
    "#444444", "#555555", "#666666", "#777777", "#888888", "#999999", ¬
    "#AAAAAA", "#BBBBBB", "#CCCCCC", "#DDDDDD", "#EEEEEE", "#FFFFFF"]¬
var howManyColors = divColors.length;
if howManyColors <= 16{
    for (var i = 0; i < howManyColors; i++)
        {
            currentDOM.setDivBackgroundColors("divs", shadeOfGray[i]);
        }
```

フレームとフレームセット関数

フレームとフレームセット関数は、フレームセット内のフレーム名の取得とフレームの 2 分割の 2 つのタスクを扱います。

dom.getFrameNames()

対応バージョン
Dreamweaver 3

説明
フレームセットにある名前付きの全フレームのリストを取得します。

引数
なし

戻り値
ストリングの配列。各ストリングは現在のフレームセット内のフレームの名前です。名称未設定のフレームはスキップします。フレームセットに名前の付いたフレームがない場合は、空白の配列を返します。

例
ドキュメントの 4 つのフレームのうち、2 つに名前が付けられている場合に `dom.getFrameNames()` 関数を呼び出すと、次のようなストリングを格納した配列が返されます。

- `"navframe"`
- `"main_content"`

dom.isDocumentInFrame()

対応バージョン
Dreamweaver 4

説明
現在のドキュメントがフレームセット内に表示されているかどうかを識別します。

引数
なし

戻り値
ドキュメントがフレームセット内に表示されている場合は `true`、それ以外の場合は `false` のブール値。

dom.saveAllFrames()

対応バージョン
Dreamweaver 4

説明
ドキュメントがフレームセットであるか、フレームセット内にある場合、ドキュメントウィンドウにあるすべてのフレームおよびフレームセットを保存します。指定されたドキュメントがフレームセット内にない場合は、ドキュメントだけを保存します。まだ保存されていないドキュメントの場合は、[新規保存] ダイアログボックスを開きます。

引数
なし

戻り値
なし

dom.splitFrame()

対応バージョン
Dreamweaver 3

説明
選択されたフレームを縦または横に 2 分割します。

引数
splitDirection

- *splitDirection* 引数には、方向を表す "up"、"down"、"left"、または "right" のいずれかのストリングを指定します。

戻り値
なし

イネーブラ
詳細については、415 ページの dom.canSplitFrame() を参照してください。

レイヤーとイメージマップ関数

レイヤーとイメージマップ関数は、レイヤーおよびイメージマップホットスポットの整列、サイズ変更、移動を行います。レイヤーとホットスポットのどちらに適用されるかは、各関数の説明セクションに記載されています。

dom.align()

対応バージョン
Dreamweaver 3

説明
選択したレイヤーまたはホットスポットを、上揃え、下揃え、右揃え、または左揃えにします。

引数
alignDirection

- *alignDirection* 引数には、レイヤーまたはホットスポットの整列の基準となる端を表すストリングを指定します。"left"、"right"、"top"、"bottom" のいずれかを指定できます。

戻り値
なし

イネーブラ
詳細については、407 ページの dom.canAlign() を参照してください。

dom.arrange()

対応バージョン
Dreamweaver 3

説明
選択したホットスポットを指定した方向に移動します。

引数
toBackOrFront

- *toBackOrFront* 引数には、ホットスポットを移動する方向 (前方または後方) を指定します。

戻り値
なし

イネーブラ
詳細については、407 ページの dom.canArrange() を参照してください。

dom.makeSizesEqual()

対応バージョン
Dreamweaver 3

説明
選択した複数のレイヤーまたはホットスポットの高さ、幅、またはその両方を統一します。最後に選択したレイヤーまたはホットスポットのサイズが基準になります。

引数
bHoriz、*bVert*

- *bHoriz* 引数には、レイヤーまたはホットスポットを横方向にサイズ変更するかどうかを示すブール値を指定します。
- *bVert* 引数には、レイヤーまたはホットスポットを縦方向にサイズ変更するかどうかを示すブール値を指定します。

戻り値
なし

dom.moveSelectionBy()

対応バージョン
Dreamweaver 3

説明
選択したレイヤーまたはホットスポットを、指定したピクセル数だけ縦および横に移動します。

引数
x、*y*

- *x* 引数には、選択範囲を横にどれだけ移動するかを表すピクセル数を指定します。
- *y* 引数には、選択範囲を縦にどれだけ移動するかを表すピクセル数を指定します。

戻り値
なし

dom.resizeSelectionBy()

対応バージョン

Dreamweaver 3

説明

現在選択されているレイヤーまたはホットスポットのサイズを変更します。

引数

left、*top*、*bottom*、*right*

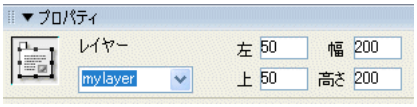
- *left* 引数には、レイヤーまたはホットスポットの左の境界線の新しい位置を指定します。
- *top* 引数には、レイヤーまたはホットスポットの上の境界線の新しい位置を指定します。
- *bottom* 引数には、レイヤーまたはホットスポットの下の境界線の新しい位置を指定します。
- *right* 引数には、レイヤーまたはホットスポットの右の境界線の新しい位置を指定します。

戻り値

なし

例

選択したレイヤーの [左]、[上]、[幅]、[高さ] の各プロパティが次のように設定されている場合に `dw.getDocumentDOM().resizeSelectionBy(-10,-30,30,10)` を呼び出すと、[左] が 40、[上] が 20、[幅] が 240、[高さ] が 240 に再設定されます。



dom.setLayerTag()

対応バージョン

Dreamweaver 3

説明

選択したレイヤーを定義する HTML タグを指定します。

引数

tagName

- *tagName* 引数には、`"layer"`、`"ilayer"`、`"div"`、`"span"` のいずれかを指定する必要があります。

戻り値

なし

レイアウト環境関数

レイアウト環境関数は、ドキュメントに対する作業の設定に関連する操作に使用されます。これらの関数は、トレーシングイメージのソース、位置、透明度に影響を与えます。また、ルーラの原点と単位の取得と設定、グリッドのオンとオフの切り替え、グリッド設定の変更、およびプラグイン再生の開始と停止を実行できます。

dom.getRulerOrigin()

対応バージョン
Dreamweaver 3

説明
ルーラの原点を取得します。

引数
なし

戻り値
2 つの整数から成る配列。配列の最初の整数は原点の *x* 座標を、2 番目の整数は *y* 座標を示します。これらの値の単位はピクセルです。

dom.getRulerUnits()

対応バージョン
Dreamweaver 3

説明
現在のルーラの単位を取得します。

引数
なし

戻り値
次の値のいずれかを含むストリング。

- `"in"`
- `"cm"`
- `"px"`

dom.getTracingImageOpacity()

対応バージョン
Dreamweaver 3

説明
ドキュメントのトレーシングイメージの透明度の設定を取得します。

引数
なし

戻り値

0 ～ 100 の値。透明度が設定されていない場合は何も返しません。

イネーブラ

詳細については、416 ページの dom.hasTracingImage() を参照してください。

dom.loadTracingImage()

対応バージョン

Dreamweaver 3

説明

[イメージソースの選択] ダイアログボックスを開きます。ユーザーがイメージを選択して [OK] をクリックすると、[トレーシングイメージ] フィールドが既に入力された状態で [ページプロパティ] ダイアログボックスが開きます。

引数

なし

戻り値

なし

dom.playAllPlugins()

対応バージョン

Dreamweaver 3

説明

ドキュメント内のすべてのプラグインコンテンツを再生します。

引数

なし

戻り値

なし

dom.playPlugin()

対応バージョン

Dreamweaver 3

説明

選択されたプラグイン項目を再生します。

引数

なし

戻り値

なし

イネーブラ

詳細については、413 ページの dom.canPlayPlugin() を参照してください。

dom.setRulerOrigin()

対応バージョン

Dreamweaver 3

説明

ルーラの原点を設定します。

引数

xCoordinate、*yCoordinate*

- *xCoordinate* 引数には、水平軸の値をピクセル数で指定します。
- *yCoordinate* 引数には、垂直軸の値をピクセル数で指定します。

戻り値

なし

dom.setRulerUnits()

対応バージョン

Dreamweaver 3

説明

現在のルーラ単位を設定します。

引数

units

- *units* 引数には、"px"、"in"、"cm" のいずれかを指定する必要があります。

戻り値

なし

dom.setTracingImagePosition()

対応バージョン

Dreamweaver 3

説明

トレーシングイメージの左上隅を、指定した座標位置に移動します。この引数を指定しない場合、[トレーシングイメージの位置を調整] ダイアログボックスが表示されます。

引数

x、*y*

- *x* 引数には、横座標を表すピクセル数を指定します。
- *y* 引数には、縦座標を表すピクセル数を指定します。

戻り値
なし

イネーブラ
詳細については、416 ページの dom.hasTracingImage() を参照してください。

dom.setTracingImageOpacity()

対応バージョン
Dreamweaver 3

説明
トレーシングイメージの透明度を設定します。

引数
opacityPercentage

- *opacityPercentage* 引数には、0 ～ 100 の数値を指定する必要があります。

戻り値
なし

イネーブラ
詳細については、416 ページの dom.hasTracingImage() を参照してください。

例
次のコードは、トレーシングイメージの透明度を 30% に設定します。

```
dw.getDocumentDOM().setTracingOpacity('30');
```

dom.snapTracingImageToSelection()

対応バージョン
Dreamweaver 3

説明
トレーシングイメージの左上隅を、現在の選択範囲の左上隅に合わせます。

引数
なし

戻り値
なし

イネーブラ
詳細については、416 ページの dom.hasTracingImage() を参照してください。

dom.stopAllPlugins()

対応バージョン
Dreamweaver 3

説明
現在ドキュメントで再生中のすべてのプラグインコンテンツを停止します。

引数
なし

戻り値
なし

dom.stopPlugin()

対応バージョン
Dreamweaver 3

説明
選択したプラグイン項目を停止します。

引数
なし

戻り値
選択範囲がプラグインで現在再生中かどうかを示すブール値。

イネーブラ
詳細については、415 ページの dom.canStopPlugin() を参照してください。

dreamweaver.arrangeFloatingPalettes()

対応バージョン
Dreamweaver 3

説明
表示されているフローティングパネルをデフォルトの位置に移動します。

引数
なし

戻り値
なし

dreamweaver.showGridSettingsDialog()

対応バージョン
Dreamweaver 3

説明
[グリッド設定] ダイアログボックスを開きます。

引数
なし

戻り値

なし

レイアウトビュー関数

レイアウトビュー関数は、ドキュメント内のレイアウトエレメントを変更する操作を処理します。これらの関数は、テーブル、列、およびセルの位置、プロパティ、外観などの設定に対して機能します。

dom.addSpacerToColumn()

対応バージョン

Dreamweaver 4

説明

現在選択されているテーブル内の指定した列の下部に、高さが 1 ピクセルの透明なスペーサーイメージを作成します。この関数は、現在の選択範囲がテーブルではない場合、または操作が成功しなかった場合は失敗します。

引数

colNum

- *colNum* 引数には、下部にスペーサーイメージを作成する列を指定します。

戻り値

なし

dom.createLayoutCell()

対応バージョン

Dreamweaver 4

説明

現在のドキュメントの指定した位置に、指定したサイズのレイアウトセルを作成します。セルは、既存のレイアウトテーブル内またはページ上の既存のコンテンツの下の領域に作成されます。セルを既存のレイアウトテーブル内に作成する場合、他のレイアウトセルまたはネストされたレイアウトテーブルをオーバーラップしたり含むことはできません。セルの四角形が既存のレイアウトテーブル内に入らない場合、Dreamweaver は新しいセルを収めるためのレイアウトテーブルの作成を試みます。この関数によって、ドキュメントがレイアウトビューに変更されることはありません。この関数は、セルを作成できない場合は失敗します。

引数

left、*top*、*width*、*height*

- *left* 引数には、セルの左ボーダーの *x* 方向の位置を指定します。
- *top* 引数には、セルの上ボーダーの *y* 方向の位置を指定します。
- *width* 引数には、セルの幅をピクセル単位で指定します。
- *height* 引数には、セルの高さをピクセル単位で指定します。

戻り値

なし

dom.createLayoutTable()

対応バージョン
Dreamweaver 4

説明
現在のドキュメントの指定した位置に、指定したサイズのレイアウトテーブルを作成します。テーブルは、既存のテーブル内またはページ上の既存のコンテンツの下の領域に作成されます。テーブルを既存のレイアウトテーブル内に作成する場合、他のレイアウトセルまたはネストされたレイアウトテーブルをオーバーラップすることはできませんが、含むことはできます。この関数によって、ドキュメントがレイアウトビューに変更されることはありません。この関数は、テーブルを作成できない場合は失敗します。

引数
left、*top*、*width*、*height*

- *left* 引数には、テーブルの左ボーダーの *x* 方向の位置を指定します。
- *top* 引数には、テーブルの上ボーダーの *y* 方向の位置を指定します。
- *width* 引数には、テーブルの幅をピクセル単位で指定します。
- *height* 引数には、テーブルの高さをピクセル単位で指定します。

戻り値
なし

dom.doesColumnHaveSpacer()

対応バージョン
Dreamweaver 4

説明
Dreamweaver で生成されたスペーサーイメージが列に含まれるかどうかを判別します。現在の選択範囲がテーブルでない場合は実行されません。

引数
colNum

- *colNum* 引数には、スペーサーイメージをチェックする列を指定します。

戻り値
現在選択されているテーブルの指定した列に、Dreamweaver で生成されたスペーサーイメージが含まれる場合は `true`、含まれない場合は `false`。

dom.doesGroupHaveSpacers()

対応バージョン
Dreamweaver 4

説明
Dreamweaver で生成されたスペーサーイメージの行が、現在選択されているテーブルに含まれるかどうかを判別します。現在の選択範囲がテーブルでない場合は実行されません。

引数
なし

戻り値
テーブルにスペーサーイメージの行が含まれる場合は `true`、それ以外の場合は `false`。

dom.getClickedHeaderColumn()

対応バージョン
Dreamweaver 4

説明
ユーザーがレイアウトビューでテーブルのヘッダーにあるメニューボタンをクリックして、そのテーブルヘッダーメニューを表示したときに、ユーザーがクリックした列のインデックスを返します。テーブルヘッダーメニューが表示されない場合の結果は未定義です。

引数
なし

戻り値
列のインデックスを表す整数。

dom.getShowLayoutTableTabs()

対応バージョン
Dreamweaver 4

説明
レイアウトビューで、現在のドキュメントにレイアウトテーブルのタブが表示されているかどうかを判別します。

引数
なし

戻り値
レイアウトビューで、現在のドキュメントにレイアウトテーブルのタブが表示されている場合は `true`、表示されていない場合は `false`。

dom.getShowLayoutView()

対応バージョン
Dreamweaver 4

説明
現在のドキュメントが、レイアウトビューとスタンダードビューのいずれで表示されているかを判別します。

引数
なし

戻り値
現在のドキュメントがレイアウトビューで表示されている場合は `true`、スタンダードビューで表示されている場合は `false`。

dom.isColumnAutostretch()

対応バージョン
Dreamweaver 4

説明
ドキュメントサイズに応じて、列が自動的に拡大または縮小するように設定されているかどうかを判別します。この関数は、現在の選択範囲がテーブルでない場合は実行されません。

引数
colNum

- *colNum* 引数には、サイズを自動的に設定する列または幅を固定する列を指定します。

戻り値
現在選択されているテーブルの指定されたインデックスの列が自動伸縮に設定されている場合は `true`、それ以外の場合は `false`。

dom.makeCellWidthsConsistent()

対応バージョン
Dreamweaver 4

説明
この関数は、現在選択されているテーブルで、現在レンダリングされている列の幅と一致するように HTML の各列の幅を設定します。この関数は、現在の選択範囲がテーブルではない場合、または操作が成功しなかった場合は失敗します。

引数
なし

戻り値
なし

dom.removeAllSpacers()

対応バージョン
Dreamweaver 4

説明
Dreamweaver が生成したすべてのスペーサーイメージを、現在選択されているテーブルから削除します。この関数は、現在の選択範囲がテーブルではない場合、または操作が成功しなかった場合は失敗します。

引数
なし

戻り値
なし

dom.removeSpacerFromColumn()

対応バージョン
Dreamweaver 4

説明
指定した列からスペーサーイメージを削除します。それによって、Dreamweaver が生成したスペーサーイメージがなくなった場合は、スペーサー行も削除します。この関数は、現在の選択範囲がテーブルではない場合、または操作が成功しなかった場合は失敗します。

引数
colNum

- *colNum* 引数には、スペーサーイメージを削除する列を指定します。

戻り値
なし

dom.setColumnAutostretch()

対応バージョン
Dreamweaver 4

説明
列に対して、自動サイズ設定にするか固定幅にするかを切り替えます。*bAutostretch* に `true` を指定すると、現在選択されているテーブルの指定されたインデックスの列は、自動伸縮に設定されます。それ以外の場合は、現在レンダリングされている幅に固定されます。この関数は、現在の選択範囲がテーブルではない場合、または幅を設定できなかった場合は失敗します。

引数
colNum、*bAutostretch*

- *colNum* 引数には、サイズを自動的に設定する列または幅を固定する列を指定します。
- *bAutostretch* 引数には、列を自動伸縮に設定する (`true`) か、固定幅に設定する (`false`) かを指定します。

戻り値
なし

dom.getShowBlockBackgrounds()

対応バージョン
Dreamweaver 8

説明
この関数は、すべてのブロックまたは div の背景を強制的に色付けするビジュアルエイドの状態を取得します。

引数
allblocks

- *allblocks* 引数 (必須) には、ブール値を指定します。値を `true` に設定すると、div タグのみに適用されます。値を `false` に設定すると、すべてのブロックエレメントに適用されます。

戻り値

ブール値。背景が強制的に色付けされている場合は true、色付けされていない場合は false。

例

次の例では、すべてのブロックの背景が強制的に色付けされているかどうかをチェックし、強制的に色付けされていない場合は、すべてのブロックの背景を強制的に色付けします。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.getShowBlockBackgrounds(false) == false){
    currentDOM.setShowBlockBackgrounds(false);
    }
```

dom.getShowBlockBorders()

対応バージョン

Dreamweaver 8

説明

この関数は、すべてのブロックまたはすべての div のボーダーを描画するビジュアルエイドの状態を取得します。

引数

allblocks

- *allblocks* 引数 (必須) には、ブール値を指定します。値を true に設定すると、div タグの状態のみが取得されます。値を false に設定すると、すべてのブロックエレメントの状態が取得されます。

