

Omachi Nov 30, 2017 1:53 AM



MVP

InDesignの表で、セルのコンテンツが特定の文字列の場合に、セルの塗りを設定するスクリプトください（JavaScript）

This question has been **Answered**.

Adobe Forum Realで頂いた質問です。本来、スクリプトを一から作ってくれというのはあり得ないんですが、

「アプリケーション自動化総合コミュニティフォーラム」オープンセールということで持ち帰ってきました。

別に自分で回答しても構わないので、しばらくお付き合いください。

要件は次の通りです。

特定の文字列	スウォッチ名
あか	C=0 M=100 Y=100 K=0
あお	C=100 M=50 Y=0 K=0
きいろ	C=0 M=0 Y=100 K=0

のような、文字列とスウォッチの組み合わせがあるとします。

InDesignで表を作って、セルに文字を入れていきます。

ヘッダー 1	ヘッダー 2	ヘッダー 3	ヘッダー 4
しろ	235	適当な文字	あお
くろ	あか	あかじゃない	オレンジ
ももいろ	きいろ	あお	きいろ

こんな感じで入力が終わったら、スクリプトを実行します。

スクリプトは、セルに入っている文字を調べて、特定の文字列と同じだったらセルの塗りにスウォッチを指定します。

というものです。

これが物足りないという方には次のレベル。

「これを入力中に自動的に色が付くようにしてほしい」

いや、それをするときはずっとInDesignを監視していなければいけないので、ものすごく大変なんです。

といいましたら、納得して頂いたので、そこまで求められているわけではありません。

自己研鑽したい方はどうぞ(^^;

初めにも書きましたが、これは私が受けてきたので、最終的には私が責任をもって回答しますが、少し時間がかかります。

その前に「できたから書きたい」という方がおられましたらご自由にどうぞ、ということです(^^)



Correct Answer

by [Omachi](#) on Dec 3, 2017 6:34 PM

使用者に、セルの色（スウォッチ）と塗るための条件（特定の文字列に等しい）を複数指定してもらい、その指示に従って、セル内の文字列が指定された条件であれば塗りにスウォッチを適用するスクリプト

使用者側で、特定の文字列とスウォッチを指定するためのファイル。どちらの動作パターンでも使用します。ファイル名は何でもよい。

```
01. ##### 指定文字列とスウォッチの設定 #####
02. #
03. # スクリプト起動時にファイル選択ダイアログが出ますので、このファイルを選択してください。
04. #
05. # #で始まる行はコメントとして扱われます。
06. # また、空行は無視されます。
07. # 文字列とスウォッチ名をタブで区切って記述します。
08. # 例
09. # さくら      C=0 M=30 Y=0 K=0
10. #
11. あお      C=100 M=50 Y=0 K=0
12. あか      C=0 M=100 Y=100 K=0
13. きいろ    C=0 M=0 Y=100 K=0
```

