

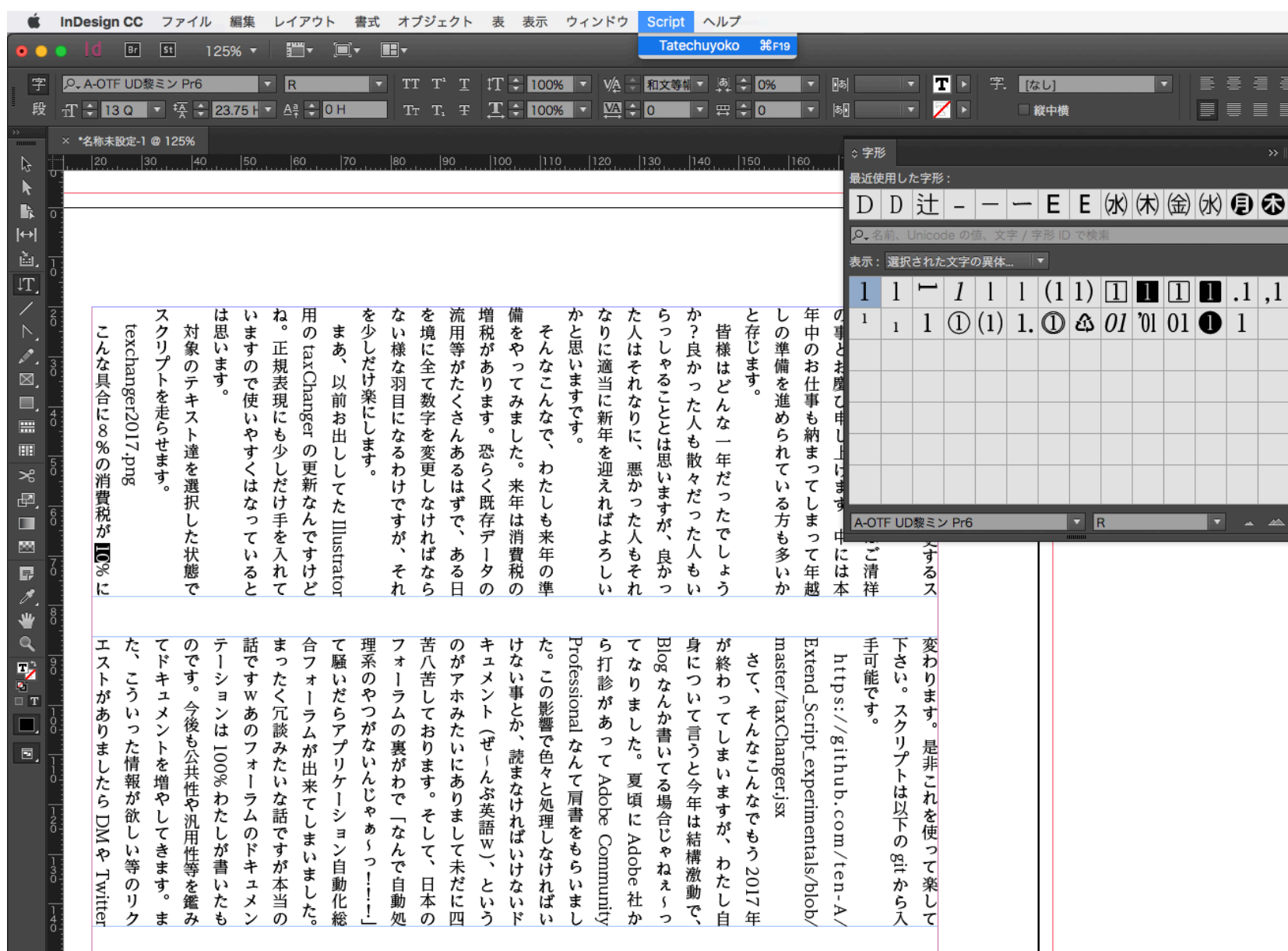


# [ID]スクリプトをメニューに組み込む方法

Posted by 10 A in アプリケーション自動化総合コミュニティフォーラム (Japan) on Feb 19, 2018 8:00:00 PM

今回用意したものはInDesignに対してカスタムメニューを追加し、そこにJavaScriptを組み込んでメニューのプルダウンやショートカットからスクリプトを実行させるようにしようという試みです。

InDesignのObjectModelをご覧いただければお分かりいただけると思いますが、InDesignではメニューに対する操作がしっかりと実装されています。このmenusやmenuActionと言ったオブジェクトを適切に操作すると下の様にメニューを追加することが可能です。



まず全体をご覧ください。

```
01. #targetengine "session"
02. var mn = app.menus.item("$ID/Main").submenus.item("Scripts");
03. mn.isValid && mn.remove();
04.
05.
06. var addCustomMenu = (function(){
```

```

07.     var customMenuAction = app.scriptMenuActions.add("Tatechuyoko");
08.     var customMenu = app.menus.item("$ID/Main").submenus.item("Scripts");
09.     var customMenuItem;
10.     if (!customMenu.isValid){
11.         customMenu = app.menus.item("$ID/Main").submenus.add("Script");
12.         customMenuItem = customMenu.menuItems.item("Tatechuyoko");
13.     }
14.     var fnc = function(){
15.         app.selection[0].tatechuyoko = true;
16.     }
17.
18.     var ev = customMenuAction.eventListeners.add("onInvoke", fnc);
19.     if (!customMenuItem.isValid){
20.         customMenu.menuItems.add(customMenuAction);
21.     }
22.     }());