戻り値

ブール値。ボーダーが表示される場合は true、表示されない場合は false。

例

次の例では、ブロックのボーダーを描画するビジュアルエイドがオンかどうかをチェックし、オフの場合はオンにします。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.getShowBlockBorders(false) == false){
    currentDOM.setShowBlockBorders(true);
    }
```

dom.getShowBlockIDs()

対応バージョン

Dreamweaver 8

説明

この関数は、すべてのブロックまたは div の ID とクラスの情報を表示するビジュアルエイドの状態を取得します。

引数

allblocks

- *allblocks* 引数 (必須) には、ブール値を指定します。値を true に設定すると、div タグの ID とクラスのみが表示されます。値を false に設定すると、すべてのブロックエレメントの ID とクラスが表示されます。

戻り値

ブール値。ID が表示される場合は true、表示されない場合は false。

例

次の例では、ブロックの ID が表示されるかどうかをチェックし、表示されない場合は表示します。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.getShowBlockIDs(false) == false){
    currentDOM.setShowBlockIDs(true);
    }
```

dom.getShowBoxModel()

対応バージョン

Dreamweaver 8

説明

この関数は、選択したブロックのボックスモデル全体を色付けするビジュアルエイドのオンとオフを切り替えます。

引数

なし

戻り値

なし

例

次の例では、選択したボックスのボックスモデル全体が色付きで表示されるかどうかをチェックし、色付きで表示されない場合は色付けします。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.getShowBoxModel() == false){
    currentDOM.setShowBoxModel(true);
    }
```

dom.setShowBlockBackgrounds()

対応バージョン

Dreamweaver 8

説明

この関数は、すべてのブロックまたはすべての div の背景を強制的に色付けするビジュアルエイドのオンとオフを切り替えます。

引数

allblocks

- *allblocks* 引数 (必須) には、ブール値を指定します。値を `true` に設定すると、div タグのみに背景のカラーリングが適用されます。値を `false` に設定すると、すべてのブロックエレメントに背景のカラーリングが適用されます。

戻り値

なし

例

詳細については、339 ページの dom.getShowBlockBackgrounds() を参照してください。

dom.setShowBlockBorders()

対応バージョン
Dreamweaver 8

説明
この関数は、すべてのブロックまたは div のボーダーを描画するビジュアルエイドのオンとオフを切り替えます。

引数
allblocks

- *allblocks* 引数 (必須) には、ブール値を指定します。値を `true` に設定すると、div タグのみにボーダーが適用されます。値を `false` に設定すると、すべてのブロックエレメントにボーダーが適用されます。

戻り値
なし

例
詳細については、340 ページの dom.getShowBlockBorders() を参照してください。

dom.setShowBlockIDs()

対応バージョン
Dreamweaver 8

説明
この関数は、すべてのブロックまたは div の ID とクラスを表示するビジュアルエイドのオンとオフを切り替えます。

引数
allblocks

- *allblocks* 引数 (必須) には、ブール値を指定します。値を `true` に設定すると、div タグの ID とクラスのみが表示されます。値を `false` に設定すると、すべてのブロックエレメントの ID とクラスが表示されます。

戻り値
なし

例
詳細については、340 ページの dom.getShowBlockIDs() を参照してください。

dom.setShowBoxModel()

対応バージョン
Dreamweaver 8

説明
この関数は、選択したブロックのボックスモデル全体を色付けするビジュアルエイドの状態を設定します。

引数
なし

戻り値

ブール値。ボックスモデルが表示される場合は `true`、表示されない場合は `false`。

例

詳細については、341 ページの dom.getShowBoxModel() を参照してください。

dom.setShowLayoutTableTabs()

対応バージョン

Dreamweaver 4

説明

現在のドキュメントがレイアウトビューで表示される場合は、常にレイアウトテーブルのタブを表示するように現在のドキュメントを設定します。この関数によって、ドキュメントがレイアウトビューに変更されることはありません。

引数

bShow

- *bShow* 引数には、現在のドキュメントがレイアウトビューで表示されるときに、レイアウトテーブルのタブを表示するかどうかを指定します。*bShow* が `true` の場合、Dreamweaver ではタブが表示されます。*bShow* が `false` の場合は表示されません。

戻り値

なし

dom.setShowLayoutView()

対応バージョン

Dreamweaver 4

説明

bShow が `true` の場合に、現在のドキュメントをレイアウトビューで表示します。

引数

bShow

- *bShow* 引数には、現在のドキュメントに対してレイアウトビューとスタンダードビューとを切り替えるブール値を指定します。*bShow* が `true` の場合、現在のドキュメントはレイアウトビューに表示されます。*bShow* が `false` の場合は、スタンダードビューに表示されます。

戻り値

なし

ズーム関数

ズーム関数は、デザインビューでのズームインおよびズームアウトを実行します。

dreamweaver.activeViewScale()

対応バージョン
Dreamweaver 8

説明
この関数は、可変浮動小数点プロパティを取得または設定します。値を取得する場合は、[ズーム] コンボボックスに表示される倍率を 100 で割った値でアクティブなビューの倍率が返されます。たとえば、100% であれば 1.0、50% であれば 0.5 というように返されます。値を設定する場合は、表示比率のコンボボックスの値が設定されます。指定できる値は 0.06 ～ 64.00 (6% ～ 6400%) です。

例
次の例では、現在のビューの倍率値を取得し、倍率が 100% 以下でズームインが可能な場合は、ズームインします。

```
if (canZoom() && dreamweaver.activeViewScale <= 1.0) {
    zoomIn();
    }
```

次の例では、現在のビューの倍率値を 50% に設定します。

```
dreamweaver.activeViewScale = 0.50;
```

dreamweaver.fitAll()

対応バージョン
Dreamweaver 8

説明
この関数は、デザインビューの現在表示されている部分にドキュメント全体が収まるように、ズームインまたはズームアウトします。

引数
なし

戻り値
なし

イネーブラ
詳細については、424 ページの dreamweaver.canZoom() を参照してください。

例
```
if (canZoom()){
    fitAll();
    }
```

dreamweaver.fitSelection()

対応バージョン
Dreamweaver 8

説明
この関数は、デザインビューの現在表示されている部分に現在の選択範囲が収まるように、ズームインまたはズームアウトします。

引数
なし

戻り値
なし

イネーブラ
詳細については、419 ページの dreamweaver.canFitSelection() を参照してください。

例
```
if (canFitSeletion()){
    fitSelection();
    }
```

dreamweaver.fitWidth()

対応バージョン
Dreamweaver 8

説明
この関数は、デザインビューの現在表示されている部分にドキュメント全体の幅が収まるように、ズームインまたはズームアウトします。

引数
なし

戻り値
なし

イネーブラ
詳細については、424 ページの dreamweaver.canZoom() を参照してください。

例
```
if (canZoom()){
    fitWidth();
    }
```

dreamweaver.zoomIn()

対応バージョン
Dreamweaver 8

説明
この関数は、現在アクティブなデザインビューをズームインします。ズームのレベルは、[表示比率] メニューの次の設定済みの値です。次の設定済みの値がない場合、この関数は実行されません。

引数
なし

戻り値
なし

イネーブラ

詳細については、424 ページの dreamweaver.canZoom() を参照してください。

例

```
if (canZoom()){
    zoomIn();
    }
```

dreamweaver.zoomOut()

対応バージョン

Dreamweaver 8

説明

この関数は、現在アクティブなデザインビューをズームアウトします。ズームのレベルは、[表示比率] メニューの次の設定済みの値です。次の設定済みの値がない場合、この関数は実行されません。

引数

なし

戻り値

なし

イネーブラ

詳細については、424 ページの dreamweaver.canZoom() を参照してください。

例

```
if (canZoom()){
    zoomOut();
    }
```

ガイド関数およびプロパティ

ガイド関数およびプロパティを使用すると、HTML ページのエレメントのサイズ測定およびレイアウトを行うためのガイドを表示、操作、および削除できます。

dom.clearGuides()

対応バージョン

Dreamweaver 8

説明

この関数は、ドキュメント内のすべてのガイドを削除するかどうかを指定します。

引数

なし

戻り値

なし

例

次の例では、ドキュメントに少なくとも 1 つのガイドがある場合に、ドキュメントからすべてのガイドが削除されます。

```
var currentDOM = dw.getDocumentDOM();
    if (currentDOM.hasGuides() == true) {
    currentDOM.clearGuides();
}
```

dom.createHorizontalGuide()

対応バージョン

Dreamweaver 8

説明

この関数は、ドキュメント内の現在の位置に水平方向のガイドを作成します。

引数

location

- *location* 引数は、ガイドの位置を表すストリングです。この引数には、値と単位の両方を、間にスペースを入れずに 1 つのストリングとして指定します。指定できる単位は、"px" (ピクセル) および "%" (パーセント) です。たとえば、10 ピクセルと指定するには location = "10px"、50% と指定するには location = "50%" になります。

戻り値

なし

例

次の例では、ドキュメント内の現在の位置に水平方向のガイドを作成します。

```
var currentDOM = dw.getDocumentDOM();
currentDOM.createHorizontalGuide("10px");
```

dom.createVerticalGuide()

対応バージョン

Dreamweaver 8

説明

この関数は、ドキュメント内の現在の位置に垂直方向のガイドを作成します。

引数

location

- *location* 引数は、ガイドの位置を表すストリングです。この引数には、値と単位の両方を、間にスペースを入れずに 1 つのストリングとして指定します。指定できる単位は、"px" (ピクセル) および "%" (パーセント) です。たとえば、10 ピクセルと指定するには location = "10px"、50% と指定するには location = "50%" になります。

戻り値

なし

例

次の例では、ドキュメント内の現在の位置に垂直方向のガイドを作成します。

```
var currentDOM = dw.getDocumentDOM();
currentDOM.createVerticalGuide("10px");
```

dom.deleteHorizontalGuide()

対応バージョン

Dreamweaver 8

説明

この関数は、指定した位置にある水平方向のガイドを削除します。

引数

location

- *location* 引数は、ドキュメント内のテストする位置を表すストリングです。この引数には、値と単位の両方を、間にスペースを入れずに 1 つのストリングとして指定します。指定できる単位は、"px" (ピクセル) および "%" (パーセント) です。たとえば、10 ピクセルと指定するには location = "10px"、50% と指定するには location = "50%" になります。

戻り値

なし

例

次の例では、ドキュメント内の指定した位置にある水平方向のガイドを削除します。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.hasHorizontalGuide("10px") == true) {
    currentDOM.deleteHorizonalGuide("10px");
}
```

dom.deleteVerticalGuide()

対応バージョン

Dreamweaver 8

説明

この関数は、指定した位置にある垂直方向のガイドを削除します。

引数

location

- *location* 引数は、ドキュメント内のテストする位置を表すストリングです。この引数には、値と単位の両方を、間にスペースを入れずに 1 つのストリングとして指定します。指定できる単位は、"px" (ピクセル) および "%" (パーセント) です。たとえば、10 ピクセルと指定するには location = "10px"、50% と指定するには location = "50%" になります。

戻り値

なし

例

次の例では、ドキュメント内の指定した位置にある垂直方向のガイドを削除します。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.hasVerticalGuide("10px") == true) {
    currentDOM.deleteVerticalGuide("10px");
}
```

dom.guidesColor

対応バージョン

Dreamweaver 8

説明

この可変カラープロパティは、ドキュメント内のガイドのカラーを決定します。このプロパティは、設定することも取得することもできます。

引数

なし

戻り値

なし

例

次の例では、ガイドを灰色にします。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.guidesColor != "#444444"){
    currentDOM.guidesColor = "#444444";
    }
```

dom.guidesDistanceColor

対応バージョン

Dreamweaver 8

説明

この可変カラープロパティは、ドキュメント内のガイドの遠くのフィードバックカラーを決定します。このプロパティは、設定することも取得することもできます。

引数

なし

戻り値

なし

例

次の例では、ガイドの遠くのフィードバックカラーを灰色にします。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.guidesDistanceColor != "#CCCCCC"){
    currentDOM.guidesDistanceColor = "#CCCCCC";
    }
```

dom.guidesLocked

対応バージョン
Dreamweaver 8

説明
この可変ブール値プロパティは、ドキュメント内のガイドをロックするかどうかを決定します。このプロパティは、設定することも取得することもできます。

引数
なし

戻り値
なし

例
次の例では、ガイドがロックされていない場合にロックします。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.guidesLocked == false) {
    currentDOM.guidesLocked = true;
}
```

dom.guidesSnapToElements

対応バージョン
Dreamweaver 8

説明
この可変ブール値プロパティは、ドキュメント内のガイドをエレメントに吸着させるかどうかを決定します。このプロパティは、設定することも取得することもできます。

引数
なし

戻り値
なし

例
次の例では、ドキュメント内のガイドをエレメントに吸着させます。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.guidesSnapToElements == false) {
    currentDOM.guidesSnapToElements = true;
    }
```

dom.guidesVisible

対応バージョン
Dreamweaver 8

説明

この可変ブール値プロパティは、ドキュメント内のガイドを表示するかどうかを決定します。このプロパティは、設定することも取得することもできます。

引数

なし

戻り値

なし

例

次の例では、ガイドが表示されていない場合に表示します。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.guidesVisible == false) {
    currentDOM.guidesVisible = true;
}
```

dom.hasGuides()

対応バージョン

Dreamweaver 8

説明

このプロパティは、ドキュメント内に 1 つでもガイドが存在するかどうかを判別します。このプロパティは、設定することも取得することもできます。

引数

なし

戻り値

なし

例

次の例では、ドキュメントに少なくとも 1 つのガイドがある場合に、ドキュメントからすべてのガイドが削除されます。

```
var currentDOM = dw.getDocumentDOM();
    if (currentDOM.hasGuides() == true) {
    currentDOM.clearGuides();
}
```

dom.hasHorizontalGuide()

対応バージョン

Dreamweaver 8

説明

この関数は、ドキュメント内の指定した位置に水平方向のガイドがあるかどうかを判別します。

引数

location

- *location* 引数は、ドキュメント内のテストする位置を表すストリングです。この引数には、値と単位の両方を、間にスペースを入れずに 1 つのストリングとして指定します。指定できる単位は、"px" (ピクセル) および "%" (パーセント) です。たとえば、10 ピクセルと指定するには location = "10px"、50% と指定するには location = "50%" になります。

戻り値

指定した位置に水平方向のガイドがある場合は true、それ以外の場合は false のブール値。

例

次の例では、ドキュメント内の指定した位置に水平方向のガイドがある場合に、ドキュメント内のすべてのガイドを削除します。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.hasHorizontalGuide("10px") == true) {
    currentDOM.clearGuides();
}
```

dom.hasVerticalGuide()

対応バージョン

Dreamweaver 8

説明

この関数は、ドキュメント内の現在の位置に垂直方向のガイドがあるかどうかを判別します。

引数

location

- *location* 引数は、ドキュメント内のテストする位置を表すストリングです。この引数には、値と単位の両方を、間にスペースを入れずに 1 つのストリングとして指定します。指定できる単位は、"px" (ピクセル) および "%" (パーセント) です。たとえば、10 ピクセルと指定するには location = "10px"、50% と指定するには location = "50%" になります。

戻り値

指定した位置に垂直方向のガイドがある場合は true、それ以外の場合は false のブール値。

例

次の例では、ドキュメント内の指定した位置に垂直方向のガイドがある場合に、ドキュメント内のすべてのガイドを削除します。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.hasVerticalGuide("10px") == true) {
    currentDOM.clearGuides();
}
```

dom.snapToGuides

対応バージョン

Dreamweaver 8

説明

この可変ブール値プロパティは、ドキュメント内のエレメントをガイドに吸着させるかどうかを決定します。このプロパティは、設定することも取得することもできます。

引数

なし

戻り値

なし

例

次の例では、ドキュメント内のエレメントをガイドに吸着させます。

```
var currentDOM = dw.getDocumentDOM();
if (currentDOM.snapToGuides == false) {
    currentDOM.snapToGuides = true;
}
```

テーブル編集関数

テーブル関数を使用して、テーブルの行と列の追加と削除、列幅と行の高さの変更、ピクセルとパーセント間の単位換算、およびその他の標準のテーブル編集タスクを実行できます。

dom.convertWidthsToPercent()

対応バージョン

Dreamweaver 3

説明

現在のテーブルのすべての WIDTH 属性を、ピクセルからパーセントに変換します。

引数

なし

戻り値

なし

dom.convertWidthsToPixels()

対応バージョン

Dreamweaver 4

説明

現在のテーブルのすべての WIDTH 属性を、パーセントからピクセルに変換します。

引数

なし

戻り値

なし

dom.decreaseColspan()

対応バージョン
Dreamweaver 3

説明
列を 1 単位縮小します。

引数
なし

戻り値
なし

イネーブラ
詳細については、409 ページの dom.canDecreaseColspan() を参照してください。

dom.decreaseRowspan()

対応バージョン
Dreamweaver 3

説明
行を 1 単位縮小します。

引数
なし

戻り値
なし

イネーブラ
詳細については、409 ページの dom.canDecreaseRowspan() を参照してください。

dom.deleteTableColumn()

対応バージョン
Dreamweaver 3

説明
選択したテーブル列を削除します。

引数
なし

戻り値
なし

イネーブラ
詳細については、410 ページの dom.canDeleteTableColumn() を参照してください。

dom.deleteTableRow()

対応バージョン
Dreamweaver 3

説明
選択したテーブル行を削除します。

引数
なし

戻り値
なし

イネーブラ
詳細については、410 ページの dom.canDeleteTableRow() を参照してください。

dom.doDeferredTableUpdate()

対応バージョン
Dreamweaver 3

説明
[一般] 環境設定で [より早いテーブル編集] オプションが選択されている場合、選択範囲をテーブルの外に移動しないまま、最新の変更をテーブルレイアウトに反映します。この関数は、[より早いテーブル編集] が選択されていない場合は無効です。

引数
なし

戻り値
なし

dom.getShowTableWidths()

対応バージョン
Dreamweaver MX 2004

説明
テーブル幅が、標準または拡張テーブルモード (レイアウトモード以外) で表示されるかどうかを返します。Dreamweaver でテーブルタブがレイアウトモードで表示されるかどうかの詳細については、337 ページの dom.getShowLayoutTableTabs() を参照してください。

引数
なし

戻り値
テーブル幅が標準または拡張テーブルモードで表示される場合は `true`、それ以外の場合は `false` のブール値。

dom.getTableExtent()

対応バージョン
Dreamweaver 3

説明
選択したテーブルの列と行の数を取得します。

引数
なし

戻り値
2 つの整数を含む配列。配列の最初の整数が列の数、2 番目の整数が行の数を示します。テーブルが選択されていない場合は、何も返されません。

dom.increaseColspan()