既に入力が完了している状態のものに色を付ける場合

cellcolor.jsx

```
01. // セル内の文字が特定の文字列の場合に、セルの塗りを設定するスクリプト
02.
03. if (app.documents.length < 1) {
04.     alert("ドキュメントを開いてください");
05.     exit();
```

```

06. }
07.
08. var cset = [];
09. var tmp, i;
10. var doc = app.activeDocument;
11. var sws = doc.swatches;
12.
13. //定義ファイルの読み込み
14. var iniFile = File.openDialog("設定ファイルを指定してください");
15. iniFile.open("r");
16. while (!iniFile.eof){
17.     tmp = iniFile.readln().split("\t");
18.     tmp[0] = tmp[0].replace(/^\s+/, "").replace(/\s+$/, "");
19.     if (tmp[0] != "" && tmp[0].slice(0, 1) != "#") {
20.         tmp[1] = tmp[1].replace(/^\s+/, "").replace(/\s+$/, "");
21.         cset.push([tmp[0], tmp[1]]);
22.     }
23. }
24. iniFile.close();
25. //スウォッチの存在確認
26. tmp = "";
27. for (i = 0; i < cset.length; i++) {
28.     if (!sws.itemByName(cset[i][1]).isValid) {
29.         tmp += cset[i][1] + "\r";
30.     }
31. }
32. if (tmp != "") {
33.     alert("以下のスウォッチが見つかりません\r\r" + tmp +
34.         "\rドキュメントを確認するか、" + iniFile.name +
35.         "\rファイル中のスウォッチ名を確認してください");
36.     exit();
37. }
38. //選択状態の確認
39. var sel = doc.selection;
40. if (sel.length != 1) {
41.     alert("表のあるテキストフレームを1つ選択するか、\r" +
42.         "表内に文字カーソルを立ててください");
43.     exit();
44. }
45. //選択されているオブジェクト別に対象を特定して実行
46. switch (sel[0].constructor.name) {
47.     case "TextFrame":
48.         //ストーリー中の表全てを対象。テキストフレーム内に限定する場合はsel[0].tables
49.         var tbls = sel[0].parentStory.tables;
50.         if (tbls.length > 0) {
51.             for (i = 0; i < tbls.length; i++){
52.                 cellcolor(tbls[i]);
53.             }
54.         } else {
55.             tmp = "notfound";
56.         }
57.         break;
58.     case "Table":
59.         cellcolor(sel[0]);
60.         break;
61.     case "Cell":
62.         cellcolor(sel[0].parent);
63.         break;
64.     case "Text":
65.     case "TextStyleRange":
66.     case "TextColumn":

```

```

67.     case "Paragraph":
68.     case "Line":
69.     case "Word":
70.     case "Character":
71.     case "InsertionPoint":
72.         if (sel[0].parent.constructor.name == "Cell") {
73.             cellcolor(sel[0].parent.parent);
74.         } else {
75.             tmp = "notfound";
76.         }
77.         break;
78.     default:
79.         tmp = "notfound";
80. }
81. //終了処理
82. if (tmp == "notfound") {
83.     alert("表は見つかりませんでした");
84. }
85.
86. //セルに色を付ける
87. function cellcolor(targetTable) {
88.     var j, k;
89.     var ac = targetTable.cells;
90.     for (j = 0; j < ac.length; j++) {
91.         for (k = 0; k < cset.length; k++) {
92.             if (ac[j].contents == cset[k][0]) {
93.                 ac[j].fillColor = sws.itemByName(cset[k][1]);
94.                 break;
95.             }
96.         }
97.     }
98. }

```

入力中に自動で色を付ける場合

スクリプト実行中は、別のファイルを開かないでください。スウォッチの一致しないドキュメントでたまたま条件にあった場合にエラーが出るはずです。また、スクリプト実行中に別のドキュメントを開いて閉じると、追加したイベントが削除されるので、何らかのエラーやメッセージが出る可能性があります。その場合、もう一度スクリプトを実行してみてください。

監視動作を開始するスクリプト

cellcolor_start.jsx

```

01.     #targetengine "cellColorSession"
02.
03.     if (app.documents.length < 1) {
04.         alert("ドキュメントを開いてください");
05.         exit();
06.     }
07.
08.     var cset = [];
09.     var tmp, i;
10.     var doc = app.activeDocument;
11.     var sws = doc.swatches;
12.     var sws0 = sws.itemByName("None");
13.
14.     //定義ファイルの読み込み

