Illustrator エクステンション　処理作成

作成日:　2017/12/05
作成者: 中島

更新履歴

更新日	更新概要	作業者
2017/12/05	・　新規作成	中島
	・	
	・	
	・	
	・	

目次

ファイル構成

下記は初期のエクステンションのファイル構成です。エクステンションの処理は main.js と hostscript.jsx に書きます。

構成	概要
xxx/	ルートフォルダ
css/	CSS フォルダ
styles.css	index.html のデザインを調整します
CSXS/	CSXS フォルダ
manifest.xml	エクステンションの各種設定を入力します
icons/	エクステンションをアイコンパネル化した際に表示されるアイコンを入れます
js/	JavaScript フォルダ
lib/	ライブラリフォルダ(CSInterface 等)
main.js	エクステンションのパネル側で動作し、HTML の要素を操作・値を取得、ボタン押下等のイベント時に CSInterface を介して hostscript.jsx の関数を呼び出すなどの処理をします
themeManager.js	main.js で themeManager.init();を実行しておくと、メニューの「編集」→「環境設定」→「ユーザーインターフェース」→「明るさ」を変更した際に、エクステンションのパネルの色も合わせて変化するようになります
jsx/	JSX フォルダ
hostscript.jsx	ツール(Illustrator)側で動作し、ドキュメントへの線・文字の追加など、様々な処理をします
index.html	エクステンションのパネルのレイアウトです

main.js(エクステンションのパネル側)の処理

下記はパネル側処理の例です。hostscript.jsx 側の呼び出しだけでなく、通常の Web ページと同様の処理も実行できます。

```
/*jslint vars: true, plusplus: true, devel: true, nomen: true, regexp: true, indent: 4, maxerr: 50 */
/*global $, window, location, CSInterface, CSEvent, SystemPath, themeManager*/

(function () {
    'use strict';

    // アラートを出す
    alert("テスト");
    // id が"txt_test"の要素(テキストボックス)に"テスト"を設定
    $("#txt_test").val("テスト");

    // CSInterface のインスタンスを作成
    var csInterface = new CSInterface();

    // id が"btn_test"の要素(ボタン)をクリックした際の処理
    $("#btn_test").click(function () {
        // evalScript で hostscript.jsx の関数 testFunc を呼び出し
        csInterface.evalScript("testFunc()");
    });
}());
```

hostscript.jsx の関数の呼び出し

evalScript()の引数は文字列で、hostscript.jsx 側で実行したいコードそのものです。

関数は下記にように呼び出します。文字列の引数がある場合はダブルクオーテーションとエスケープも指定する必要があります。

```
// 引数なし
csInterface.evalScript("testFunc()");

// 引数が数値
var fontSize = 20;
csInterface.evalScript("testFunc2(" + fontSize + ")");

// 引数が文字列
var text = "ABC"
csInterface.evalScript("testFunc3(¥"" + text + "¥")");
```

evalScript()のコールバック関数

evalScript()の第 2 引数にコールバック関数を設定できます。

hostscript.jsx 側から値を受取ることもできます。値は文字列でしか受け取れないため、配列を文字列に変換して返してもらうなどの調整が必要なことがあります。

main.js

```
// Illustrator のフォント名一覧をプルダウンに設定
var $selectFont = $(“#select_font”);
var js = "getFontNames()";
csInterface.evalScript(js, function(fontNamesString){  // fontNamesString：戻り値
    var fontNames = fontNamesString.split(" ");
    $.each(fontNames, function(index, value){
        var $option = $("<option>").val(value).text(value);
        $selectFont.append($option);
    });
});
```

hostscript.js

```
// フォント名(内部名)一覧を取得する
function getFontNames(){
    var textFonts = app.textFonts; // フォント配列
    var fontNames = [];
    for(i = 0; i < textFonts.length; i++){
        // 内部名を取得
        fontNames.push(textFonts[i].name);
    }

    // 文字列しか返せないため空白で分割した文字列に変換
    return fontNames.join("  ");  // “Meiryo Meiryo-Bold Meiryo-Bolditalic …”
}
```