対応バージョン
Dreamweaver 3

説明
列を 1 単位拡大します。

引数
なし

戻り値
なし

イネーブラ
詳細については、411 ページの dom.canIncreaseColspan() を参照してください。

dom.increaseRowspan()

対応バージョン
Dreamweaver 3

説明
行を 1 単位拡大します。

引数
なし

戻り値
なし

イネーブラ
詳細については、411 ページの dom.canIncreaseRowspan() を参照してください。

dom.insertTableColumns()

対応バージョン
Dreamweaver 3

説明
現在のテーブルに指定した数の列を挿入します。

引数
numberOfCols、*bBeforeSelection*

- *numberOfCols* 引数には、挿入する列の数を指定します。
- *bBeforeSelection* 引数はブール値です。選択範囲を含む列の前に列を挿入する場合は `true`、それ以外の場合は `false` を指定します。

戻り値
なし

イネーブラ
詳細については、411 ページの dom.canInsertTableColumns() を参照してください。

dom.insertTableRows()

対応バージョン
Dreamweaver 3

説明
現在のテーブルに指定した数の行を挿入します。

引数
numberOfRows、*bBeforeSelection*

- *numberOfRows* 引数には、挿入する行の数を指定します。
- *bBeforeSelection* 引数はブール値です。選択範囲を含む行の上に行を挿入する場合は `true`、それ以外の場合は `false` を指定します。

戻り値
なし

イネーブラ
詳細については、412 ページの dom.canInsertTableRows() を参照してください。

dom.mergeTableCells()

対応バージョン
Dreamweaver 3

説明
選択したテーブルセルをマージします。

引数
なし

戻り値
なし

イネーブラ
詳細については、413 ページの dom.canMergeTableCells() を参照してください。

dom.removeAllTableHeights()

対応バージョン
Dreamweaver 3

説明
選択したテーブルからすべての HEIGHT 属性を削除します。

引数
なし

戻り値
なし

dom.removeAllTableWidths()

対応バージョン
Dreamweaver 3

説明
選択したテーブルからすべての WIDTH 属性を削除します。

引数
なし

戻り値
なし

dom.removeColumnWidth()

対応バージョン
Dreamweaver MX 2004

説明
選択した 1 列からすべての WIDTH 属性を削除します。

引数
なし

戻り値
なし

dom.selectTable()

対応バージョン
Dreamweaver 3

説明
テーブル全体を選択します。

引数
なし

戻り値
なし

イネーブラ
詳細については、414 ページの dom.canSelectTable() を参照してください。

dom.setShowTableWidths()

対応バージョン
Dreamweaver MX 2004

説明
標準または拡張テーブルモード (レイアウトモード以外) で、テーブル幅の表示のオンとオフを切り替えます。この関数は、特に指定しない限り、現在のドキュメントと今後作成するすべてのドキュメントに対して値を設定します。レイアウトモードにおけるテーブルタブ表示の設定の詳細については、343 ページの dom.setShowLayoutTableTabs() を参照してください。

引数
bShow

- *bShow* 引数はブール値です。現在のドキュメントが標準または拡張テーブルモード (レイアウトモード以外) で表示されているときに、テーブルにテーブル幅を表示するかどうかを指定します。*bShow* が `true` の場合は、幅が表示されます。*bShow* が `false` の場合は表示されません。

戻り値
なし

dom.setTableCellTag()

対応バージョン
Dreamweaver 3

説明
選択したセルのタグを指定します。

引数
tdOrTh

- *tdOrTh* 引数には、`"td"` または `"th"` のいずれかを指定する必要があります。

戻り値
なし

dom.setTableColumns()

対応バージョン
Dreamweaver 3

説明
選択したテーブルの列の数を設定します。

引数
numberOfCols

- *numberOfCols* 引数には、テーブルに設定する列数を指定します。

戻り値
なし

dom.setTableRows()

対応バージョン
Dreamweaver 3

説明
選択したテーブルの行の数を設定します。

引数
numberOfCols

- *numberOfRows* 引数には、テーブルに設定する行数を指定します。

戻り値
なし

dom.showInsertTableRowsOrColumnsDialog()

対応バージョン
Dreamweaver 3

説明
[行または列の挿入] ダイアログボックスを開きます。

引数
なし

戻り値
なし

イネーブラ
詳細については、411 ページの dom.canInsertTableColumns() または 412 ページの dom.canInsertTableRows() を参照してください。

dom.splitTableCell()

対応バージョン

Dreamweaver 3

説明

現在のテーブルセルを指定した数の行または列に分割します。両方の引数を指定しないと、[セルの分割] ダイアログボックスが表示されます。

引数

{colsOrRows}、*{numberToSplitInto}*

- *colsOrRows* 引数 (オプション) には、"columns" または "rows" のいずれかを指定する必要があります。
- *numberToSplitInto* 引数 (オプション) には、セルを分割する行または列の数を指定します。

戻り値

なし

イネーブラ

詳細については、415 ページの dom.canSplitTableCell() を参照してください。

第 18 章 : コード

コード関数は、コードビューに表示されるドキュメントに関する処理を実行します。コード関数で実行する処理には、コードヒントメニューへの新規 menu タグまたは function タグの追加、ストリングパターンの検索と置換、ドキュメントからの現在の選択範囲の削除、すべてのコードまたは選択したコードの印刷、タグの編集、選択したコードへのシンタックスフォーマットの適用などがあります。

本章では関数をグループ分けして、それぞれを以下の項で説明します。

コード関数

コードヒントは、コードビューで特定の文字パターンを入力したときに Adobe® Dreamweaver® CS3 によって表示されるメニューです。コードヒントによって、入力しようとしているストリングの候補のリストが提示されるため、入力の手間を省くことができます。入力しようとしているストリングがメニューに表示されている場合は、そのストリングまでスクロールして Enter キーまたは Return キーを押すと、入力が完了します。たとえば、「<」を入力すると、ポップアップメニューにタグ名のリストが表示されます。タグ名の残り部分を入力する代わりに、メニューからタグを選択してテキストに追加することができます。

Dreamweaver にコードヒントメニューを追加するには、"CodeHints.xml" ファイルの中に定義します。"CodeHints.xml" ファイルの詳細については、『Dreamweaver 拡張ガイド』を参照してください。

Dreamweaver に "CodeHints.xml" ファイルのコンテンツがロードされた後に、JavaScript を使用して動的に新しいコードヒントメニューを追加することもできます。たとえば、JavaScript のコードによって、[バインディング] パネルのセッション変数リストの項目を設定します。同じコードを使用してコードヒントメニューを追加すると、ユーザーがコードビューで「**Session**」と入力したときに、セッション変数のメニューが表示されます。

"CodeHints.xml" ファイルと JavaScript API はコードヒントエンジンの便利なサブセットを備えていますが、Dreamweaver にある機能の一部は利用できません。たとえば、カラーピッカーを表示する JavaScript フックは存在しないため、JavaScript を使用して [属性値] メニューを表すことはできません。実行できるのは、テキスト項目のメニューを表示し、そのメニューからテキストを挿入することだけです。

コードカラーリングでは、コードのカラーリングスタイルの指定、既存のコードカラーリングスキームの修正、またはコードカラーリングスキームの新規作成を行います。コードのカラーリングスタイルとカラーリングスキームは、"Colors.xml" ファイルとコードカラーリングスキームファイルを修正して指定することができます。これらのファイルの詳細については、『Dreamweaver 拡張ガイド』を参照してください。

コードヒントおよびコードカラーリング用の JavaScript API は、次の関数から構成されています

dreamweaver.codeHints.addMenu()

対応バージョン

Dreamweaver MX

説明

"CodeHints.xml" ファイルに新しい menu タグを動的に定義します。同じパターンとドキュメントタイプが指定された menu タグが既にある場合は、既存のメニューに項目が追加されます。

引数

menuGroupId、*pattern*、*labelArray*、*{valueArray}*、*{iconArray}*、*{doctypes}*、*{casesensitive}*

- *menuGroupId* 引数には、menugroup タグのいずれかの ID 属性を指定します。
- *pattern* 引数には、新しい menu タグの pattern 属性を指定します。
- *labelArray* 引数は、ストリングの配列です。各ストリングは、ポップアップメニューの 1 つのメニュー項目のテキストです。
- *valueArray* 引数 (オプション) は、ストリングの配列です。*labelArray* 引数と同じ長さにする必要があります。ユーザーがポップアップメニューから項目を選択すると、この配列のストリングがドキュメントに挿入されます。挿入されるストリングが常にメニューラベルと同じである場合は、この引数の値に null 値を指定できます。
- *iconArray* 引数 (オプション) は、ストリングまたはストリングの配列です。ストリングの場合は、メニューのすべての項目に対して使用される 1 つのイメージファイルの URL を指定します。ストリングの配列の場合は、*labelArray* 引数と同じ長さにする必要があります。各ストリングは、対応するメニュー項目のアイコンとして使用されるイメージファイルの URL です。この URL は、Dreamweaver の "Configuration" フォルダを基準とする相対 URL です。この引数が null 値である場合は、アイコンのないメニューが表示されます。
- *doctypes* 引数 (オプション) は、特定のドキュメントタイプに対してのみこのメニューをアクティブにする場合に指定します。doctypes 引数は、ドキュメントタイプ ID をカンマで区切ったリストとして指定します。Dreamweaver のドキュメントタイプのリストについては、Dreamweaver の "Configuration/Documenttypes/MMDocumentTypes.xml" ファイルを参照してください。
- *casesensitive* 引数 (オプション) には、パターンで大文字と小文字を区別するかどうかを指定します。*casesensitive* 引数に指定できる値は、true または false のブール値です。この引数を省略した場合のデフォルト値は false です。*casesensitive* 引数が true の場合は、ユーザーが入力したテキストが pattern 属性に指定されたパターンと完全に一致するときにのみ、コードヒントメニューが表示されます。*casesensitive* が false の場合は、パターンが小文字で、入力されたテキストが大文字の場合でもメニューが表示されます。

戻り値

なし

例

ユーザーが "myRs" というレコードセットを作成した場合、以下のコードによって myRS のメニューが作成されます。

```
dw.codeHints.addMenu(
    "CodeHints_object_methods",          // オブジェクトメソッドが有効な場合にメニューは有効になります。
    "myRS.",                             // ユーザーが「myRS.」と入力すると、ポップアップメニューが表示されます。
    new Array("firstName", "lastName"),  // myRS 用のドロップダウンメニューの項目
    new Array("firstName", "lastName"),  // ドキュメントに実際に挿入するテキスト
    null,                                // このメニューにはアイコンなし
    "ASP_VB, ASP_JS");                   // ASP ドキュメントタイプ固有
```

dreamweaver.codeHints.addFunction()

対応バージョン

Dreamweaver MX

説明

新しい function タグを動的に定義します。同じパターンとドキュメントタイプが指定された function タグが既にある場合は、既存の function タグが置き換えられます。

引数

menuGroupId、*pattern*、*{doctypes}*、*{casesensitive}*

- *menuGroupId* 引数は、menugroup タグの ID ストリング属性です。
- *pattern* 引数は、新しい function タグの pattern 属性を指定するストリングです。
- *doctypes* 引数 (オプション) は、特定のドキュメントタイプに対してのみこの関数をアクティブにするときに指定します。doctypes 引数は、ドキュメントタイプ ID をカンマで区切ったリストとして指定します。Dreamweaver のドキュメントタイプのリストについては、Dreamweaver の "Configuration/Documenttypes/MMDocumentTypes.xml" ファイルを参照してください。
- *casesensitive* 引数 (オプション) には、パターンで大文字と小文字を区別するかどうかを指定します。casesensitive 引数に指定できる値は、true または false のブール値です。この引数を省略した場合のデフォルト値は false です。*casesensitive* 引数が true の場合は、ユーザーが入力したテキストが pattern 属性に指定されたパターンと完全に一致するときにのみ、コードヒントメニューが表示されます。*casesensitive* 引数が false の場合は、パターンが小文字で、入力されたテキストが大文字の場合でもメニューが表示されます。

戻り値

なし

例

次の dw.codeHints.addFunction() 関数の例では、コードヒントメニューの CodeHints_Object_Methods グループに関数名のパターン out.newLine() を追加し、この関数を JSP ドキュメントタイプにのみ有効にします。

```
dw.codeHints.addFunction(
    "CodeHints_Object_Methods",
    "out.newLine()",
    "JSP")
```

dreamweaver.codeHints.resetMenu()

対応バージョン

Dreamweaver MX

説明

指定された menu タグまたは function タグを、Dreamweaver が "CodeHints.xml" ファイルを読み取った直後の状態にリセットします。つまり、この関数を呼び出すと、それまでに呼び出した addMenu() 関数および addFunction() 関数の効果が消去されます。

引数

menuGroupId、*pattern*、*{doctypes}*

- *menuGroupId* 引数は、menugroup タグの ID ストリング属性です。
- *pattern* 引数は、リセットする menu タグまたは function タグの pattern 属性を指定するストリングです。
- *doctypes* 引数 (オプション) は、特定のドキュメントタイプに対してのみこのメニューをアクティブにする場合に指定します。doctypes 引数は、ドキュメントタイプ ID をカンマで区切ったリストとして指定します。Dreamweaver のドキュメントタイプのリストについては、Dreamweaver の "Configuration/Documenttypes/MMDocumentTypes.xml" ファイルを参照してください。

戻り値

なし

例

JavaScript コードを記述して、ユーザー定義セッション変数が表示されるコードヒントメニューを作成します。このコードでは、セッション変数のリストが変わるたびにメニューを更新する必要があります。新しいセッション変数のリストをメニューにロードする前に、古いリストを削除する必要があります。この関数を呼び出して、古いセッション変数を削除します。

dreamweaver.codeHints.showCodeHints()

対応バージョン

Dreamweaver MX

説明

この関数は、ユーザーが [編集]-[コードヒントの表示] を選択したときに、Dreamweaver によって呼び出されます。この関数は、コードビューの現在の選択範囲の位置にコードヒントメニューを表示します。

引数

なし

戻り値

なし

例

次の例では、コードビューでドキュメントの現在の挿入ポイントにコードヒントメニューを表示します。

```
dw.codeHints.showCodeHints()
```

dreamweaver.reloadCodeColoring()

説明

コードカラーリングファイルを Dreamweaver の "Configuration/Code Coloring" フォルダからリロードします。

引数

なし

戻り値

なし

例

```
dreamweaver.reloadCodeColoring()
```

検索 / 置換関数

検索 / 置換関数は、検索および置換操作を行います。これらの関数には、検索パターンの次の出現箇所を見つけるなどの基本的な機能があり、ユーザーの介入を必要とせずに複雑な置換操作を行うこともできます。

dreamweaver.findNext()

対応バージョン

Dreamweaver 3、Dreamweaver MX 2004 (Dreamweaver MX 2004 では機能変更)

説明

368 ページの dreamweaver.setUpFind() または 367 ページの dreamweaver.setUpComplexFind() によって指定された検索ストリング、あるいは [検索] ダイアログボックスでユーザーが指定した検索ストリングの、次の出現箇所を検索します。

引数

{bUseLastSetupSearch}

- *bUseLastSetupSearch* 引数 (オプション) はブール値です。*bUseLastSetupSearch* が `true` (引数を指定しなかった場合のデフォルト) である場合、この関数は、`dreamweaver.setupComplexFind()` 関数または `dreamweaver.setupComplexFindReplace()` 関数が前回呼び出されたときに指定されたパラメータを使用して、次を検索する処理を行います。*bUseLastSetupSearch* が `false` に設定された場合は、前回設定された検索を無視し、ドキュメント内で現在選択されているテキストの次の出現箇所を検索します。

戻り値

なし

イネーブラ

詳細については、419 ページの dreamweaver.canFindNext() を参照してください。

dreamweaver.replace()

対応バージョン

Dreamweaver 3

説明

369 ページの dreamweaver.setUpFindReplace() または 367 ページの dreamweaver.setUpComplexFindReplace() によって指定された検索条件、あるいは [置換] ダイアログボックスでユーザーが指定した検索条件と現在の選択範囲が一致するかどうかを調べ、一致する場合は選択範囲を検索リクエストで指定された置換テキストに置き換えます。

引数

なし

戻り値

なし

dreamweaver.replaceAll()

対応バージョン

Dreamweaver 3

説明

369 ページの dreamweaver.setUpFindReplace() または 367 ページの dreamweaver.setUpComplexFindReplace() によって指定された検索条件、あるいは [置換] ダイアログボックスでユーザーが指定した検索条件に一致する現在のドキュメントの各部分を、指定された置換内容に置き換えます。

引数

なし

戻り値

なし

dreamweaver.setUpComplexFind()

対応バージョン

Dreamweaver 3

説明

テキストまたはタグの高度な検索の準備として、指定された XML クエリーを読み込みます。

引数

xmlQueryString

- *xmlQueryString* 引数には、dwquery で始まり /dwquery で終わる XML コードのストリングを指定します。正しいフォーマットのストリングを取得するには、[検索] ダイアログボックスでクエリーを設定して [クエリーの保存] ボタンをクリックし、テキストエディタでクエリーファイルを開いて dwquery タグの開始点から /dwquery タグの終了点までの全体をコピーします。

注意 : クエリーでは、バックスラッシュ文字 (\) などの一部の特殊文字をエスケープする必要があります。そのため、クエリーでバックスラッシュを使用するには、「\\」と記述します。

戻り値

なし

例

次の例では、最初の行でタグ検索を設定して検索のスコープを現在のドキュメントに指定し、2 行目で検索操作を実行します。

```
dreamweaver.setUpComplexFind('<dwquery><queryparams matchcase="false" ¬
ignorewhitespace="true" useregexp="false"/><find>¬
<qtag qname="a"><qattribute qname="href" qcompare="=" qvalue="#">¬
    </qattribute><qattribute qname="onMouseOut" qcompare="=" qvalue="" qnegate="true">¬
    </qattribute></qtag></find></dwquery>');
dw.findNext();
```

dreamweaver.setUpComplexFindReplace()

対応バージョン

Dreamweaver 3

説明

テキストまたはタグの高度な検索の準備として、指定された XML クエリーを読み込みます。

引数

xmlQueryString

- *xmlQueryString* 引数には、dwquery タグで始まり /dwquery タグで終わる XML コードのストリングを指定します。正しいフォーマットのストリングを取得するには、[検索] ダイアログボックスでクエリーを設定して [クエリーの保存] ボタンをクリックし、テキストエディタでクエリーファイルを開いて dwquery タグの開始点から /dwquery タグの終了点までの全体をコピーします。