```

```

15.     var iniFile = File.openDialog("設定ファイルを指定してください");
16.     iniFile.open("r");
17.     while (!iniFile.eof){
18.         tmp = iniFile.readLine().split("\t");
19.         tmp[0] = tmp[0].replace(/^\\s+/, "").replace(/\\s+$/, "");
20.         if (tmp[0] != "" && tmp[0].slice(0, 1) != "#") {
21.             tmp[1] = tmp[1].replace(/^\\s+/, "").replace(/\\s+$/, "");
22.             cset.push([tmp[0], tmp[1]]);
23.         }
24.     }
25.     iniFile.close();
26.     //スウォッチの存在確認
27.     tmp = "";
28.     for (i = 0; i < cset.length; i++) {
29.         if (!sws.itemByName(cset[i][1]).isValid) {
30.             tmp += cset[i][1] + "\\r";
31.         }
32.     }
33.     if (tmp != "") {
34.         alert("以下のスウォッチが見つかりません\\r\\r" + tmp +
35.             "\\rドキュメントを確認するか、" + iniFile.name +
36.             "\\rファイル中のスウォッチ名を確認してください");
37.         exit();
38.     }
39.
40.     //イベントを追加
41.     app.addEventListener(Event.AFTER_SELECTION_CHANGED, fillCellColor).name = "cellColorEvent";
42.
43.     function fillCellColor(ev){
44.         if (app.selection.length == 0) return;
45.         var parentObject = app.selection[0].parent;
46.         if (parentObject.constructor.name != "Cell") return;
47.
48.         var filled = false;
49.         for (var k = 0; k < cset.length; k++) {
50.             if (parentObject.contents == cset[k][0]) {
51.                 parentObject.fillColor = sws.itemByName(cset[k][1]);
52.                 filled = true;
53.                 break;
54.             }
55.         }
56.         if (!filled) parentObject.fillColor = sws0;
57.     }
58.
59.     //ドキュメントを閉じるときに追加したイベントを削除（これをしないと閉じるときにメッセージが表示される）
60.     app.addEventListener(Event.BEFORE_CLOSE, function (ev) {
61.         if (app.eventListeners.itemByName("cellColorEvent").isValid) {
62.             app.eventListeners.itemByName("cellColorEvent").remove();
63.         }
64.     });

```

監視動作を終了するスクリプト
途中でやめる場合に使用します。

cellcolor_stop.jsx

```

01.     if (app.eventListeners.itemByName("cellColorEvent").isValid) {
02.         app.eventListeners.itemByName("cellColorEvent").remove();
03.     }

```

3 HELPFUL

MOST LIKED

macneko.ayu Nov 30, 2017 6:21 AM (in response to Omachi)

Re: InDesignの表で、セルのコンテンツが特定の文字列の場合に、セルの塗りを設定するスクリプトください (JavaScript)

CS5 omachi さん

こんばんは。

下記のスクリプトでどうでしょうか。

テキストフレームを選択してから実行することを前提にしています。

セルの背景色にスウォッチを適用するところはスウォッチが存在するか確認してから適用させることが正解だと思ったのですが、ちょっと冗長かなあと思ったので try ~ catch で処理しています。

```
01. (function () {
02.     var docObj = app.activeDocument;
03.     if (app.selection[0].constructor.name !== "TextFrame") { return; }
04.     if (app.selection[0].tables.length === 0) { return; }
05.     app.selection[0].tables[0].select();
06.     var tableObj = app.selection[0];
07.     for (var i = 0, len = tableObj.cells.length; i < len; i++) {
08.         var cellObj = tableObj.cells[i];
09.         try {
10.             cellObj.fillColor = app.activeDocument.swatches.itemByName(cellObj.con
11.         } catch (e) {
12.
13.         }
14.     }
15. })();

01. (function () {
02.     var docObj = app.activeDocument;
03.     if (app.selection[0].constructor.name !== "TextFrame") { return; }
04.     if (app.selection[0].tables.length === 0) { return; }
05.     app.selection[0].tables[0].select();
06.     var tableObj = app.selection[0];
07.     for (var i = 0, len = tableObj.cells.length; i < len; i++) {
08.         var cellObj = tableObj.cells[i];
09.         try {
10.             cellObj.fillColor = app.activeDocument.swatches.itemByName(cellObj.con
11.         } catch (e) {
12.
13.         }
14.     }
15. })();
```