hostscript.jsx(Illustrator 側)の処理

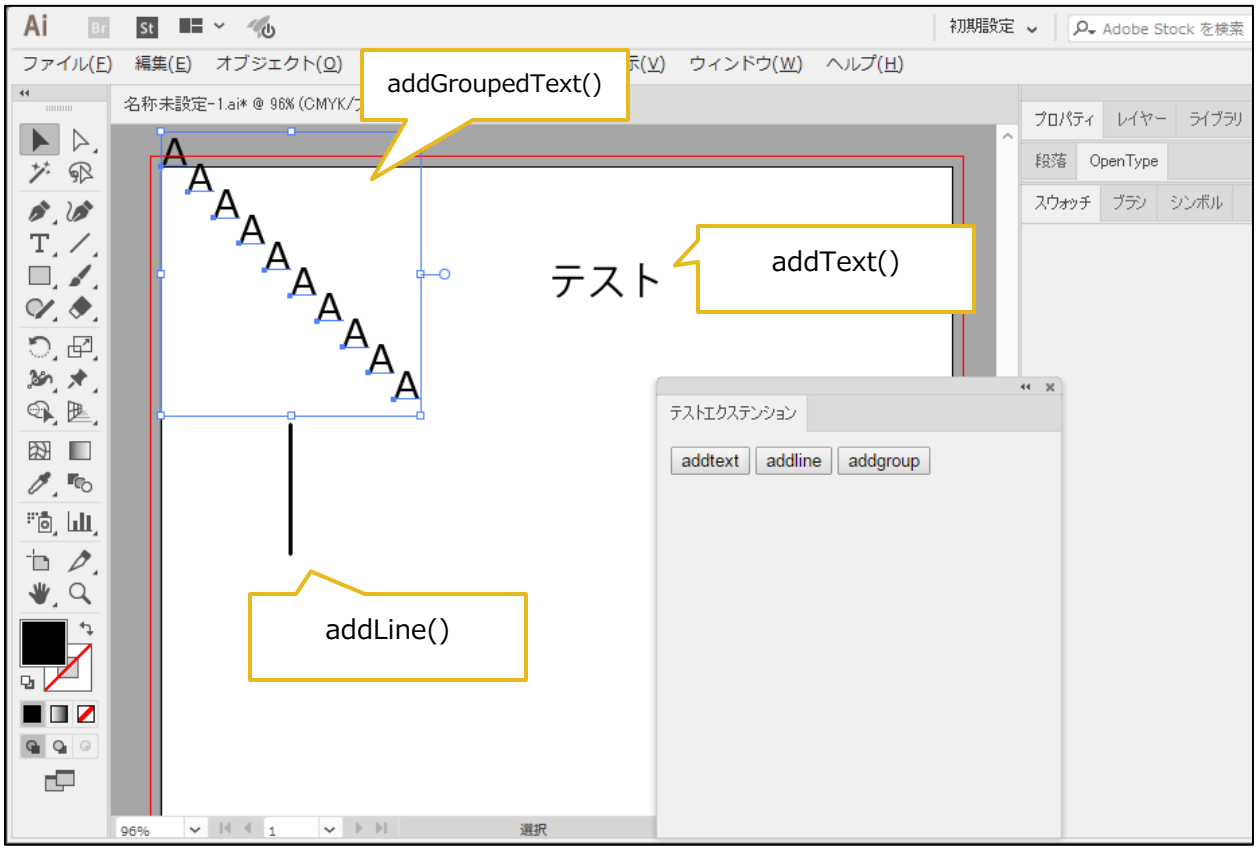
Illustrator を操作する処理を書きます。通常、処理は関数として用意しておき、main.js から呼び出して実行します。

処理例

```
// RGB で色を設定する(他の関数で使用)
function setColor(r,g,b){
    var tmpColor = new RGBColor();
    tmpColor.red = r;
    tmpColor.green = g;
    tmpColor.blue = b;
    return tmpColor;
}

// アラートを出す
function alertTest(){
    alert("テスト");
}

// 文字を追加する
function addText(text){
    // 現在のドキュメント
    var docObj = activeDocument;
    // 文字を追加する
    var textObj = docObj.textFrames.add();
    textObj.translate(300, -100); // 位置を指定(Y 座標は上が 0 で下に行くほどマイナスが大きくなる)
    textObj.contents = text; // 内容
    textObj.paragraphs[0].size = 30; // フォントサイズ(pt)
    textObj.paragraphs[0].textFont =  app.textFonts.getByName("Meiryo"); // フォント"
}
```

```
// 直線を追加する
function addLine(){
    var docObj = activeDocument;
    // 直線を追加する
    var pathStraightObj = docObj.pathItems.add();
    pathStraightObj.strokeColor = setColor(0, 0 ,0); // 黒
    pathStraightObj.strokeWidth = 3; // 線幅(pt)
    pathStraightObj.strokeCap = StrokeCap.ROUNDENDCAP; // 線端を丸にする
    // 縦の線になるようパスを設定
    pathStraightObj.setEntirePath([
        [100, -200],
        [100, -300]
    ]);
}

// グループ化された文字を追加する(“A”を斜めに 10 個並べる)
function addGroupedText(){
    // グループ化対象保持
    var groupObjects = [];
    // 現在のドキュメント
    var docObj = activeDocument;
    // 文字を追加する
    for(var i = 0; i < 10; i++){
        var textObj = docObj.textFrames.add();
        textObj.contents = "A"; // 内容
        textObj.paragraphs[0].size = 30; // フォントサイズ(pt)
        textObj.paragraphs[0].textFont =  app.textFonts.getByName("Meiryo"); // フォント”
        textObj.translate(i * 20, i * -20);
        groupObjects.push(textObj);
    }

    // グループ化
    var grp=docObj.groupItems.add();
    for(var i = 0; i < groupObjects.length ;i++){
        var obj = groupObjects[i]
        obj.move(grp,ElementPlacement.PLACEATEND);
    }
}
```

実行例


addGroupedText()
テスト
addText()
addLine()
テストエクステンション
addtext
addline
addgroup

参考になるサイト

- Illustrator Scripting | Adobe Developer Connection
  http://www.adobe.com/devnet/illustrator/scripting.html

  (「Adobe Illustrator CC 2017 Reference: JavaScript」の PDF 等)

- Illustrator CS 自動化作戦 with JavaScript
  http://www.openspc2.org/book/IllustratorCS/