注意 : クエリーでは、バックスラッシュ文字 (\) などの一部の特殊文字をエスケープする必要があります。そのため、クエリーでバックスラッシュを使用するには、「\\」と記述します。

戻り値

なし

例

次の例では、最初のステートメントでタグ検索を設定して検索のスコープを 4 つのファイルに指定し、2 番目のステートメントで検索と置換の操作を実行します。

```
dreamweaver.setUpComplexFindReplace('<dwquery><queryparams ¬
matchcase="false" ignorewhitespace="true" useregexp="false"/>¬
<find><qtag qname="a"><qattribute qname="href" qcompare="=" qvalue="#">¬
</qattribute><qattribute qname="onMouseOut" ¬qcompare="=" qvalue="" qnegate="true">¬
</qattribute></qtag></find><replace action="setAttribute" param1="onMouseOut" ¬
param2="this.style.color='#000000';this.style.¬
fontWeight='normal'"/></dwquery>');
dw.replaceAll();
```

dreamweaver.setUpFind()

対応バージョン

Dreamweaver 3

説明

テキスト検索または HTML ソース検索の準備として、この後に行う `dreamweaver.findNext()` 操作の検索パラメータを定義します。

引数

searchObject

searchObject 引数には、以下のプロパティを定義できるオブジェクトを指定します。

- *searchString* は検索対象のテキストです。
- *searchSource* プロパティは、HTML ソースを検索するかどうかを示すブール値です。
- *{matchCase}* プロパティ (オプション) は、大文字と小文字を区別して検索するかどうかを示すブール値です。このプロパティを明示的に設定しない場合のデフォルト値は `false` です。
- *{ignoreWhitespace}* プロパティ (オプション) は、ホワイトスペースの違いを無視するかどうかを示すブール値です。*ignoreWhitespace* プロパティを省略したときのデフォルト値は、*useRegularExpressions* プロパティの値が `true` の場合は `false`、*useRegularExpressions* プロパティが `false` の場合は `true` です。
- *{useRegularExpressions}* プロパティは、`searchString` プロパティで正規表現を使用するかどうかを示すブール値です。このプロパティを明示的に設定しない場合のデフォルト値は `false` です。

戻り値

なし

例

次のコード例に、*searchObject* オブジェクトを作成する 3 とおりの方法を示します。

```
var searchParams;
searchParams.searchString = 'bgcolor="#FFCCFF"';
searchParams.searchSource = true;
dreamweaver.setUpFind(searchParams);
```

```
var searchParams = {searchString: 'bgcolor="#FFCCFF"', searchSource: true};
dreamweaver.setUpFind(searchParams);
```

```
dreamweaver.setUpFind({searchString: 'bgcolor="#FFCCFF"', searchSource: ¬true});
```

dreamweaver.setUpFindReplace()

対応バージョン

Dreamweaver 3

説明

テキスト検索または HTML ソース検索の準備として、この後に行う `dreamweaver.replace()` または `dreamweaver.replaceAll()` 操作の検索パラメータおよびスコープを定義します。

引数

searchObject

searchObject 引数には、以下のプロパティを定義できるオブジェクトを指定します。

- *searchString* プロパティは検索対象のテキストです。
- *replaceString* プロパティは、選択範囲を置き換えるテキストです。
- *searchSource* プロパティは、HTML ソースを検索するかどうかを示すブール値です。
- *{matchCase}* プロパティ (オプション) は、大文字と小文字を区別して検索するかどうかを示すブール値です。このプロパティを明示的に設定しない場合のデフォルト値は `false` です。
- *{ignoreWhitespace}* プロパティ (オプション) は、ホワイトスペースの違いを無視するかどうかを示すブール値です。*ignoreWhitespace* プロパティを省略したときのデフォルト値は、*useRegularExpressions* プロパティの値が `true` の場合は `false`、*useRegularExpressions* プロパティが `false` の場合は `true` です。
- *{useRegularExpressions}* プロパティは、`searchString` プロパティで正規表現を使用するかどうかを示すブール値です。このプロパティを明示的に設定しない場合のデフォルト値は `false` です。

戻り値

なし

例

次のコード例に、*searchObject* オブジェクトを作成する 3 とおりの方法を示します。

```
var searchParams;
searchParams.searchString = 'bgcolor="#FFCCFF"';
searchParams.replaceString = 'bgcolor="#CCFFCC"';
searchParams.searchSource = true;
dreamweaver.setUpFindReplace(searchParams);

var searchParams = {searchString: 'bgcolor="#FFCCFF"', replaceString: 'bgcolor="#CCFFCC"',
    searchSource: true};
dreamweaver.setUpFindReplace(searchParams);

dreamweaver.setUpFindReplace({searchString: 'bgcolor="#FFCCFF"',
replaceString: 'bgcolor="#CCFFCC"', searchSource: true});
```

dreamweaver.showFindDialog()

対応バージョン

Dreamweaver 3

説明

[検索] ダイアログボックスを開きます。

引数
なし

戻り値
なし

イネーブラ
詳細については、423 ページの dreamweaver.canShowFindDialog() を参照してください。

dreamweaver.showFindReplaceDialog()

対応バージョン
Dreamweaver 3

説明
[置換] ダイアログボックスを開きます。

引数
なし

戻り値
なし

イネーブラ
詳細については、423 ページの dreamweaver.canShowFindDialog() を参照してください。

一般編集関数

一般編集関数は、ドキュメントウィンドウの中で使用します。これらの関数を使用して、テキスト、HTML、およびオブジェクトの挿入、フォントおよび文字マークアップの適用、変更、および削除、タグおよび属性の修正などを行います。

dom.applyCharacterMarkup()

対応バージョン
Dreamweaver 3

説明
選択範囲に、指定されたタイプの文字マークアップを適用します。選択範囲が挿入ポイントである場合は、この後入力されるテキストに、指定された文字マークアップを適用します。

引数
tagName

- *tagName* 引数には、文字マークアップに関連付けられているタグ名を指定します。"b"、"cite"、"code"、"dfn"、"em"、"i"、"kbd"、"samp"、"s"、"strong"、"tt"、"u"、または "var" のいずれかを指定する必要があります。

戻り値
なし

dom.applyFontMarkup()

対応バージョン
Dreamweaver 3

説明
FONT タグおよび指定された属性と値を、現在の選択範囲に適用します。

引数
attribute、*value*

- *attribute* 引数には、"face"、"size"、"color" のいずれかを指定します。
- *value* には、属性に割り当てる値を指定します。たとえば、"Arial,Helvetica,sans-serif"、"5"、"#FF0000" などです。

戻り値
なし

dom.deleteSelection()

対応バージョン
Dreamweaver 3

説明
ドキュメントで選択範囲を削除します。

引数
なし

戻り値
なし

dom.editAttribute()

対応バージョン
Dreamweaver 3

説明
指定されたドキュメント属性を編集するための適切なインターフェイスを表示します。通常、このインターフェイスはダイアログボックスです。この関数はアクティブなドキュメントのみに適用されます。

引数
attribute

- *attribute* は、編集するタグ属性を指定するストリングです。

戻り値
なし

dom.exitBlock()

対応バージョン
Dreamweaver 3

説明
挿入ポイントをすべてのブロックエレメントの外側に置いたまま、現在の段落または見出しブロックを終了します。

引数
なし

戻り値
なし

dom.getCharSet()

対応バージョン
Dreamweaver 4

説明
ドキュメントの meta タグ内の `charset` 属性を返します。

引数
なし

戻り値
ドキュメントのエンコード ID。たとえば、Latin1 ドキュメントの場合、この関数は `iso-8859-1` を返します。

dom.getFontMarkup()

対応バージョン
Dreamweaver 3

説明
現在の選択範囲に適用される `FONT` タグの指定された属性の値を取得します。

引数
`attribute`

- *`attribute`* 引数には、`"face"`、`"size"`、`"color"` のいずれかを指定します。

戻り値
指定された属性の値を含むストリング。属性が設定されていない場合は空白のストリング。

dom.getLineFromOffset()

対応バージョン
Dreamweaver MX

説明
ファイルのテキスト (HTML または JavaScript コード) 内で、特定の文字オフセットの行番号を検索します。

引数

offset

- *offset* 引数には、ファイルの先頭を基準とした文字の位置を表す整数を指定します。

戻り値

ドキュメント内の行番号を表す整数。

dom.getLinkHref()

対応バージョン

Dreamweaver 3

説明

現在の選択範囲を囲むリンクを取得します。この関数の機能は、リンクが見つかるまで現在のノードの親、その親という順番でループし、見つかったリンクに対して `getAttribute('HREF')` を呼び出すのと同じです。

引数

なし

戻り値

リンク先のファイル名を file:// で始まる URL 形式で表記したストリング。

dom.getLinkTarget()

対応バージョン

Dreamweaver 3

説明

現在の選択範囲を囲むリンクのターゲットを取得します。この関数の機能は、リンクが見つかるまで現在のノードの親、その親という順番でループし、見つかったリンクに対して `getAttribute('TARGET')` 関数を呼び出すのと同じです。

引数

なし

戻り値

リンクの `TARGET` 属性の値を含むストリング。ターゲットが指定されていない場合は空白のストリング。

dom.getListTag()

対応バージョン

Dreamweaver 3

説明

選択されたリストのスタイルを取得します。

引数

なし

戻り値

リストに関連付けられたタグを含むストリング (`"ul"`、`"ol"`、または `"dl"`)。リストに関連付けられたタグがない場合は空白のストリング。この値は常に小文字で返されます。

dom.getTextAlignment()

対応バージョン

Dreamweaver 3

説明

選択範囲を含むブロックの整列条件を取得します。

引数

なし

戻り値

ブロックに関連付けられているタグの `ALIGN` 属性の値を含むストリング。タグの `ALIGN` 属性が設定されていない場合は空白のストリング。この値は常に小文字で返されます。

dom.getTextFormat()

対応バージョン

Dreamweaver 3

説明

選択されたテキストのブロックフォーマットを取得します。

引数

なし

戻り値

テキストに関連付けられたブロックタグを含むストリング (たとえば `"p"`、`"h1"`、`"pre"` など)。選択範囲にブロックタグが関連付けられていない場合は空白のストリング。この値は常に小文字で返されます。

dom.hasCharacterMarkup()

対応バージョン

Dreamweaver 3

説明

選択範囲に、指定された文字マークアップが存在するかどうかをチェックします。

引数

markupTagName

- *markupTagName* 引数には、チェックするタグの名前を指定します。`"b"`、`"cite"`、`"code"`、`"dfn"`、`"em"`、`"i"`、`"kbd"`、`"samp"`、`"s"`、`"strong"`、`"tt"`、`"u"`、または `"var"` のいずれかを指定する必要があります。

戻り値

指定された文字マークアップが選択範囲全体にあるかどうかを示すブール値。指定されたマークアップが選択範囲の一部だけにある場合は、`false` 値を返します。

dom.indent()

対応バージョン
Dreamweaver 3

説明
BLOCKQUOTE タグを使用して、選択範囲をインデントします。選択範囲がリスト項目の場合は、その項目をネストされたリストに変換することによって選択範囲をインデントします。このネストされたリストは、外側のリストと同じタイプのリストで、1 つの項目 (元の選択範囲) から構成されます。

引数
なし

戻り値
なし

dom.insertHTML()

対応バージョン
Dreamweaver 3

説明
ドキュメントの現在の挿入ポイントに HTML コンテンツを挿入します。

引数
contentToInsert、*{bReplaceCurrentSelection}*

- *contentToInsert* 引数には、挿入するコンテンツを指定します。
- *bReplaceCurrentSelection* 引数 (オプション) には、このコンテンツで現在の選択範囲を置換するかどうかを示すブール値を指定します。*bReplaceCurrentSelection* 引数の値が true の場合は、現在の選択範囲がコンテンツに置き換えられます。値が false の場合、コンテンツは現在の選択範囲の後に挿入されます。

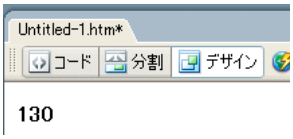
戻り値
なし

例
次のコードは、現在のドキュメントに 130 という HTML ストリングを挿入します。

```
var theDOM = dw.getDocumentDOM();

theDOM.insertHTML('<b>130</b>');
```

この結果は、次のようにドキュメントウィンドウに表示されます。



dom.insertObject()

対応バージョン
Dreamweaver 3

説明

指定されたオブジェクトを挿入します。必要に応じて、ユーザーにパラメータの指定を要求します。

引数

objectName

- *objectName* 引数には、"Configuration¥Objects" フォルダ内のオブジェクトの名前を指定します。

戻り値

なし

例

dom.insertObject('Button') 関数を呼び出すと、アクティブなドキュメントの現在の選択範囲の後にフォームボタンが挿入されます。何も選択していない場合は、現在の挿入ポイントの位置にボタンが挿入されます。

注意 : オブジェクトファイルはいくつかのフォルダに分けて保存することができますが、必ず一意のファイル名を付けてください。"*Button.htm*" というファイルが "*Forms*" フォルダと "*MyObjects*" フォルダの両方に保存されていると、*Dreamweaver* はこの *2* つを区別できません。

dom.insertText()

対応バージョン

Dreamweaver 3

説明

ドキュメントの現在の挿入ポイントの位置にテキストコンテンツを挿入します。

引数

contentToInsert、*{bReplaceCurrentSelection}*

- *contentToInsert* 引数には、挿入するコンテンツを指定します。
- *bReplaceCurrentSelection* 引数 (オプション) には、このコンテンツで現在の選択範囲を置換するかどうかを示すブール値を指定します。*bReplaceCurrentSelection* 引数の値が true の場合は、現在の選択範囲がコンテンツに置き換えられます。値が false の場合、コンテンツは現在の選択範囲の後に挿入されます。

戻り値

なし

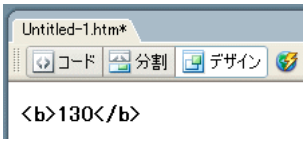
例

次のコードは、現在のドキュメントに 130 というテキストを挿入します。

```
var theDOM = dreamweaver.getDocumentDOM();

theDOM.insertText('<b>130</b>');
```

この結果は、次のようにドキュメントウィンドウに表示されます。



dom.newBlock()

対応バージョン
Dreamweaver 3

説明
現在の選択範囲を含むブロックと同じタグおよび属性を持つ新しいブロックを作成します。ポインタがすべてのブロックの外側にある場合は、新しい段落を作成します。

引数
なし

戻り値
なし

例
現在の選択範囲が中央揃えされた段落の中にある場合、`dom.newBlock()` 関数を呼び出すと、現在の段落の後に `<p align="center">` が挿入されます。

dom.notifyFlashObjectChanged()

対応バージョン
Dreamweaver 4

説明
現在の Flash オブジェクトファイルが変更されていることを Dreamweaver に通知します。Dreamweaver では、プレビュー表示が更新されます。このとき、必要に応じてプレビュー表示のサイズが変更されますが、変更前のサイズの幅と高さの比率は維持されます。たとえば、ユーザーが Flash テキストのプロパティを [コマンド] ダイアログボックスで変更した場合に、この機能を使用してレイアウトビューのテキストが更新されます。

引数
なし

戻り値
なし

dom.outdent()

対応バージョン
Dreamweaver 3

説明
選択範囲のインデントを解除します。

引数
なし

戻り値
なし

dom.removeCharacterMarkup()

対応バージョン
Dreamweaver 3

説明
選択範囲から、指定されたタイプの文字マークアップを削除します。

引数
tagName

- *tagName* 引数には、文字マークアップに関連付けられているタグ名を指定します。"b"、"cite"、"code"、"dfn"、"em"、"i"、"kbd"、"samp"、"s"、"strong"、"tt"、"u"、または "var" のいずれかを指定する必要があります。

戻り値
なし

dom.removeFontMarkup()

対応バージョン
Dreamweaver 3

説明
FONT タグから、指定された属性とその値を削除します。属性を削除すると FONT タグだけが残る場合は、FONT タグも削除します。

引数
attribute

- *attribute* 引数には、"face"、"size"、"color" のいずれかを指定します。

戻り値
なし

dom.removeLink()

対応バージョン
Dreamweaver 3

説明
選択範囲からハイパーテキストリンクを削除します。

引数
なし

戻り値
なし

dom.resizeSelection()

対応バージョン
Dreamweaver 3

説明
選択されたオブジェクトを指定のサイズに変更します。

引数
newWidth、*newHeight*

- *newWidth* 引数には、選択したオブジェクトに設定する新しい幅を指定します。
- *newHeight* 引数には、選択したオブジェクトに設定する新しい高さを指定します。

戻り値
なし

dom.setAttributeWithErrorChecking()

対応バージョン
Dreamweaver 3

説明
現在の選択範囲の指定された属性に、指定された値を設定します。値のタイプが無効な場合や有効範囲外の場合は、ユーザーに再入力を要求します。この関数はアクティブなドキュメントのみに適用されます。

引数
attribute、*value*

- *attribute* 引数には、現在の選択範囲に設定する属性を指定します。
- *value* 引数には、属性に設定する値を指定します。

戻り値
なし

dom.setLinkHref()

対応バージョン
Dreamweaver 3

説明
選択範囲をハイパーリンクにします。または、現在の選択範囲を囲む HREF タグの URL 値を変更します。

引数
linkHREF

- *linkHREF* 引数には、リンクを構成する URL (ドキュメント相対パス、ルート相対パス、または絶対 URL) を指定します。この引数を省略すると、[HTML ファイルの選択] ダイアログボックスが表示されます。

戻り値
なし

イネーブラ
詳細については、414 ページの dom.canSetLinkHref() を参照してください。

dom.setLinkTarget()

対応バージョン
Dreamweaver 3

説明
現在の選択範囲を囲むリンクのターゲットを設定します。この関数の機能は、リンクが見つかるまで現在のノードの親、その親という順番でループし、見つかったリンクに対して `setAttribute('TARGET')` 関数を呼び出すのと同じです。

引数
`{linkTarget}`

- *`linkTarget`* 引数 (オプション) には、フレーム名、ウィンドウ名、または予約されているターゲットの 1 つ (`"_self"`、`"_parent"`、`"_top"`、または `"_blank"`) を表すストリングを指定します。この引数を省略すると、[ターゲットの設定] ダイアログボックスが表示されます。

戻り値
なし

dom.setListBoxKind()

対応バージョン
Dreamweaver 3

説明
選択された `SELECT` メニューの種類を変更します。

引数
`kind`

- *`kind`* 引数には、`"menu"` または `"list box"` のいずれかを指定する必要があります。

戻り値
なし

dom.showListPropertiesDialog()