> 「これを入力中に自動的に色が付くようにしてほしい」

EventListener で監視するのがいいんですかねー。

時間あるときの宿題にしてみます。

1 person found this helpful

Actions ▾

 Edit

 Delete

 Report Abuse

 Liked (1)



ACP

10 A Nov 30, 2017 3:59 PM (in response to macneko.ayu)

Re: InDesignの表で、セルのコンテンツが特定の文字列の場合に、セルの塗りを設定するスクリプトください (JavaScript)

イベント操作の例なのですが、選択関連での操作には**AFTER_SELECTION_ATTRIBUTE_CHANGED**が便利かと思います。以下がそのサンプルです。

```
01. #targetengine "session"
02.
03. app.addEventListener(Event.AFTER_SELECTION_ATTRIBUTE_CHANGED , test).name = "e
04.
05. function test(ev){
06.     if (app.selection[0].parent.toString()!="[object Cell]") return;
07.     alert(app.selection[0].parent.contents);
08. }

01. #targetengine "session"
02.
03. app.addEventListener(Event.AFTER_SELECTION_ATTRIBUTE_CHANGED , test).name = "e
04.
05. function test(ev){
06.     if (app.selection[0].parent.toString()!="[object Cell]") return;
07.     alert(app.selection[0].parent.contents);
08. }
```

ただし、**AFTER_SELECTION_ATTRIBUTE_CHANGED**をそのまま取り回すと全ての操作においてイベントが発生しているため面倒なことになります。その為、Callback関数の冒頭でターゲット以外のものが選択されている場合に処理を中断する仕組みを組み込んでいます。

リスナにnameプロパティを当てているのはあとでitemByNameでリスナを停止させるためです。こうしておくとな下の様なワンライナーでイベントリスナを停止させることができます。

```
01. app.eventListeners.itemByName("eventTest").remove();
01. app.eventListeners.itemByName("eventTest").remove();
```

以上、イベント関連のTipsでした(^-^)/

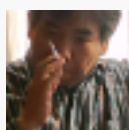
2 people found this helpful

Actions ▾

 Edit

 Delete

 Like (1)



Omachi Dec 2, 2017 6:50 AM (in response to 10 A)

MVP

Re: InDesignの表で、セルのコンテンツが特定の文字列の場合に、セルの塗りを設定するスクリプトください (JavaScript)

例文をやってみたんですが、**AFTER_SELECTION_ATTRIBUTE_CHANGED**だとツールの切り替え時にもイベントが発生してエラーになっちゃいました。そこで**AFTER_SELECTION_CHANGED**にしました。
結果はあとで投稿します。

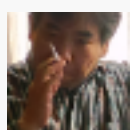
Actions

Edit

Delete

Report Abuse

Like (0)



Omachi Nov 30, 2017 4:18 PM (in response to macneko.ayu)

MVP

Re: InDesignの表で、セルのコンテンツが特定の文字列の場合に、セルの塗りを設定するスクリプトください (JavaScript)

>macneko.ayuさん

すみませんが、

```
01. cellObj.fillColor = app.activeDocument.swatches.itemByName(cellObj.contents);  
01. cellObj.fillColor = app.activeDocument.swatches.itemByName(cellObj.contents);
```

のところ勘違いしてます。

平たく書けば

```
01. if (cellObj.contents == "あか") {  
02.     cellObj.fillColor = app.activeDocument.swatches.itemByName("C=0 M=100 Y=1  
03. } else if (...  
01. if (cellObj.contents == "あか") {  
02.     cellObj.fillColor = app.activeDocument.swatches.itemByName("C=0 M=100 Y=1  
03. } else if (...
```

になります。

この、（条件=特定の文字列と、スウォッチ名のペア）は、あとから変更する可能性が高いので、変数（配列など）で先頭に定義しておく必要があります。

（もしくはファイルから読み込むとか）

よろしくお願いします。

あと、書き忘れてましたが、スウォッチは全て定義済みで、該当するスウォッチが見つからないというエラーは発生しないという前提です（さすがにそこまで面倒見切れません）。