```

```

01.     #targetengine "session"
02.     var mn = app.menus.item("$ID/Main").submenus.item("Scripts");
03.     mn.isValid && mn.remove();
04.
05.
06.     var addCustomMenu = (function(){
07.         var customMenuAction = app.scriptMenuActions.add("Tatechuyoko");
08.         var customMenu = app.menus.item("$ID/Main").submenus.item("Scripts");
09.         var customMenuItem;
10.         if (!customMenu.isValid){
11.             customMenu = app.menus.item("$ID/Main").submenus.add("Script");
12.             customMenuItem = customMenu.menuItems.item("Tatechuyoko");
13.         }
14.         var fnc = function(){
15.             app.selection[0].tatechuyoko = true;
16.         }
17.
18.         var ev = customMenuAction.eventListeners.add("onInvoke", fnc);
19.         if (!customMenuItem.isValid){
20.             customMenu.menuItems.add(customMenuAction);
21.         }
22.     }());

```

一行目のtargetengineについて少し解説を。

```
01.     #targetengine "session"
```

```
01.     #targetengine "session"
```

通常、ExtendScriptでは**main**エンジンにてスクリプトが実行されます。この**main**ではスクリプトの**実行終了後globalに置かれた変数なども全て破棄**されます。しかし、この**targetengine**を別のエンジン上で実行すると**スクリプト終了後もグローバル変数等も保持**されます。今回のスクリプトの様にイベント絡みでオブジェクトをGCさせたくない場合等にセットします。

また、このtargetディレクティブに関しては以下のドキュメントに詳しく説明されています。一度チェックしてみてください。

[Adobe ExtendScriptのプリプロセッサディレクティブ - chalcedonyの外部記憶装置・出張版](#)

続く 2 行を見えます。

```
01.  var mn = app.menus.item("$ID/Main").submenus.item("Scripts");
02.  mn.isValid && mn.remove();

01.  var mn = app.menus.item("$ID/Main").submenus.item("Scripts");
02.  mn.isValid && mn.remove();
```

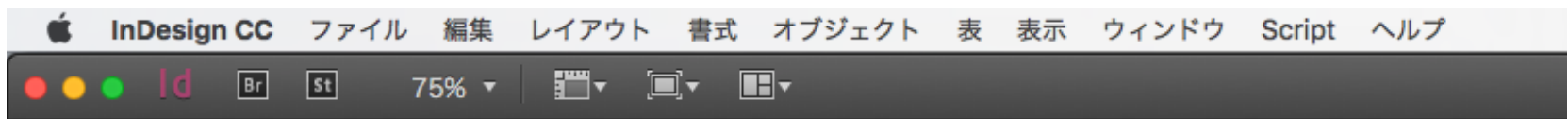
単純に既にScriptsというメニュー項目が追加されている場合はそれを初期化のために削除します。  
続く関数が本体です。メニューは基本的にメニュー項目の設定とそれに付随するアクションの設定にて成り立ちます。

```
01.  var customMenuAction = app.scriptMenuActions.add("Tatechuyoko");
01.  var customMenuAction = app.scriptMenuActions.add("Tatechuyoko");
```

こちらはアクションの定義です。しかし、実際の動作はまだ設定されていません。

```
01.  var customMenu = app.menus.item("$ID/Main").submenus.item("Scripts");
01.  var customMenu = app.menus.item("$ID/Main").submenus.item("Scripts");
```

こちらではMainのサブメニューとしてScriptsというサブメニューを参照しています。  
ちなみにmainのsubmenuというのは



こちらになります。ファイルや編集、表示と言ったトップレベルのメニュー達は全てこのmainのsubmenuクラスとなります。

先程の参照は一番はじめに**submenu**を取っ払った影響で**isValid**が**false**を返します。

```
01.  customMenu = app.menus.item("$ID/Main").submenus.add("Script");
01.  customMenu = app.menus.item("$ID/Main").submenus.add("Script");
```

こちらにて**Scripts**という**submenu**を追加します。

```
01.  customMenuItem = customMenu.menuItems.item("Tatechuyoko");
01.  customMenuItem = customMenu.menuItems.item("Tatechuyoko");
```

こちらは**Scripts**の子要素に対する参照を定義します。

```
01.  var fnc = function(){
02.    app.selection[0].tatechuyoko = true;
03.  }
04.  var ev = customMenuAction.eventListeners.add("onInvoke", fnc,);

01.  var fnc = function(){
02.    app.selection[0].tatechuyoko = true;
03.  }
04.  var ev = customMenuAction.eventListeners.add("onInvoke", fnc,);
```

fnc関数は実際にメニュー項目が選択された場合に実行される関数です。

続くevは先に生成した**customMenuAction**に対してイベントリスナを設定します。引数は基本的に2つでOKです。一つ目が**eventType**ですが、これは**onInvoke**のみでいけます。続くのが**ハンドラ**で先に作った関

数を割り当てます。

```
01. customMenu.menuItems.add(customMenuAction);
```

```
01. customMenu.menuItems.add(customMenuAction);
```

最後に**customMenuAction**自体も**isValid**が**false**なので**customMenu**の子要素として**add**します。

以上で一連の手順は終了です。この様に**追加したメニューであってもInDesignではショートカットを割り当てる事が可能**です。（スクリプトに割り当てるのは元々可能...）

上手く利用して使い勝手を良くしてください(^-^)/

968 Views  Tags : [javascript](#), [indesign](#), [menu](#), [extendscript](#)

## 0 Comments