対応バージョン
Dreamweaver 3

説明
[リストプロパティ] ダイアログボックスを開きます。

引数
なし

戻り値
なし

イネーブラ
詳細については、414 ページの dom.canShowListPropertiesDialog() を参照してください。

dom.setListTag()

対応バージョン
Dreamweaver 3

説明
選択されたリストのスタイルを設定します。

引数
listTag

- *listTag* 引数には、リストに関連付けられたタグを指定します。"ol"、"ul"、"dl"、または空白のストリングのいずれかを指定する必要があります。

戻り値
なし

dom.setTextAlignment()

対応バージョン
Dreamweaver 3

説明
選択範囲を含むブロックの ALIGN 属性に、指定された値を設定します。

引数
alignValue

- *alignValue* 引数には、"left"、"center"、または "right" のいずれかを指定する必要があります。

戻り値
なし

dom.setTextFieldKind()

対応バージョン
Dreamweaver 3

説明
選択されたテキストフィールドのフォーマットを設定します。

引数
fieldType

- *fieldType* 引数には、"input"、"textarea"、または "password" のいずれかを指定する必要があります。

戻り値
なし

dom.setTextFormat()

対応バージョン
Dreamweaver 4

説明
選択されたテキストのブロックフォーマットを設定します。

引数
blockFormat

- *blockFormat* 引数は、"" (フォーマットなし)、"p"、"h1"、"h2"、"h3"、"h4"、"h5"、"h6"、"pre" のいずれかのフォーマットを指定するストリングです。

戻り値
なし

dom.showFontColorDialog()

対応バージョン
Dreamweaver 3

説明
[カラーピッカー] ダイアログボックスを開きます。

引数
なし

戻り値
なし

dreamweaver.deleteSelection()

対応バージョン
Dreamweaver 3

説明
アクティブなドキュメントまたは [サイト] パネル内の選択範囲を削除します。Macintosh の場合は、ダイアログボックスまたはフローティングパネルのフォーカスのあるテキストボックス内の選択範囲を削除します。

引数
なし

戻り値
なし

イネーブラ
詳細については、418 ページの dreamweaver.canDeleteSelection() を参照してください。

dreamweaver.editFontList()

対応バージョン
Dreamweaver 3

説明
[フォントリストの編集] ダイアログボックスを開きます。

引数
なし

戻り値
なし

dreamweaver.getFontList()

対応バージョン
Dreamweaver 3

説明
テキストのプロパティインスペクタおよび [スタイル定義] ダイアログボックスに表示されるすべてのフォントグループのリストを取得します。

引数
なし

戻り値
フォントリストの各項目を表すストリングの配列。

例
Dreamweaver のデフォルトのインストールでは、`dreamweaver.getFontList()` 関数を呼び出すと次の項目を含む配列が返されます。

- `"Arial, Helvetica, sans-serif"`
- `"Times New Roman, Times, serif"`
- `"Courier New, Courier, mono"`
- `"Georgia, Times New Roman, Times, serif"`
- `"Verdana, Arial, Helvetica, sans-serif"`

dreamweaver.getFontStyles()

対応バージョン
Dreamweaver 4

説明
指定された TrueType フォントがサポートするスタイルを返します。

引数
`fontName`

- *`fontName`* 引数には、フォントの名前を含むストリングを指定します。

戻り値
フォントがサポートするスタイルを示す 3 つのブール値の配列。最初のブール値は、フォントがボールドをサポートするかどうかを示し、2 番目のブール値は、フォントがイタリックをサポートするかどうかを示し、3 番目のブール値は、フォントがボールドとイタリックの両方をサポートするかどうかを示します。

dreamweaver.getKeyState()

対応バージョン

Dreamweaver 3

説明

指定された修飾キーが押されているかどうかを判別します。

引数

key

- *key* 引数には、"Cmd"、"Ctrl"、"Alt"、または "Shift" のいずれかを指定する必要があります。Windows では "Cmd" と "Ctrl" は Ctrl キーとして解釈されます。Macintosh では "Alt" は Option キーとして解釈されます。

戻り値

キーが押されているかどうかを示すブール値。

例

次のコードは、ある操作を実行する前に、Shift キーと Ctrl キー (Windows) または Shift キーと Command キー (Macintosh) が押されているどうかを調べます。

```
if (dw.getKeyState("Shift") && dw.getKeyState("Cmd")){
    // コードを実行します。
}
```

dreamweaver.getNaturalSize()

対応バージョン

Dreamweaver 4

説明

グラフィックオブジェクトの幅と高さを返します。

引数

url

- *url* 引数には、サイズを求めるグラフィックオブジェクトの URL を指定します。このグラフィックオブジェクトは、Dreamweaver がサポートするオブジェクト (GIF、JPEG、PNG、Flash、および Shockwave) である必要があります。getNaturalSize() 関数の引数として指定する URL は、ローカルファイルを指す絶対 URL です。相対 URL を指定することはできません。

戻り値

2 つの整数の配列。最初の整数はオブジェクトの幅を表し、2 番目の整数は高さを表します。

dreamweaver.getSystemFontList()

対応バージョン

Dreamweaver 4

説明

システムのフォントリストを返します。この関数では、すべてのフォントを取得することも、TrueType フォントだけを取得することもできます。これらのフォントは Flash Text オブジェクトに必要です。

引数

fontTypes

- *fontTypes* 引数には、"all" または "TrueType" のいずれかを含むストリングを指定します。

戻り値

すべてのフォント名を格納したストリングの配列。フォントがまったく見つからない場合は、null 値を返します。

プリント関数

プリント関数を使用すると、ユーザーはコードビューからコードを印刷することができます。

dreamweaver.PrintCode()

対応バージョン

Dreamweaver MX

説明

Windows では、この関数はコードビューのすべてのコードまたは選択した一部のコードを印刷します。Macintosh では、すべてのコードまたはコードのページ範囲を印刷します。

引数

showPrintDialog、*document*

- *showPrintDialog* 引数には、true または false を指定します。Windows では、この引数を true に設定して dreamweaver.PrintCode() 関数を実行すると、[印刷] ダイアログボックスが表示され、すべてのテキストを印刷するか、または一部のテキストを印刷するかを尋ねられます。Macintosh で dreamweaver.PrintCode() 関数を実行すると、[印刷] ダイアログボックスが表示され、すべてのテキストを印刷するか、またはページ範囲を印刷するかを尋ねられます。

引数を false に設定すると、dreamweaver.PrintCode() はユーザーの以前の選択範囲を使用します。デフォルト値は true です。

- *document* 引数には、印刷するドキュメントの DOM を指定します。ドキュメントの DOM を取得する方法の詳細については、219 ページの dreamweaver.getDocumentDOM() を参照してください。

戻り値

コードを印刷できる場合は true、それ以外の場合は false のブール値。

例

次の例では、dw.PrintCode() を呼び出して、ユーザーのドキュメントに対する [印刷] ダイアログボックスを表示します。関数から false が返された場合は、印刷リクエストを実行できないことをユーザーに知らせる警告メッセージを表示します。

```
var theDOM = dreamweaver.getDocumentDOM("document");
if(!dreamweaver.PrintCode(true, theDOM))
{
    alert("Unable to execute your print request!");
}
```

クイックタグ編集関数

クイックタグ編集関数を使用して、現在の選択範囲内またはその周囲にあるタグをナビゲートします。これらの関数は、階層内のタグを削除したり、新規タグ内にある選択範囲をタグで囲んだり、ユーザーがタグの特定の属性を編集できるようにクイックタグ編集を表示したりします。

dom.selectChild()

対応バージョン

Dreamweaver 3

説明

現在の選択範囲の子を選択します。この関数を呼び出すと、ドキュメントウィンドウの一番下にあるタグセレクタで、右隣りのタグを選択するのと同じ操作が行われます。

引数

なし

戻り値

なし

dom.selectParent()

対応バージョン

Dreamweaver 3

説明

現在の選択範囲の親を選択します。この関数を呼び出すと、ドキュメントウィンドウの一番下にあるタグセレクタで、左隣りのタグを選択した場合と同じ操作が行われます。

引数

なし

戻り値

なし

dom.stripTag()

対応バージョン

Dreamweaver 3

説明

現在の選択範囲の周囲からタグを削除します。コンテンツは維持されます。選択範囲にタグがない場合、または複数のタグがある場合は、エラーが表示されます。

引数

なし

戻り値

なし

dom.wrapTag()

対応バージョン
Dreamweaver 3

説明
現在の選択範囲を指定されたタグで囲みます。選択範囲が不平衡である場合は、Dreamweaver からエラーが返されます。

引数
startTag

- *startTag* 引数には、開始タグに関連付けられているソースを指定します。

戻り値
なし

例
次のコードは、現在の選択範囲をリンクで囲みます。

```
var theDOM = dw.getDocumentDOM();
var theSel = theDOM.getSelectedNode();
if (theSel.nodeType == Node.TEXT_NODE){
    theDOM.wrapTag('<a href="foo.html">');
}
```

dreamweaver.showQuickTagEditor()

対応バージョン
Dreamweaver 3

説明
現在の選択範囲のクイックタグ編集を表示します。

引数
{nearWhat}、*{mode}*

- オプションの *nearWhat* 引数を指定する場合は、"selection" か "tag selector" のいずれかを指定する必要があります。この引数を省略した場合のデフォルト値は "selection" です。
- オプションの *mode* 引数を指定する場合は、"default"、"wrap"、"insert"、"edit" のいずれかを指定する必要があります。*mode* に "default" を指定するか、または省略した場合、Dreamweaver はヒューリスティクスを使用して現在の選択範囲に適用するモードを決定します。*nearWhat* が "tag selector" の場合、*mode* 引数は無視されます。

戻り値
なし

コードビュー関数

コードビュー関数には、ドキュメントソースコードの編集に関連する操作が含まれています。これらを実行すると、デザインビューも影響されます。この項の関数を使用すると、分割ドキュメントビューまたは [コードインスペクタ] ウィンドウ内のコードビューに、ナビゲーションのコントロールを追加することができます。

dom.formatRange()

対応バージョン
Dreamweaver MX

説明
[環境設定] の [コードフォーマット] ダイアログボックスの設定に従って、コードビューで指定された文字範囲に Dreamweaver 自動シンタックスフォーマットを適用します。

引数
startOffset、*endOffset*

- *startOffset* 引数には、指定範囲の先頭をドキュメントの先頭からのオフセットで示す整数を指定します。
- *endOffset* 引数には、指定範囲の末尾をドキュメントの先頭からのオフセットで示す整数を指定します。

戻り値
なし

dom.formatSelection()

対応バージョン
Dreamweaver MX

説明
[環境設定] の [コードフォーマット] ダイアログボックスの設定に従って、選択したコンテンツに Dreamweaver 自動シンタックスフォーマットを適用します。これは、[コマンド]-[ソースフォーマットを選択範囲に適用] オプションを選択するのと同じです。

引数
なし

戻り値
なし

dom.getShowNoscript()

対応バージョン
Dreamweaver MX

説明
noscript コンテンツオプション ([表示]-[ノースクリプトコンテンツ] メニューオプションから選択) の現在の状態を取得します。デフォルトでは、noscript タグは、ブラウザでレンダリング可能またはレンダリング不能 (どちらかを指定します) なページスクリプトコンテンツを識別します。

引数
なし

戻り値
noscript タグコンテンツが現在レンダリングされている場合は true、それ以外の場合は false のブール値。

dom.getAutoValidationCount()

対応バージョン

Dreamweaver MX 2004

説明

ドキュメントに対して最後に実行された自動検証 (インライン検証) でのエラー、警告、および情報メッセージの数を取得します。現在、自動検証で実行されるのはターゲットブラウザチェックだけです。詳細については、227 ページの dom.runValidation() を参照してください。

注意 : この関数は、ドキュメントの [結果] ウィンドウに現在表示されている結果のみを返します。最新の数を取得するには、この関数の前に *dom.runValidation()* を呼び出してください。

引数

なし

戻り値

次のプロパティを持つオブジェクト。

- エラーの数を表す `numError` プロパティ
- 警告の数を表す `numWarning` プロパティ
- 情報メッセージの数を表す `numInfo` プロパティ

例

```
theDom = dw.getDocumentDOM();
theDom.runValidation();
theDom.getAutoValidationCount();
```

dom.isDesignViewUpdated()

対応バージョン

Dreamweaver 4

説明

デザインビューおよびテキストビューのコンテンツが、ドキュメントの状態が有効であることを必要とする Dreamweaver の操作に対して同期しているかどうかを判別します。

引数

なし

戻り値

デザインビュー (WYSIWYG) がテキストビューのテキストと同期している場合は `true`、それ以外の場合は `false` のブール値。

dom.isSelectionValid()

対応バージョン

Dreamweaver 4

説明

選択範囲が有効かどうかを判別します。つまり、現在、選択範囲とデザインビューが同期しているか、または操作を実行する前に選択範囲を移動する必要があるかどうかを判別します。

引数
なし

戻り値
現在の選択範囲が有効なコードの一部である場合は true、ドキュメントが同期していない場合は、選択範囲が更新されていないので false のブール値。

dom.setShowNoscript()

対応バージョン
Dreamweaver MX

説明
noscript コンテンツのオプションをオンまたはオフに切り替えます。これは、[表示]-[ノースクリプトコンテンツ] オプションを選択した場合と同じです。デフォルトでは、noscript タグは、ブラウザでレンダリング可能またはレンダリング不能 (どちらかを指定します) なページスクリプトコンテンツを識別します。

引数
{bShowNoscript}

- *bShowNoscript* 引数 (オプション) は、noscript タグコンテンツをレンダリングするかどうかを示すブール値です。noscript タグコンテンツをレンダリングする場合は true、それ以外の場合は false を指定します。

戻り値
なし

dom.source.arrowDown()

対応バージョン
Dreamweaver 4

説明
コードビュードキュメントで挿入ポイントを 1 行ずつ下に移動します。コンテンツが既に選択されている場合、この関数は選択範囲を 1 行ずつ拡張します。

引数
{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを移動する行数を指定します。*nTimes* を指定しない場合のデフォルト値は 1 です。
- *bShiftIsDown* 引数 (オプション) には、コンテンツが選択されているかどうかを示すブール値を指定します。*bShiftIsDown* が true の場合、コンテンツが選択されています。

戻り値
なし

dom.source.arrowLeft()

対応バージョン
Dreamweaver 4

説明

コードビューの現在の行で挿入ポイントを左に移動します。コンテンツが既に選択されている場合、この関数は選択範囲を左に拡張します。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを移動する文字数を指定します。*nTimes* を指定しない場合のデフォルト値は 1 です。
- *bShiftIsDown* 引数 (オプション) には、コンテンツが選択されているかどうかを示すブール値を指定します。*bShiftIsDown* が `true` の場合、コンテンツが選択されています。

戻り値

なし

dom.source.arrowRight()

対応バージョン

Dreamweaver 4

説明

コードビューの現在の行で挿入ポイントを右に移動します。コンテンツが既に選択されている場合、この関数は選択範囲を右に拡張します。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを移動する文字数を指定します。*nTimes* を指定しない場合のデフォルト値は 1 です。
- *bShiftIsDown* 引数 (オプション) には、コンテンツが選択されているかどうかを示すブール値を指定します。*bShiftIsDown* が `true` の場合は、コンテンツが選択されています。それ以外の場合、コンテンツは選択されていません。

戻り値

なし

dom.source.arrowUp()

対応バージョン

Dreamweaver 4

説明

コードビュードキュメントで挿入ポイントを 1 行ずつ上に移動します。コンテンツが既に選択されている場合、この関数は選択範囲を 1 行ずつ拡張します。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数には、挿入ポイントを移動する行数を指定します。*nTimes* を指定しない場合のデフォルト値は 1 です。
- *bShiftIsDown* 引数には、コンテンツが選択されているかどうかを示すブール値を指定します。*bShiftIsDown* が `true` の場合、コンテンツが選択されています。

戻り値

なし

dom.source.balanceBracesTextview()

対応バージョン

Dreamweaver 4

説明

この関数は、カッコの均衡を使用可能にするコードビューの拡張機能です。`dom.source.balanceBracesTextview()` を呼び出すことにより、現在ハイライト表示されている選択範囲または挿入ポイントを拡張して、ステートメントを囲む先頭のカッコから、それに対応する末尾のカッコまでが含まれるようにし、`[]`、`{}`、および `()` の均衡を取ることができます。続いて実行される呼び出しは、より下位レベルの句読点のネストを使用して選択範囲を拡張します。

引数

なし

戻り値

なし

dom.source.endOfDocument()

対応バージョン

Dreamweaver 4

説明

挿入ポイントを現在のコードビュードキュメントの末尾に配置します。コンテンツが既に選択されている場合、この関数は選択範囲をドキュメントの末尾まで拡張します。

引数

`bShiftIsDown`

- *`bShiftIsDown`* 引数には、コンテンツが選択されているかどうかを示すブール値を指定します。*`bShiftIsDown`* が `true` の場合、コンテンツが選択されています。

戻り値

なし

dom.source.endOfLine()

対応バージョン

Dreamweaver 4

説明

挿入ポイントを現在の行の末尾に配置します。コンテンツが既に選択されている場合、この関数は選択範囲を現在の行の末尾まで拡張します。

引数

bShiftIsDown

- *bShiftIsDown* 引数には、コンテンツが選択されているかどうかを示すブール値を指定します。*bShiftIsDown* が `true` の場合、コンテンツが選択されています。

戻り値

なし

dom.source.endPage()

対応バージョン

Dreamweaver 4

説明

挿入ポイントを現在のページの末尾に移動します。挿入ポイントが既にページの末尾にある場合は、次のページの末尾に移動します。コンテンツが既に選択されている場合、この関数は選択範囲を 1 ページずつ拡張します。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを移動するページ数を指定します。*nTimes* を指定しない場合のデフォルト値は 1 です。
- *bShiftIsDown* 引数 (オプション) には、コンテンツが選択されているかどうかを示すブール値を指定します。`bShiftIsDown` が `true` の場合、コンテンツが選択されています。

戻り値

なし

dom.source.getCurrentLines()

対応バージョン

Dreamweaver 4

説明

ドキュメントの先頭を基準とする指定オフセットの位置の行番号を返します。

引数

なし

戻り値

現在の選択範囲の行番号。

dom.source.getSelection()

説明

現在のドキュメント内の選択範囲を、ドキュメントのコードビューにおける文字オフセットとして取得します。

引数

なし

戻り値

ソースドキュメントの先頭からのオフセットを表す整数のペア。最初の整数は選択範囲の開始点を示し、2 番目の整数は選択範囲の終了点を示します。これら 2 つの数値が同じ場合、選択範囲が挿入ポイントであることを示します。ソースに選択範囲がない場合、これら 2 つの数値は -1 になります。

dom.source.getLineFromOffset()