Actions

Edit

Delete

Report Abuse

Like (0)

Re: InDesignの表で、セルのコンテンツが特定の文字列の場合に、セルの塗りを設定するスクリプトください (JavaScript)

前提を「テキストフレーム全体から探す」にした場合
ということで。case～switch文で分岐しました。



ヘッダー 1	ヘッダー 2	ヘッダー 3	ヘッダー 4
しろ	235	適当な文字	あお
くろ	あか	あかじゃない	オレンジ
ももいろ	きいろ	あお	きいろ

ヘッダー 1	ヘッダー 2	ヘッダー 3	ヘッダー 4
しろ	235	適当な文字	あお
くろ	あか	あかじゃない	オレンジ
ももいろ	きいろ	あお	きいろ

ヘッダー 1	ヘッダー 2	ヘッダー 3	ヘッダー 4
しろ	235	適当な文字	あお
くろ	あか	あかじゃない	オレンジ
ももいろ	きいろ	あお	きいろ

ヘッダー 1	ヘッダー 2	ヘッダー 3	ヘッダー 4
しろ	235	適当な文字	あお
くろ	あか	あかじゃない	オレンジ
ももいろ	きいろ	あお	きいろ

```
01. var Doc=app.activeDocument;
02.
03.
04. //スウォッチを定義
05. var red=Doc.swatches.itemByName("C=0 M=100 Y=100 K=0");//あか
```

```

06. var blue=Doc.swatches.itemByName("C=100 M=50 Y=0 K=0");//あお
07. var yellow=Doc.swatches.itemByName("C=0 M=0 Y=100 K=0");//きいろ
08.
09.
10. //テキストフレーム全体から探す
11. var txtFrms=Doc.textFrames;
12. var tfCount=txtFrms.length;
13. for(var i=0; i<tfCount; i++){
14.     var tbs=txtFrms[i].tables;
15.     var tbsCount=tbs.length;
16.     for(var j=0; j<tbsCount;j++){
17.         var ce=tbs[j].cells;
18.         var cellCount=ce.length;
19.         for(var k=cellCount-1; k>=0; k--){
20.             switch(ce[k].contents){
21.                 case "あか":
22.                     ce[k].fillColor=red;
23.                     break;
24.                 case "きいろ":
25.                     ce[k].fillColor=yellow;
26.                     break;
27.                 case "あお":
28.                     ce[k].fillColor=blue;
29.                     break;
30.                 default:
31.                     //当てはまらないならスルー
32.                     break;
33.             }
34.         }
35.     }

```

```

01. var Doc=app.activeDocument;
02.
03.
04. //スウォッチを定義
05. var red=Doc.swatches.itemByName("C=0 M=100 Y=100 K=0");//あか
06. var blue=Doc.swatches.itemByName("C=100 M=50 Y=0 K=0");//あお
07. var yellow=Doc.swatches.itemByName("C=0 M=0 Y=100 K=0");//きいろ
08.
09.
10. //テキストフレーム全体から探す
11. var txtFrms=Doc.textFrames;
12. var tfCount=txtFrms.length;
13. for(var i=0; i<tfCount; i++){
14.     var tbs=txtFrms[i].tables;
15.     var tbsCount=tbs.length;
16.     for(var j=0; j<tbsCount;j++){
17.         var ce=tbs[j].cells;
18.         var cellCount=ce.length;
19.         for(var k=cellCount-1; k>=0; k--){
20.             switch(ce[k].contents){
21.                 case "あか":
22.                     ce[k].fillColor=red;
23.                     break;
24.                 case "きいろ":
25.                     ce[k].fillColor=yellow;
26.                     break;
27.                 case "あお":
28.                     ce[k].fillColor=blue;
29.                     break;
30.                 default:
31.                     //当てはまらないならスルー

```

```
32.         break;
33.     }
34. }
35. }
```

1 person found this helpful

Actions ▾

 Edit

 Delete

 Report Abuse

 Liked (3)

