

本スクリプトの特長

- 扱いが面倒臭い ScriptUI を少しだけ楽に生成できます。
- Window オブジェクトを用意し、そこに追加したいコントロール要素を配列で定義するだけで最低限の SUI を用意します。

スクリプトのダウンロード

スクリプト自体は GitHub にアップしてあります。

<https://github.com/UskeS/InDesign-Scripts/blob/master/SUI/createInterface2.jsx>

基本的な使い方

通常の SUI を生成するときと同じように、まずは Window オブジェクトを用意します。

```
#targetengine "SUITest"
var dlgObj = new Window("palette", "myDialog");
```

続いて、このダイアログに表示したいコントロールを配列として用意します。

```
var myControls = [
    "statictext: 静的テキスト",
    "et: 編集エリア",
    "button: ボタン"
];
```

このように定義したら、createInterface2 関数を呼び出します。引数は Window オブジェクト、コントロール要素を定義した配列、という順です。

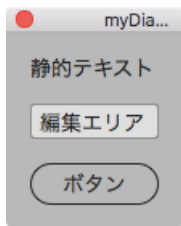
```
var mySUI = createInterface2(dlgObj, myControls);
```

これを show メソッドを使って表示するだけです。

```
dlgObj.show();
```

これだけで、右図のような SUI が生成できます。

これだけでは少し不格好なので、お好みで各レイアウト要素を調整してください。



コントロールの定義ルール

コントロールを配列として用意する際に以下のルールに従ってください。なお、すべて大文字・小文字は不問です。

[コントロールタイプ] : [表示するテキスト]

要するに、半角のコロンを挟んで左側がコントロールタイプ、右側がその表示内容ということです。コントロールタイプに用意している（利用できる）のは以下の 6 種類です。

- 静的テキスト : st、もしくは statictext
- 編集テキストエリア : et、もしくは edittext
- チェックボックス : cb、もしくは checkbox
- ラジオボタン : rb、もしくは radiobutton
- リスト : dd、もしくは dropdownlist
- ボタン : bt、もしくは button

これらのうち、リストとボタンだけは特殊な使い方があります。

リストの使い方

dropDownList コントロールにに表示するリストは、事前に配列で用意してください。例えばこのように定義します。

```
#targetengine "SUITest"
var dlgObj = new Window("palette", "myDialog");
var myList = ["123", "あいう", "XYZ"];
var myControls = [
    "statictext: リストから選んでください",
    "dd: myList", //<-配列を指定
    "button: ボタン"
];
var mySUI = createInterface2(dlgObj, myControls);
dlgObj.show();
```

このコードは右のような SUI を生成します。

※コントロールにリストを指定すると、無条件でリストの最初が選択された状態になります



ボタンの使い方

事前に関数を用意しておくことで、ボタンを押したとき（onClick イベント）の動作を指定することができます。その場合は、

[コントロールタイプ] : [表示するテキスト] | [関数名]

という形で記述してください。例えば、前述のコードのボタンに関数を追加すると以下ようになります。

```
#targetengine "SUITest"
var dlgObj = new Window("palette", "myDialog");
var myList = ["123", "あいう", "XYZ"];
var myFunc = function(){alert("ボタンを押しました!")};
var myControls = [
    "statictext: リストから選んでください",
    "dd: myList",
    "button: ボタン | myFunc" //<-関数を指定
];
var mySUI = createInterface2(dlgObj, myControls);
dlgObj.show();
```

ちなみに、半角コロンやパイプラインの前後のスペースはあってもなくても対応可能です。お好みでどうぞ。

パネルとしてグルーピングしたい

線で囲むパネルとして要素をまとめたい場合は、オブジェクトとして指定してください。その際のキーが、パネルとしてくった際の名称になります。

```
#targetengine "SUITest"
var dlgObj = new Window("palette", "myDialog");
var myList = ["123", "あいう", "XYZ"];
var myFunc = function(){alert("ボタンを押しました!")};
var myControls = [
    {"パネル表示":
```

```
[ "statictext: リストから選んでください",  
  "dd: myList" ]  
  
},  
  
"button: ボタン | myFunc"  
];  
  
var mySUI = createInterface2(dlgObj, myControls);  
dlgObj.show();
```

パネル表示を利用する場合も、オブジェクトの値自体は配列として定義することを忘れないでください。



各コントロールにアクセスするには？

この関数はすべてのコントロールが配列として戻ります。ですので、先のパネルを使った例では、変数 mySUI には

- 0：パネル
- 1：静的テキスト
- 2：リスト
- 3：ボタン

が入っています。



これを利用して各コントロールにアクセスして値を取得したり、コントロールの内容や状態をいじったりできます。

ということで、ひとまず説明はおしまいです。少しでもストレスフリーに SUI を利用していただければ本望です。

不具合の報告やご質問等はブログか Twitter でご連絡ください。基本的に無償公開しているものなので早急な対応はできない場合がありますが、致命的な不具合等であればなるべく早急に対応いたします。

注意事項

- 本スクリプトのライセンスは GitHub にあります。詳しくはライセンスをお読みください。
- そもそもこのスクリプトで SUI をすべて制御してやろうなどとは考えていません。あくまでよく使う小規模な SUI を生成する手間を減らそうと思って開発したものです。したがって iconButton や image など一部のコントロールには非対応です。今後対応する予定もありません。
- リストの配列の参照とボタンの関数の参照は eval メソッドを利用しています。
- ある程度カスタマイズしやすいように作ったつもりなので、自分用のオレオレ SUI 生成関数に仕立て上げてもらっても結構です。

改訂履歴

- 2018-08-17：v 0.1.0 公開