対応バージョン

Dreamweaver MX

説明

ソースドキュメントへのオフセットを取得します。

引数

なし

戻り値

対応する行番号。オフセットが負の値である場合、またはファイルの末尾を超えている場合は、-1。

dom.source.getText()

対応バージョン

Dreamweaver 4

説明

ソースで指定されたオフセット間のテキストストリングを返します。

引数

startOffset、*endOffset*

- *startOffset* 引数には、ドキュメントの先頭からのオフセットを表す整数を指定します。
- *endOffset* 引数には、ドキュメントの末尾を表す整数を指定します。

戻り値

ソースコード内で、*start* と *end* のオフセット間のテキストを表すストリング。

dom.source.getValidationErrorsForOffset()

対応バージョン

Dreamweaver MX 2004

説明

指定されたオフセットでの検証エラーのリストを返すか、またはオフセットから次のエラーを検索します。見つからない場合は `null` を返します。

引数

offset、*{searchDirection}*

- *offset* 引数は、エラーを取得するコード内のオフセットを示す数値です。
- *searchDirection* 引数 (オプション) には、"empty"、"forward"、または "back" のストリングを指定します。指定した場合、指定したオフセットから前方または後方へエラーのある次の文字を検索し、返します。省略した場合は、指定したオフセットでエラーをチェックします。

戻り値

オブジェクトの配列、または null 値。配列の各オブジェクトには、以下のプロパティがあります。

- message オブジェクトは、エラーメッセージを含むストリングです。
- floaterName オブジェクトは、[結果] ウィンドウの名前を含むストリングです。この値は、showResults() 関数または setFloaterVisibility() 関数に渡すことができます。
- floaterIndex オブジェクトは、フローターの結果リストに含まれる項目のインデックスです。
- start オブジェクトは、アンダーラインの付いたコードの開始インデックスです。
- end オブジェクトは、アンダーラインの付いたコードの終了インデックスです。

注意 : 関数から返されたフローターのインデックスは、保存しないでください。これは、ドキュメントの開閉などに応じて、インデックスが頻繁に変化するためです。

例

次の例では、getValidationErrorsForOffset() を呼び出して、現在の選択範囲のオフセットに、エラーがあるかどうかをチェックしています。関数からエラーが返された場合は、alert() 関数を呼び出し、エラーメッセージを表示してユーザーに知らせます。

```
var offset = dw.getDocumentDOM().source.getSelection()[0];
var errors = dw.getDocumentDOM().source.getValidationErrorsForOffset(offset);
if ( errors && errors.length > 0 )
    alert( errors[0].message );
```

dom.source.indentTextview()

対応バージョン

Dreamweaver 4

説明

選択したコードビューテキストを 1 つのタブストップ分右に移動します。

引数

なし

戻り値

なし

dom.source.insert()

対応バージョン

Dreamweaver 4

説明

ソースファイルの先頭を基準とする指定オフセット位置にあるソースコードに指定したストリングを挿入します。オフセットが 0 より小さい場合、挿入は実行されず、関数は `false` を返します。

引数

`offset`、*`string`*

- *`offset`* 引数には、ストリングを挿入する位置をファイル先頭からのオフセットで指定します。
- *`string`* 引数には、挿入するストリングを指定します。

戻り値

ブール値。成功した場合は `true`、失敗した場合は `false`。

dom.source.nextWord()

対応バージョン

Dreamweaver 4

説明

コードビューで、挿入ポイントを次の単語の先頭に移動します。複数の単語が指定された場合は、複数の単語をスキップして移動します。コンテンツが既に選択されている場合、この関数は選択範囲を右に拡張します。

引数

`{nTimes}`、*`{bShiftIsDown}`*

- *`nTimes`* 引数 (オプション) には、挿入ポイントを移動する単語数を指定します。*`nTimes`* を指定しない場合のデフォルト値は 1 です。
- *`bShiftIsDown`* 引数 (オプション) には、コンテンツが選択されているかどうかを示すブール値を指定します。*`bShiftIsDown`* が `true` の場合、コンテンツが選択されています。

戻り値

なし

dom.source.outdentTextview()

対応バージョン

Dreamweaver 4

説明

選択したコードビューテキストを 1 つのタブストップ分左に移動します。

引数

なし

戻り値

なし

dom.source.pageDown()

対応バージョン

Dreamweaver 4

説明

コードビュードキュメントで挿入ポイントを 1 ページずつ下に移動します。コンテンツが既に選択されている場合、この関数は選択範囲を 1 ページずつ拡張します。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを移動するページ数を指定します。*nTimes* を指定しない場合のデフォルト値は 1 です。
- *bShiftIsDown* 引数 (オプション) には、コンテンツが選択されているかどうかを示すブール値を指定します。*bShiftIsDown* が `true` の場合、コンテンツが選択されています。

戻り値

なし

dom.source.pageUp()

対応バージョン

Dreamweaver 4

説明

コードビュードキュメントで挿入ポイントを 1 ページずつ上に移動します。コンテンツが既に選択されている場合、この関数は選択範囲を 1 ページずつ拡張します。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを移動するページ数を指定します。*nTimes* を指定しない場合のデフォルト値は 1 です。
- *bShiftIsDown* 引数 (オプション) には、コンテンツが選択されているかどうかを示すブール値を指定します。*bShiftIsDown* が `true` の場合、コンテンツが選択されています。

戻り値

なし

dom.source.previousWord()

対応バージョン

Dreamweaver 4

説明

コードビューで、挿入ポイントを前の単語の先頭に移動します。複数の単語を指定した場合は、前の複数の単語をスキップして移動します。コンテンツが既に選択されている場合、この関数は選択範囲を左に拡張します。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを移動する単語数を指定します。*nTimes* を指定しない場合のデフォルト値は 1 です。
- *bShiftIsDown* 引数 (オプション) には、コンテンツが選択されているかどうかを示すブール値を指定します。*bShiftIsDown* が `true` の場合、コンテンツが選択されています。

戻り値
なし

dom.source.replaceRange()

対応バージョン
Dreamweaver 4

説明
startOffset から *endOffset* までのソーステキストの範囲を *string* に置換します。*startOffset* の値が *endOffset* の値よりも大きいか、いずれかのオフセットが正の整数でない場合は、何も実行せずに false を返します。*endOffset* の値がファイル内の文字数よりも大きい場合は、*startOffset* からファイルの末尾までの範囲を置換します。*startOffset*、および *endOffset* の値が両方ともファイル内の文字数よりも大きい場合は、ファイルの末尾にテキストを挿入します。

引数
startOffset、*endOffset*、*string*

- *startOffset* 引数には、置換するブロックの先頭を示すオフセットを指定します。
- *endOffset* 引数には、置換するブロックの末尾を示すオフセットを指定します。
- *string* 引数には、挿入するストリングを指定します。

戻り値
ブール値。成功した場合は true、失敗した場合は false。

dom.source.scrollEndFile()

対応バージョン
Dreamweaver 4

説明
挿入ポイントを移動しないまま、コードビューをドキュメントファイルの最下部までスクロールします。

引数
なし

戻り値
なし

dom.source.scrollLineDown()

対応バージョン
Dreamweaver 4

説明
挿入ポイントを移動しないまま、コードビューを 1 行ずつ下にスクロールします。

引数
nTimes

- *nTimes* 引数には、スクロールする行数を指定します。*nTimes* を指定しない場合のデフォルト値は 1 です。

戻り値
なし

dom.source.scrollLineUp()

対応バージョン
Dreamweaver 4

説明
挿入ポイントを移動しないまま、コードビューを 1 行ずつ上にスクロールします。

引数
nTimes

- *nTimes* 引数には、スクロールする行数を指定します。*nTimes* を指定しない場合のデフォルト値は 1 です。

戻り値
なし

dom.source.scrollPageDown()

対応バージョン
Dreamweaver 4

説明
挿入ポイントを移動しないまま、コードビューを 1 ページずつ下にスクロールします。

引数
nTimes

- *nTimes* 引数には、スクロールするページ数を指定します。*nTimes* を指定しない場合のデフォルト値は 1 です。

戻り値
なし

dom.source.scrollPageUp()

対応バージョン
Dreamweaver 4

説明
挿入ポイントを移動しないまま、コードビューを 1 ページずつ上にスクロールします。

引数
nTimes

- *nTimes* 引数には、スクロールするページ数を指定します。*nTimes* を指定しない場合のデフォルト値は 1 です。

戻り値
なし

dom.source.scrollTopFile()

対応バージョン
Dreamweaver 4

説明
挿入ポイントを移動しないまま、コードビューをドキュメントファイルの最上部までスクロールします。

引数
なし

戻り値
なし

dom.source.selectParentTag()

対応バージョン
Dreamweaver 4

説明
この関数は、タグの均衡を使用可能にするコードビューの拡張機能です。`dom.source.selectParentTag()` を呼び出して、現在ハイライト表示されている選択範囲または挿入ポイントを前後の開始タグから終了タグまで拡張することができます。連続して呼び出すと、選択範囲を囲むタグがなくなるまで、前後の他のタグの位置まで選択範囲を拡張します。

引数
なし

戻り値
なし

dom.source.setCurrentLine()

対応バージョン
Dreamweaver 4

説明
挿入ポイントを指定した行の先頭に配置します。*`lineNumber`* 引数の値が正の整数でない場合は、何も実行せずに `false` を返します。*`lineNumber`* の値がソースの行数より大きい場合は、最後の行の先頭に挿入ポイントを配置します。

引数
`lineNumber`

- *`lineNumber`* 引数には、先頭に挿入ポイントを配置する行を指定します。

戻り値
ブール値。成功した場合は `true`、失敗した場合は `false`。

dom.source.startOfDocument()

対応バージョン
Dreamweaver 4

説明

挿入ポイントを現在のコードビュードキュメントの先頭に配置します。コンテンツが既に選択されている場合、この関数は選択範囲をドキュメントの先頭まで拡張します。

引数

bShiftIsDown

- *bShiftIsDown* 引数には、コンテンツが選択されているかどうかを示すブール値を指定します。*bShiftIsDown* が `true` の場合、コンテンツが選択されています。

戻り値

なし

dom.source.startOfLine()

対応バージョン

Dreamweaver 4

説明

挿入ポイントを現在の行の先頭に配置します。コンテンツが既に選択されている場合、この関数は選択範囲を現在の行の先頭まで拡張します。

引数

bShiftIsDown

- *bShiftIsDown* 引数には、コンテンツが選択されているかどうかを示すブール値を指定します。*bShiftIsDown* が `true` の場合、コンテンツが選択されています。

戻り値

なし

dom.source.topPage()

対応バージョン

Dreamweaver 4

説明

挿入ポイントを現在のページの最上部に移動します。挿入ポイントが既にページの最上部にある場合は、前のページの最上部に移動します。コンテンツが既に選択されている場合、この関数は選択範囲を 1 ページずつ拡張します。

引数

{nTimes}、*{bShiftIsDown}*

- *nTimes* 引数 (オプション) には、挿入ポイントを移動するページ数を指定します。*nTimes* を指定しない場合のデフォルト値は 1 です。
- *bShiftIsDown* 引数 (オプション) には、コンテンツが選択されているかどうかを示すブール値を指定します。*bShiftIsDown* が `true` の場合、コンテンツが選択されています。

戻り値

なし

dom.source.wrapSelection()

対応バージョン
Dreamweaver 4

説明
startTag のテキストを現在の選択範囲の前に、*endTag* のテキストを現在の選択範囲の後ろに挿入します。その後、挿入したタグを含み、両方のタグに挟まれた範囲全体を選択します。現在の選択範囲が挿入ポイントの場合、*startTag* と *endTag* の間に挿入ポイントを配置します。(*startTag* と *endTag* には、タグだけでなく任意のテキストを指定できます。)

引数
startTag、*endTag*

- *startTag* 引数には、選択範囲の先頭に挿入するテキストを指定します。
- *endTag* 引数には、選択範囲の末尾に挿入するテキストを指定します。

戻り値
なし

dom.synchronizeDocument()

対応バージョン
Dreamweaver 4

説明
デザインビューとコードビューを同期化します。

引数
なし

戻り値
なし

タグエディタおよびタグライブラリ関数

タグエディタを使用して、新しいタグの挿入、既存タグの編集、およびタグに関するリファレンス情報の参照を行うことができます。また、タグ選択を使用すると、使用頻度の高いタグを簡単に選択できるように、タグを整理することができます。Dreamweaver に付属するタグライブラリには、標準ベースのマークアップ言語および一般的なタグベースのスクリプト言語で使用されるタグに関する情報が格納されています。拡張機能で、タグエディタやタグライブラリにアクセスし、それらを操作する必要がある場合は、JavaScript タグエディタ、タグ選択、およびタグライブラリ関数を使用することができます。

dom.getTagSelectorTag()

対応バージョン
Dreamweaver MX

説明
ドキュメントウィンドウの一番下にあるタグセレクタバーで現在選択されているタグの DOM ノードを取得します。

引数
なし

戻り値
現在選択されているタグの DOM ノード。タグが選択されていない場合は、`null` が返されます。

dreamweaver.popupInsertTagDialog()

対応バージョン
Dreamweaver MX

説明
VTM ファイルを参照して、タグにタグエディタが定義されているかどうかを確認します。定義されている場合は、そのタグ用のエディタが表示され、開始タグが使用されます。定義されていない場合は、開始タグがそのままユーザーのドキュメントに挿入されます。

引数
`start_tag_string`

次のいずれかのタイプの初期値を含む開始タグストリングを指定します。

- `<input>` などのタグ
- `<input type='text'>` などの属性があるタグ
- `<%= %>` などのディレクティブ

戻り値
ドキュメントにタグが挿入された場合は `true`、それ以外の場合は `false` のブール値。

dreamweaver.popupEditTagDialog()

対応バージョン
Dreamweaver MX

説明
タグが選択されているとき、タグを編集できるように、そのタグ用のタグエディタを表示します。

引数
なし

戻り値
なし

イネーブラ
詳細については、420 ページの dreamweaver.canPopupEditTagDialog() を参照してください。

dreamweaver.showTagChooser()

対応バージョン
Dreamweaver MX

説明
[タグ選択] ダイアログボックスを前面に表示し、フォーカスを設定します。

引数
なし

戻り値
なし

dreamweaver.showTagLibraryEditor()

対応バージョン
Dreamweaver MX

説明
タグライブラリエディタを開きます。

引数
なし

戻り値
なし

dreamweaver.tagLibrary.getTagLibraryDOM()

対応バージョン
Dreamweaver MX

説明
filename.vtm ファイルの URL を指定すると、そのファイルの DOM が返されるので、ファイルのコンテンツを編集することができます。この関数は、タグライブラリエディタがアクティブな場合にのみ呼び出すことができます。

引数
fileURL

- *fileURL* 引数には、次の例のように *filename*.vtm ファイルの URL を "Configuration/Tag Libraries" フォルダを基準として指定します。 "HTML/img.vtm"

戻り値
"TagLibraries" フォルダ内の新規または既存のファイルへの DOM ポインタ。

dreamweaver.tagLibrary.getSelectedLibrary()

対応バージョン
Dreamweaver MX

説明
タグライブラリエディタでライブラリノードが選択されている場合は、ライブラリ名を取得します。

引数
なし

戻り値
タグライブラリエディタで、現在選択されているライブラリの名前のストリング。ライブラリが選択されていない場合は、空白のストリングが返されます。

dreamweaver.tagLibrary.getSelectedTag()

対応バージョン
Dreamweaver MX

説明
属性ノードが現在選択されている場合に、属性を含むタグの名前を取得します。

引数
なし

戻り値
タグライブラリエディタで、現在選択されているタグの名前のストリング。タグが選択されていない場合は、空白のストリングが返されます。

dreamweaver.tagLibrary.importDTDOrSchema()

対応バージョン
Dreamweaver MX

説明
DTD またはスキーマファイルをリモートサーバーからタグライブラリに読み込みます。

引数
fileURL、*Prefix*

- *fileURL* 引数には、DTD またはスキーマファイルへのパスをローカル URL 書式で指定します。
- *Prefix* 引数には、このタグライブラリのすべてのタグに追加する接頭辞ストリングを指定します。

戻り値
読み込まれたタグライブラリの名前。

dreamweaver.tagLibrary.getImportedTagList()

対応バージョン
Dreamweaver MX

説明
読み込まれたタグライブラリから `tagInfo` オブジェクトのリストを生成します。

引数

libname

- *libname* 引数には、読み込まれたタグライブラリの名前を指定します。

戻り値

tagInfo オブジェクトの配列。

taginfo オブジェクトは、タグライブラリ内の 1 つのタグに関する情報を保持しています。tagInfo オブジェクトには、次のプロパティが定義されています。

- tagName プロパティはストリングです。
- attributes プロパティは、ストリングの配列です。各ストリングは、このタグに定義されている属性の名前です。

例 :

次の例では、dw.tagLibrary.getImportedTagList() 関数を使用して libName ライブラリからタグの配列を取得しています。

```
    // "fileURL" および "prefix" はユーザーによって入力されています。
    // タグライブラリが DTD/Schema を読み込むように指示します。
var libName = dw.tagLibrary.importDTDOrSchema(fileURL, prefix);

    // このライブラリのタグの配列を取得します。
    // これが tagInfo オブジェクトです。
var tagArray = dw.tagLibrary.getImportedTagList(libName);

    // tagInfo オブジェクトの配列を取得しました。
    // これらのオブジェクトから情報を取得できます。最初の配列から情報が取得されます。
    // メモ : 配列内に少なくとも 1 つの tagInfo があることを前提としています。
var firstTagName = tagArray[0].name;
var firstTagAttributes = tagArray[0].attributes;
    // firstTagAttributes は属性の配列です。
```

第 19 章 : イネーブラ

Adobe® Dreamweaver® CS3 のイネーブラ関数は、他の関数が現在のコンテキストで個々の処理を実行できるかどうかを決定します。以下の関数の戻り値の説明では、各関数から `true` が返される一般的な条件が記述されています。ただし、この記述は包括的なものではありません。`false` 値が返される場合については説明されないこともあります。

イネーブラ関数

JavaScript API のイネーブラ関数には、以下のものがあります。

dom.canAlign()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [左揃え]、[右揃え]、[上揃え]、[下揃え] の各操作を実行できるかどうかをチェックします。

引数
なし

戻り値
2 つ以上のレイヤーまたはホットスポットが選択されているかどうかを示すブール値。

dom.canApplyTemplate()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [ページに適用] の処理を実行できるかどうかをチェックします。この関数はアクティブなドキュメントのみに適用されます。

引数
なし

戻り値
ドキュメントがライブラリ項目またはテンプレート以外であるかどうか、また選択範囲が `NOFRAMES` タグで囲まれていないかどうかを示すブール値。

dom.canArrange()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [前面に移動] または [背面に移動] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
ホットスポットが選択されているかどうかを示すブール値。

dom.canClipCopyText()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [テキストとしてコピー] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
選択範囲の開始オフセットと終了オフセットが異なる場合は `true`、同じ場合は何も選択されていないことを示す `false` のブール値。

dom.canClipPaste()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [ペースト] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
Dreamweaver にペーストできるコンテンツがクリップボードに格納されている場合は `true`、それ以外の場合は `false` のブール値。

dom.canClipPasteText()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [テキストとしてペースト] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
Dreamweaver にテキストとしてペーストできるコンテンツがクリップボードに格納されている場合は `true`、それ以外の場合は `false` のブール値。

dom.canConvertLayersToTable()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [レイヤーをテーブルに変換] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
ドキュメントの BODY セクションのコンテンツがすべてレイヤー内にある場合は true、それ以外の場合は false のブール値。

dom.canConvertTablesToLayers()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [テーブルをレイヤーに変換] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
ドキュメントの BODY セクションのコンテンツがすべてテーブル内にあり、ドキュメントにテンプレートが使われていない場合は true、それ以外の場合は false のブール値。

dom.canDecreaseColspan()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [列の縮小] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
現在のセルに COLSPAN 属性があり、この属性の値が 2 以上である場合は true、それ以外の場合は false のブール値。

dom.canDecreaseRowspan()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [行の縮小] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
現在のセルに ROWSPAN 属性があり、この属性の値が 2 以上である場合は true、それ以外の場合は false のブール値。

dom.canDeleteTableColumn()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [列の削除] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
挿入ポイントがセルの中にあるか、セルまたは列が選択されている場合は true、それ以外の場合は false のブール値。

dom.canDeleteTableRow()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [行の削除] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
挿入ポイントがセルの中にあるか、セルまたは行が選択されている場合は true、それ以外の場合は false のブール値。

site.canEditColumns()

説明
サイトが存在するかどうかをチェックします。

引数
なし

戻り値
サイトが存在する場合は true、それ以外の場合は false のブール値。

dom.canEditNoFramesContent()

対応バージョン
Dreamweaver 3

説明

Dreamweaver が [フレームなしコンテンツの編集] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

現在のドキュメントがフレームセットであるか、またはフレームセット内にある場合は `true`、それ以外の場合は `false` のブール値。

dom.canIncreaseColspan()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [列の拡大] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

現在のセルの右にセルがある場合は `true`、それ以外の場合は `false` のブール値。

dom.canIncreaseRowspan()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [行の拡大] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

現在のセルの下にセルがある場合は `true`、それ以外の場合は `false` のブール値。

dom.canInsertTableColumns()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [列の挿入] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

選択範囲がテーブル内にある場合は `true`、選択範囲がテーブル全体であるか、またはテーブル内にない場合は `false` のブール値。

dom.canInsertTableRows()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [行の挿入] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

選択範囲がテーブル内にある場合は `true`、選択範囲がテーブル全体であるか、またはテーブル内にない場合は `false` のブール値。

dom.canMakeNewEditableRegion()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [新規の編集可能領域] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

現在のドキュメントがテンプレート (DWT) ファイルである場合は `true` のブール値。

dom.canMarkSelectionAsEditable()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [選択範囲を編集可能としてマーク] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

選択範囲があり、現在のドキュメントが DWT ファイルである場合は `true`、それ以外の場合は `false` のブール値。

dom.canMergeTableCells()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [セルのマージ] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
選択範囲が隣り合ったテーブルセルの集合である場合は `true`、それ以外の場合は `false` のブール値。

dom.canPlayPlugin()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [再生] の処理を実行できるかどうかをチェックします。この関数はアクティブなドキュメントのみに適用されます。

引数
なし

戻り値
選択範囲をプラグインで再生できる場合は `true` のブール値。

dom.canRedo()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [やり直し] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
やり直す手順が残っている場合は `true`、それ以外の場合は `false` のブール値。

dom.canRemoveEditableRegion()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [編集可能領域のマーク解除] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
現在のドキュメントがテンプレートである場合は `true`、それ以外の場合は `false` のブール値。

dom.canSelectTable()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [テーブルの選択] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
挿入ポイントまたは選択範囲がテーブル内にある場合は `true`、それ以外の場合は `false` のブール値。

dom.canSetLinkHref()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が現在の選択範囲に設定されているリンクを変更できるかどうか、または必要な場合はリンクを作成できるかどうかをチェックします。

引数
なし

戻り値
選択範囲がイメージかテキストである場合、または挿入ポイントがリンク内にある場合は `true` 、それ以外の場合は `false` のブール値。テキスト選択範囲とは、テキストのプロパティインスペクタが表示される選択範囲を指します。

dom.canShowListPropertiesDialog()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [リストプロパティ] ダイアログボックスを表示できるかどうかをチェックします。

引数
なし

戻り値
選択範囲が `LI` タグ内にある場合は `true`、それ以外の場合は `false` のブール値。

dom.canSplitFrame()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [フレームを左に分割]、[フレームを右に分割]、[フレームを上に分割]、[フレームを下に分割] の各操作を実行できるかどうかをチェックします。

引数
なし

戻り値
選択範囲がフレーム内にある場合は `true`、それ以外の場合は `false` のブール値。

dom.canSplitTableCell()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [セルの分割] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
挿入ポイントがテーブルセル内にあるか、または選択範囲がテーブルセルである場合は `true`、それ以外の場合は `false` のブール値。

dom.canStopPlugin()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [停止] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
選択範囲がプラグインで現在再生中である場合は `true`、それ以外の場合は `false` のブール値。

dom.canUndo()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [取り消し] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
取り消す手順が残っている場合は `true`、それ以外の場合は `false` のブール値。

dom.hasTracingImage()

対応バージョン
Dreamweaver 3

説明
ドキュメントにトレーシングイメージがあるかどうかをチェックします。

引数
なし

戻り値
ドキュメントにトレーシングイメージがある場合は `true`、それ以外の場合は `false` のブール値。

dreamweaver.assetPalette.canEdit()

対応バージョン
Dreamweaver 4

説明
メニュー項目を [アセット] パネルで編集可能にします。

引数
なし

戻り値
アセットが編集可能な場合は `true` 、編集できない場合は `false` のブール値。[サイト] リストのカラーと URL に対しては `false` 値、[お気に入り] リストで複数選択したカラーと URL に対しては `false` 値がそれぞれ返されます。

dreamweaver.assetPalette.canInsertOrApply()

対応バージョン
Dreamweaver 4

説明
選択したエレメントが挿入可能または適用可能かどうかをチェックします。メニュー項目を挿入または適用の操作に対して使用可能または使用不可にできるように、`true` と `false` のいずれかの値を返します。

引数
なし

戻り値

選択されているエレメントを挿入または適用できる場合は true、現在のページがテンプレートで、現在のカテゴリが [テンプレート] である場合は false のブール値。また、開いているドキュメントがない場合、またはドキュメントでライブラリ項目が選択されていて、現在のカテゴリが [ライブラリ] である場合も、false 値を返します。

dreamweaver.canClipCopy()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [コピー] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

クリップボードにコピーできるコンテンツが選択されている場合は true、それ以外の場合は false のブール値。

dreamweaver.canClipCut()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [カット] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

クリップボードにカットできるコンテンツが選択されている場合は true、それ以外の場合は false のブール値。

dreamweaver.canClipPaste()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [ペースト] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

現在のドキュメントまたは [サイト] パネルのアクティブなペインにペースト可能なコンテンツ、または Macintosh の場合はフローティングパネルまたはダイアログボックスのテキストフィールドにペースト可能なコンテンツが、クリップボードに含まれている場合は true、それ以外の場合は false のブール値。

dreamweaver.canDeleteSelection()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が現在の選択範囲を削除できるかどうかをチェックします。フォーカスがあるウィンドウによっては、ドキュメントウィンドウや [サイト] パネル、または Macintosh の場合はダイアログボックスやフローティングパネルのテキストフィールドで、削除が実行される場合があります。

引数
なし

戻り値
選択範囲の開始オフセットと終了オフセットが異なる場合は、選択範囲があることを示す `true`、オフセットが一致している場合は、挿入ポイントしかないことを示す `false` のブール値。

dreamweaver.canExportCSS() (非推奨)

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [CSS スタイルの書き出し] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
ドキュメントの `HEAD` セクションにクラススタイルが定義されている場合は `true`、それ以外の場合は `false` のブール値。

dreamweaver.canExportTemplateDataAsXML()

対応バージョン
Dreamweaver MX

説明
Dreamweaver が現在のドキュメントを XML として書き出せるかどうかをチェックします。

引数
なし

戻り値
現在のドキュメントで書き出しを実行できる場合は `true`、実行できない場合は `false` のブール値。

例

次の例では、dw.canExportTemplateDataAsXML() を呼び出して、Dreamweaver が現在のドキュメントを XML として書き出せるかどうかを判別し、true が返された場合は dw.ExportTemplateDataAsXML() を呼び出して書き出しを実行します。

```
if(dreamweaver.canExportTemplateDataAsXML())
{
    dreamweaver.exportTemplateDataAsXML("file:///c|/dw_temps/mytemplate.txt")
}
```

dreamweaver.canFindNext()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [次を検索] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

検索パターンが既に設定されている場合は true、それ以外の場合は false のブール値。

dreamweaver.canFitSelection()

対応バージョン

Dreamweaver 8

説明

アクティブなデザインビューに選択範囲があるかどうか、つまり、fitSelection() を呼び出せるかどうかをチェックします。

引数

なし

戻り値

アクティブなデザインビューに選択範囲がある場合は true、それ以外の場合は false のブール値。

dreamweaver.canOpenInFrame()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [フレーム内に開く] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値
選択範囲または挿入ポイントがフレーム内にある場合は `true`、それ以外の場合は `false` のブール値。

dreamweaver.canPasteSpecial()

対応バージョン
Dreamweaver 8

説明
Dreamweaver が [ペーストスペシャル] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
クリップボードにテキスト、HTML、または Dreamweaver HTML があり、フォーカスがコードビュー、デザインビュー、またはコードインスペクタにある場合は `true`、それ以外の場合は `false` のブール値。

dreamweaver.canPlayRecordedCommand()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [記録済みコマンドの再生] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
アクティブなドキュメントと再生可能な記録済みのコマンドがある場合は `true` 、それ以外の場合は `false` のブール値。

dreamweaver.canPopupEditTagDialog()

対応バージョン
Dreamweaver MX

説明
現在の選択範囲がタグかどうか、また [タグの編集] メニュー項目がアクティブになっているかどうかをチェックします。

引数
なし

戻り値
現在選択されているタグの名前。タグが選択されていない場合は、`null` 値が返されます。

dreamweaver.canRedo()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が現在のコンテキストで [やり直し] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
取り消し可能な操作があるかどうかを示すブール値。

dreamweaver.canRevertDocument()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が最後に保存したバージョンへの [復帰] の処理を実行できるかどうかをチェックします。

引数
documentObject

- *documentObject* 引数には、ドキュメントの DOM ツリーのルートにあるオブジェクト (`dreamweaver.getDocumentDOM()` 関数で返される値) を指定します。

戻り値
このドキュメントについて、現在の状態が保存されておらず、保存されたバージョンがローカルドライブに存在するかどうかを示すブール値。

dreamweaver.canSaveAll()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [すべて保存] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
保存されていないドキュメントが開いているかどうかを示すブール値。

dreamweaver.canSaveDocument()

対応バージョン
Dreamweaver 3

説明

Dreamweaver が、指定したドキュメントに対して [保存] の処理を実行できるかどうかをチェックします。

引数

documentObject

- *documentObject* 引数には、ドキュメントの DOM のルート (`dreamweaver.getDocumentDOM()` 関数で返される値と同じ) を指定します。

戻り値

保存されていない変更がドキュメント内にあるかどうかを示すブール値。

dreamweaver.canSaveDocumentAsTemplate()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が、指定したドキュメントに対して [テンプレートとして保存] の処理を実行できるかどうかをチェックします。

引数

documentObject

- *documentObject* 引数には、ドキュメントの DOM のルート (`dreamweaver.getDocumentDOM()` 関数で返される値と同じ) を指定します。

戻り値

ドキュメントをテンプレートとして保存できるかどうかを示すブール値。

dreamweaver.canSaveFrameset()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が、指定したドキュメントに対して [フレームセットの保存] の処理を実行できるかどうかをチェックします。

引数

documentObject

- *documentObject* 引数には、ドキュメントの DOM のルート (`dreamweaver.getDocumentDOM()` 関数で返される値と同じ) を指定します。

戻り値

ドキュメントが、保存されていない変更を含むフレームセットであるかどうかを示すブール値。

dreamweaver.canSaveFramesetAs()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が、指定したドキュメントに対して [フレームセットの新規保存] の処理を実行できるかどうかをチェックします。

引数

documentObject

- *documentObject* 引数には、ドキュメントの DOM のルート (`dreamweaver.getDocumentDOM()` 関数で返される値と同じ) を指定します。

戻り値

ドキュメントがフレームセットであるかどうかを示すブール値。

dreamweaver.canSelectAll()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [すべて選択] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

[すべて選択] の処理を実行できるかどうかを示すブール値。

dreamweaver.canShowFindDialog()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [検索] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

[サイト] パネルまたはドキュメントウィンドウが開いている場合に `true` となるブール値。この関数は、選択範囲が `HEAD` セクション内にある場合は `false` 値を返します。

dreamweaver.canUndo()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が現在のコンテキストで [取り消し] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
取り消し可能な操作があるかどうかを示すブール値。

dreamweaver.canZoom()

対応バージョン
Dreamweaver 8

説明
アクティブなデザインビューがあるかどうか、つまり基本的なズームコマンドを適用できるかどうかをチェックします。

引数
なし

戻り値
アクティブなデザインビューがある場合は `true`、それ以外の場合は `false` のブール値。

dreamweaver.cssRuleTracker.canEditSelectedRule()

対応バージョン
Dreamweaver MX 2004

説明
選択されているルールをプロパティグリッドで修正できるかどうかチェックします。プロパティグリッドでは、ロックされたファイルのルールも表示できるため、戻り値が `true` であってもルールを修正できるとは限りません。

引数
なし

戻り値
選択されているルールにプロパティグリッドエディタを適用できる場合は `true`、それ以外の場合は `false` のブール値。

例
次のコードでは、選択されているルールへの編集を許可する前に、イネーブラ関数が値 `true` に設定されているかどうかをチェックします。

```
if(dw.cssRuleTracker.canEditSelectedRule()){
    dw.cssRuleTracker.editSelectedRule();
}
```

dreamweaver.cssStylePalette.canApplySelectedStyle()

対応バージョン
Dreamweaver MX

説明

現在アクティブなドキュメントに、選択したスタイルを適用できるかどうかをチェックします。

引数

{ pane }

- *pane* 引数 (オプション) は、この関数を [スタイル] パネルのどのペインに適用するかを指定するストリングです。指定可能な値は、`"stylelist"` ([すべて] モードのスタイルのリスト)、`"cascade"` ([現在] モードの適用可能な関連ルールのリスト)、`"summary"` ([現在] モードの現在の選択範囲のプロパティのリスト)、`"ruleInspector"` ([現在] モードのプロパティの編集可能なリストまたはグリッド) です。デフォルト値は `"stylelist"` です。

戻り値

選択したスタイルにクラスセレクタがある場合は `true`、ない場合は `false` のブール値。

dreamweaver.cssStylePalette.canDeleteSelectedStyle()

対応バージョン

Dreamweaver MX

説明

現在の選択範囲で、選択したスタイルを削除できるかどうかを判別します。

引数

{ pane }

- *pane* 引数 (オプション) は、この関数を [スタイル] パネルのどのペインに適用するかを指定するストリングです。指定可能な値は、`"stylelist"` ([すべて] モードのスタイルのリスト)、`"cascade"` ([現在] モードの適用可能な関連ルールのリスト)、`"summary"` ([現在] モードの現在の選択範囲のプロパティのリスト)、`"ruleInspector"` ([現在] モードのプロパティの編集可能なリストまたはグリッド) です。デフォルト値は `"stylelist"` です。

戻り値

選択範囲を削除できる場合は `true`、削除できない場合は `false` のブール値。

dreamweaver.cssStylePalette.canDuplicateSelectedStyle()

対応バージョン

Dreamweaver MX

説明

現在アクティブなドキュメントで、選択したスタイルを複製できるかどうかをチェックします。

引数

{ pane }

- *pane* 引数 (オプション) は、この関数を [スタイル] パネルのどのペインに適用するかを指定するストリングです。指定可能な値は、`"stylelist"` ([すべて] モードのスタイルのリスト)、`"cascade"` ([現在] モードの適用可能な関連ルールのリスト)、`"summary"` ([現在] モードの現在の選択範囲のプロパティのリスト)、`"ruleInspector"` ([現在] モードのプロパティの編集可能なリストまたはグリッド) です。デフォルト値は `"stylelist"` です。

戻り値

選択したスタイルを複製できる場合は `true`、複製できない場合は `false` のブール値。

dreamweaver.cssStylePalette.canEditSelectedStyle()

対応バージョン
Dreamweaver MX

説明
現在アクティブなドキュメントで、選択したスタイルを編集できるかどうかをチェックします。

引数
{ pane }

- *pane* 引数 (オプション) は、この関数を [スタイル] パネルのどのペインに適用するかを指定するストリングです。指定可能な値は、`"stylelist"` ([すべて] モードのスタイルのリスト)、`"cascade"` ([現在] モードの適用可能な関連ルールのリスト)、`"summary"` ([現在] モードの現在の選択範囲のプロパティのリスト)、`"ruleInspector"` ([現在] モードのプロパティの編集可能なリストまたはグリッド) です。デフォルト値は `"stylelist"` です。

戻り値
選択したスタイルを編集できる場合は `true`、編集できない場合は `false` のブール値。

dreamweaver.cssStylePalette.canEditSelectedStyleInCodeview()

対応バージョン
Dreamweaver MX

説明
現在アクティブなドキュメントで選択したスタイルをコードビューで編集できるかどうかをチェックします。

引数
{ pane }

- *pane* 引数 (オプション) は、この関数を [スタイル] パネルのどのペインに適用するかを指定するストリングです。指定可能な値は、`"stylelist"` ([すべて] モードのスタイルのリスト)、`"cascade"` ([現在] モードの適用可能な関連ルールのリスト)、`"summary"` ([現在] モードの現在の選択範囲のプロパティのリスト)、`"ruleInspector"` ([現在] モードのプロパティの編集可能なリストまたはグリッド) です。デフォルト値は `"stylelist"` です。

戻り値
選択したスタイルを編集できる場合は `true`、編集できない場合は `false` のブール値。

dreamweaver.cssStylePalette.canEditStyleSheet()

対応バージョン
Dreamweaver MX

説明
現在の選択範囲に、編集可能なスタイルシートエレメントが含まれているかどうかをチェックします。

引数
なし

戻り値
選択範囲がスタイルシートノードまたはスタイルシートノード内のスタイル定義で、そのスタイルシートが非表示でも現在のドキュメントでもない場合は `true`、選択範囲が非表示か、または現在のドキュメントにある場合は `false` のブール値。

dreamweaver.cssStylePalette.canRenameSelectedStyle()

対応バージョン
Dreamweaver MX

説明
現在アクティブなドキュメントで、選択したスタイルの名前を変更できるかどうかをチェックします。

引数
{ pane }

- *pane* 引数 (オプション) は、この関数を [スタイル] パネルのどのペインに適用するかを指定するストリングです。指定可能な値は、`"stylelist"` ([すべて] モードのスタイルのリスト)、`"cascade"` ([現在] モードの適用可能な関連ルールのリスト)、`"summary"` ([現在] モードの現在の選択範囲のプロパティのリスト)、`"ruleInspector"` ([現在] モードのプロパティの編集可能なリストまたはグリッド) です。デフォルト値は `"stylelist"` です。

戻り値
選択したスタイルの名前を変更できる場合は `true`、変更できない場合は `false` のブール値。

dreamweaver.isRecording()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が現在コマンドを記録しているかどうかを報告します。

引数
なし

戻り値
Dreamweaver がコマンドを記録しているかどうかを示すブール値。

dreamweaver.htmlStylePalette.canEditSelection()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [HTML スタイル] パネルで選択範囲の編集、削除、複製を実行できるかどうかをチェックします。

引数
なし

戻り値
Dreamweaver が [HTML スタイル] パネルで選択範囲の編集、削除、複製を実行できる場合は `true`、スタイルが選択されていないか、またはクリアなスタイルのいずれかが選択されている場合は `false` のブール値。

dreamweaver.resultsPalette.canClear()

対応バージョン
Dreamweaver MX

説明
現在フォーカスがある [結果] パネルのコンテンツを消去できるかどうかをチェックします。

引数
なし

戻り値
コンテンツをクリアできる場合は `true`、クリアできない場合は `false` のブール値。

dreamweaver.resultsPalette.canCopy()

対応バージョン
Dreamweaver MX

説明
コピーしたメッセージを、現在の [結果] ウィンドウのコンテンツ内に表示できるかどうかをチェックします。

引数
なし

戻り値
コンテンツを表示できる場合は `true`、表示できない場合は `false` のブール値。

dreamweaver.resultsPalette.canCut()

対応バージョン
Dreamweaver MX

説明
カットしたメッセージを現在の [結果] ウィンドウのコンテンツ内に表示できるかどうかをチェックします。

引数
なし

戻り値
コンテンツを表示できる場合は `true`、表示できない場合は `false` のブール値。

dreamweaver.resultsPalette.canPaste()

対応バージョン
Dreamweaver MX

説明
ペーストしたメッセージを現在の [結果] ウィンドウのコンテンツ内に表示できるかどうかをチェックします。

引数
なし

戻り値
コンテンツを表示できる場合は `true`、表示できない場合は `false` のブール値。

dreamweaver.resultsPalette.canOpenInBrowser()

対応バージョン
Dreamweaver MX

説明
現在のレポートをブラウザで表示できるかどうかをチェックします。

引数
なし

戻り値
コンテンツを表示できる場合は `true`、表示できない場合は `false` のブール値。

dreamweaver.resultsPalette.canOpenInEditor()

対応バージョン
Dreamweaver MX

説明
現在のレポートをエディタで表示できるかどうかをチェックします。

引数
なし

戻り値
コンテンツを表示できる場合は `true`、表示できない場合は `false` のブール値。

dreamweaver.resultsPalette.canSave()

対応バージョン
Dreamweaver MX

説明
現在のウィンドウに対して、[保存] ダイアログボックスを開くことができるかどうかをチェックします。現時点では、[保存] ダイアログボックスは、[サイトレポート]、[ターゲットブラウザチェック]、[バリデータ]、[リンクチェック] パネルでサポートされています。

引数
なし

戻り値
[保存] ダイアログボックスを表示できる場合は `true`、表示できない場合は `false` のブール値。

dreamweaver.resultsPalette.canSelectAll()

対応バージョン
Dreamweaver MX

説明
現在フォーカスがあるウィンドウに、[すべて選択] メッセージを送信できるかどうかをチェックします。

引数
なし

戻り値
[すべて選択] メッセージを送信できる場合は `true`、送信できない場合は `false` のブール値。

dreamweaver.siteSyncDialog.canCompare()

対応バージョン
Dreamweaver 8

説明
この関数は、[同期] ダイアログボックスに [比較] コンテキストメニューを表示できるかどうかをチェックします。

引数
なし

戻り値
[サイトの同期] ダイアログボックスで [比較] コンテキストメニューを表示できる場合は `true`、表示できない場合は `false` のブール値。

dreamweaver.siteSyncDialog.canMarkDelete()

対応バージョン
Dreamweaver 8

説明
この関数は、[同期] ダイアログボックスに [アクションを削除に変更] コンテキストメニューを表示できるかどうかをチェックします。

引数
なし

戻り値
[削除するアクションの変更] コンテキストメニューを表示できる場合は `true`、表示できない場合は `false` のブール値。

dreamweaver.siteSyncDialog.canMarkGet()

対応バージョン
Dreamweaver 8

説明
この関数は、[同期] ダイアログボックスに [アクションを GET に変更] コンテキストメニューを表示できるかどうかをチェックします。

引数
なし

戻り値
[取得するアクションの変更] コンテキストメニューを表示できる場合は `true`、表示できない場合は `false` のブール値。

dreamweaver.siteSyncDialog.canMarkIgnore()

対応バージョン
Dreamweaver 8

説明
この関数は、[同期] ダイアログボックスに [アクションを無視に変更] コンテキストメニューを表示できるかどうかをチェックします。

引数
なし

戻り値
[無視するアクションの変更] コンテキストメニューを表示できる場合は `true`、表示できない場合は `false` のブール値。

dreamweaver.siteSyncDialog.canMarkPut()

対応バージョン
Dreamweaver 8

説明
この関数は、[同期] ダイアログボックスに [アクションを PUT に変更] コンテキストメニューを表示できるかどうかをチェックします。

引数
なし

戻り値
[送信するアクションの変更] コンテキストメニューを表示できる場合は `true`、表示できない場合は `false` のブール値。

dreamweaver.siteSyncDialog.canMarkSynced()

対応バージョン
Dreamweaver 8

説明
この関数は、[同期] ダイアログボックスに [アクションを同期に変更] コンテキストメニューを表示できるかどうかをチェックします。

引数
なし

戻り値
[同期するアクションの変更] コンテキストメニューを表示できる場合は `true`、表示できない場合は `false` のブール値。

dreamweaver.snippetpalette.canEditSnippet()

対応バージョン
Dreamweaver MX

説明
現在選択している項目を編集できるかどうかをチェックします。編集するためにメニュー項目を有効または無効にできるように、`true` と `false` のいずれかの値が返されます。

引数
なし

戻り値
現在選択している項目を編集できる場合は `true`、それ以外の場合は `false` のブール値。

dreamweaver.snippetpalette.canInsert()

対応バージョン
Dreamweaver MX

説明
選択したエレメントを挿入または適用できるかどうかをチェックします。挿入または適用するためのメニュー項目を有効または無効にできるように、`true` と `false` のいずれかの値が返されます。

引数
なし

戻り値
選択したエレメントを挿入または適用できる場合は `true`、それ以外の場合は `false` のブール値。

site.browseDocument()

対応バージョン
Dreamweaver 4

説明
選択したすべてのドキュメントをブラウザウィンドウで開きます。[ブラウザでプレビュー] コマンドを使用することと同じです。

引数
`browserName`

- *`browserName`* 引数には、[ブラウザでプレビュー] 環境設定で定義されているブラウザの名前を指定します。この引数を省略すると、デフォルトとしてユーザーのプライマリブラウザが使用されます。

戻り値

なし

site.canAddLink()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [既存ファイルへリンク] または [新規ファイルへリンク] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

サイトマップで選択したドキュメントが HTML ファイルである場合は `true`、それ以外の場合は `false` ブール値。

site.canChangeLink()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [リンクの変更] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

HTML または Flash ファイルから、サイトマップで選択したファイルにリンクが設定されている場合は `true` 、それ以外の場合は `false` のブール値。

site.canCheckIn()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [チェックイン] の処理を実行できるかどうかをチェックします。

引数

`siteOrURL`

- *`siteOrURL`* 引数には、この関数が [サイト] パネルの選択範囲に対して動作することを示す `site` キーワード、または 1 つのファイルの URL を指定する必要があります。

戻り値

以下のすべての条件が満たされている場合は `true`、それ以外の場合は `false` のブール値。

- リモートサイトが定義されていること。

- ドキュメントウィンドウにフォーカスがある場合は、ファイルがローカルサイトに保存されていること。または、[サイト] パネルにフォーカスがある場合は、1 つまたは複数のファイルまたはフォルダが選択されていること。
- サイトのチェックイン / チェックアウト機能がオンになっていること。

site.canCheckOut()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が、指定したファイルを [チェックアウト] できるかどうかをチェックします。

引数

siteOrURL

- *siteOrURL* 引数には、この関数が [サイト] パネルの選択範囲に対して動作することを示す `site` キーワード、または 1 つのファイルの URL を指定する必要があります。

戻り値

以下のすべての条件が満たされている場合は `true`、それ以外の場合は `false` のブール値。

- リモートサイトが定義されていること。
- ドキュメントウィンドウにフォーカスがある場合は、ファイルがローカルサイトの一部であり、まだチェックアウトされていないこと。または、[サイト] パネルにフォーカスがある場合は、1 つまたは複数のファイルまたはフォルダが選択されていて、選択されたファイルのうち少なくとも 1 つがまだチェックアウトされていないこと。
- サイトのチェックイン / チェックアウト機能がオンになっていること。

site.canCloak()

対応バージョン

Dreamweaver MX

説明

Dreamweaver が [クローク] の処理を実行できるかどうかを判別します。

引数

siteOrURL

- *siteOrURL* 引数には、`canCloak()` 関数が [サイト] パネルの選択範囲に対して動作することを示す `site` キーワード、または指定されたフォルダとそのすべてのコンテンツに対して動作することを示す、特定のフォルダの URL を指定する必要があります。

戻り値

Dreamweaver が現在のサイトまたは指定したフォルダで [クローク] の操作を実行できる場合は `true`、実行できない場合は `false` のブール値。

site.canCompareFiles()

対応バージョン

Dreamweaver 8

説明
この関数は、選択したファイルに対して Compare 関数を実行できるかどうかをチェックします。

引数
なし

戻り値
2 つのファイル (ローカルファイルとリモートファイル、2 つのローカルファイル、または 2 つのリモートファイル) が選択されている場合は `true`、それ以外の場合は `false` のブール値。

site.canConnect()

対応バージョン
Dreamweaver 3

説明
Dreamweaver がリモートサイトに接続できるかどうかをチェックします。

引数
なし

戻り値
現在のリモートサイトが FTP サイトである場合は `true`、それ以外の場合は `false` のブール値。

site.canDisplaySyncInfoForFile()

対応バージョン
Dreamweaver CS3

説明
Dreamweaver が同期情報の表示の処理を実行できるかどうかをチェックします。

引数
`path`、'`site`'

- *`path`* はローカルファイルの URL です。
- '`site`' は、この関数が [サイト] パネルで選択されたファイルを使用することを示します。

戻り値
選択された 1 つのファイルがローカルファイルビューにあり、'site' がパラメータである場合、または渡された path が site の一部である場合は `true`。それ以外の場合は `false`。

site.canFindLinkSource()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [リンクソースの検索] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
サイトマップで選択したリンクが、ホームページではないかどうかを示すブール値。

site.canGet()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [GET] の処理を実行できるかどうかをチェックします。

引数
siteOrURL

- *siteOrURL* 引数には、この関数が [サイト] パネルの選択範囲に対して動作することを示す site キーワード、または 1 つのファイルの URL を指定する必要があります。

戻り値
引数に site を指定した場合は、[サイト] パネルで 1 つまたは複数のファイルかフォルダが選択されていて、リモートサイトが定義されているかどうかを示すブール値。引数に URL を指定した場合は、リモートサイトが定義されているサイトにドキュメントが所属するかどうかを示すブール値が返されます。

site.canLocateInSite()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が引数に応じて [ローカルサイトに配置] または [リモートサイトに配置] の処理を実行できるかどうかをチェックします。

引数
localOrRemote、*siteOrURL*

- *localOrRemote* 引数には、local か remote のいずれかを指定する必要があります。
- *siteOrURL* 引数には、この関数が [サイト] パネルの選択範囲に対して動作することを示す site キーワード、または 1 つのファイルの URL を指定する必要があります。

戻り値
次のいずれかの値を返します。

- 最初の引数が local キーワードで、2 番目の引数が URL の場合は、ドキュメントがサイトに所属するかどうかを示すブール値が返されます。
- 最初の引数が remote キーワードで、2 番目の引数が URL の場合は、リモートサイトが定義されているサイトにドキュメントが所属するかどうか、およびサーバータイプが [ローカル / ネットワーク] の場合は、ドライブがマウントされているかどうかを示すブール値が返されます。
- 2 番目の引数が site キーワードの場合は、両方のウィンドウにサイトマップではなくサイトファイルが含まれているかどうか、また選択範囲が引数と反対のペインにあるかどうかを示すブール値が返されます。

site.canMakeEditable()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [読み取り専用を解除する] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
Dreamweaver が [読み取り専用を解除する] の処理を実行できる場合は `true`、選択したファイルのいずれかがロックされている場合は `false` のブール値。

site.canMakeNewFileOrFolder()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [サイト] パネルで、[新規ファイル] または [新規フォルダ] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
[サイト] パネルの選択したペインに表示されているファイルがある場合は `true`、それ以外の場合は `false` のブール値。

site.canOpen()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が、[サイト] パネルで現在選択されているファイルまたはフォルダを開くことができるかどうかをチェックします。

引数
なし

戻り値
[サイト] パネルで選択されているファイルまたはフォルダがある場合は `true`、それ以外の場合は `false` のブール値。

site.canPut()

対応バージョン
Dreamweaver 3

説明

Dreamweaver が [PUT] の処理を実行できるかどうかをチェックします。

引数

siteOrURL

- *siteOrURL* 引数には、この関数が [サイト] パネルの選択範囲に対して動作することを示す site キーワード、または 1 つのファイルの URL を指定する必要があります。

戻り値

次のいずれかの値を返します。

- 引数に site キーワードを指定した場合は、[サイト] パネルでファイルかフォルダが選択されていて、リモートサイトが定義されているときに値 true が返され、それ以外の場合は false が返されます。
- 引数に URL を指定した場合は、リモートサイトが定義されているサイトにドキュメントが所属するときに値 true が返され、それ以外の場合は false が返されます。

site.canRecreateCache()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [サイトキャッシュの再作成] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

現在のサイトで、[キャッシュを使用して更新をスピードアップ] オプションが使用可能な場合は true のブール値。

site.canRefresh()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [ローカルの更新] または [リモートの更新] の処理を実行できるかどうかをチェックします。

引数

localOrRemote

- *localOrRemote* 引数には、local か remote のいずれかのキーワードを指定する必要があります。

戻り値

localOrRemote 引数に local キーワードが指定されている場合は true。それ以外の場合は、リモートサイトが定義されているかどうかを示すブール値。

site.canRemoveLink()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [リンクの削除] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
HTML または Flash ファイルから、サイトマップで選択したファイルにリンクが設定されているかどうかを示すブール値。

site.canSetLayout()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [レイアウト] の処理を実行できるかどうかをチェックします。

引数
なし

戻り値
サイトマップが表示されている場合は `true`、それ以外の場合は `false` のブール値。

site.canSelectAllCheckedOutFiles()

対応バージョン
Dreamweaver 4

説明
現在作業しているサイトで、チェックイン / チェックアウト機能が使用可能かどうかを判別します。

引数
なし

戻り値
サイトでチェックイン / チェックアウトが使用可能な場合は `true`、使用できない場合は `false` のブール値。

site.canSelectNewer()

対応バージョン
Dreamweaver 3

説明
Dreamweaver が [新しいリモートアイテムの選択] または [新しいローカルアイテムの選択] の処理を実行できるかどうかをチェックします。

引数

localOrRemote

- *localOrRemote* 引数には、local か remote のいずれかのキーワードを指定する必要があります。

戻り値

リモートサイトが定義されているサイトに、ドキュメントが所属するかどうかを示すブール値。

site.canShowPageTitles()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [ページタイトルの表示] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

サイトマップが表示されている場合は true、それ以外の場合は false のブール値。

site.canSynchronize()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [同期] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

リモートサイトが定義されているかどうかを示すブール値。

site.canUncloak()

対応バージョン

Dreamweaver MX

説明

Dreamweaver がクローク解除の処理を実行できるかどうかをチェックします。

引数

siteOrURL

- *siteOrURL* 引数には、canUncloak() 関数が [サイト] パネルの選択範囲に対して動作することを示す site キーワード、または canUncloak() 関数が指定されたフォルダとそのすべてのコンテンツに対して動作することを示す、特定のフォルダの URL を指定する必要があります。

戻り値

Dreamweaver が現在のサイトまたは指定したフォルダでクローク解除の操作を実行できる場合は `true`、それ以外の場合は `false` のブール値。

site.canUndoCheckOut()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [チェックアウトの取り消し] の処理を実行できるかどうかを判別します。

引数

`siteOrURL`

- *`siteOrURL`* 引数には、この関数が [サイト] パネルの選択範囲に対して動作することを示す `site` キーワード、または 1 つのファイルの URL を指定する必要があります。

戻り値

指定したファイルの少なくとも 1 つがチェックアウトされている場合は `true` のブール値。

site.canViewAsRoot()

対応バージョン

Dreamweaver 3

説明

Dreamweaver が [ルートとして表示] の処理を実行できるかどうかをチェックします。

引数

なし

戻り値

指定したファイルが HTML ファイルまたは Flash ファイルである場合は `true`、それ以外の場合は `false` のブール値。

索引

D

G

H

N

O

S

た

と

へ

ろ

わ