Omachi Dec 2, 2017 7:05 AM (in response to Omachi)

Re: InDesignの表で、セルのコンテンツが特定の文字列の場合に、セルの塗りを設定するスクリプトください (JavaScript)

みなさん、書き込みありがとうございます。

しかしながら、私の提示の仕方が悪かったです。仕様をもう少し詰めてから書き込めばよかったと反省しています。

例示として「あか」「あお」「きいろ」としただけで、**実際にどの文字列で反応させるかはお客さんの方で別途指定する**というつもりだったんですが、伝わらなかったですね。

この後に私の回答を書き込みます。

- 既に入力が完了している状態のものに色を付ける場合
- 入力中に自動で色を付ける場合

の両方のパターンで作成できました。一応テストしたつもりですが、不備がありましたら教えてください。

「入力中に自動で色を付ける場合」は自分でも感動しました。是非参考にしてください。

私の回答には一応正解マークをつけておきますが、よりよい回答を書き込まれた場合はそちらを正解にします。

Actions ▾

 Edit

 Delete

 Report Abuse

 Like (0)

Omachi Dec 3, 2017 6:34 PM (in response to Omachi)

Correct Answer Re: InDesignの表で、セルのコンテンツが特定の文字列の場合に、セルの塗りを設定するスクリプトください (JavaScript)

使用者に、セルの色（スウォッチ）と塗るための条件（特定の文字列に等しい）を複数指定してもらい、

その指示に従って、セル内の文字列が指定された条件であれば塗りにスウォッチを適用するスクリプト

使用者側で、特定の文字列とスウォッチを指定するためのファイル。どちらの

動作パターンでも使用します。ファイル名は何でもよい。

```
01. ##### 指定文字列とスウォッチの設定 #####
02. #
03. # スクリプト起動時にファイル選択ダイアログが出ますので、このファイルを選択してください。
04. #
05. # #で始まる行はコメントとして扱われます。
06. # また、空行は無視されます。
07. # 文字列とスウォッチ名をタブで区切って記述します。
08. # 例
09. # さくら      C=0 M=30 Y=0 K=0
10. #
11. # あお        C=100 M=50 Y=0 K=0
12. # あか        C=0 M=100 Y=100 K=0
13. # きいろ      C=0 M=0 Y=100 K=0

01. ##### 指定文字列とスウォッチの設定 #####
02. #
03. # スクリプト起動時にファイル選択ダイアログが出ますので、このファイルを選択してください。
04. #
05. # #で始まる行はコメントとして扱われます。
06. # また、空行は無視されます。
07. # 文字列とスウォッチ名をタブで区切って記述します。
08. # 例
09. # さくら      C=0 M=30 Y=0 K=0
10. #
11. # あお        C=100 M=50 Y=0 K=0
12. # あか        C=0 M=100 Y=100 K=0
13. # きいろ      C=0 M=0 Y=100 K=0
```

既に入力が完了している状態のものに色を付ける場合

cellcolor.jsx

```
01. // セル内の文字が特定の文字列の場合に、セルの塗りを設定するスクリプト
02.
03. if (app.documents.length < 1) {
04.     alert("ドキュメントを開いてください");
05.     exit();
06. }
07.
08. var cset = [];
09. var tmp, i;
10. var doc = app.activeDocument;
11. var sws = doc.swatches;
12.
13. //定義ファイルの読み込み
14. var iniFile = File.openDialog("設定ファイルを指定してください");
15. iniFile.open("r");
16. while (!iniFile.eof){
17.     tmp = iniFile.readln().split("\t");
18.     tmp[0] = tmp[0].replace(/\s+/, "").replace(/\s+$/, "");
19.     if (tmp[0] != "" && tmp[0].slice(0, 1) != "#") {
20.         tmp[1] = tmp[1].replace(/\s+/, "").replace(/\s+$/, "");
21.         cset.push([tmp[0], tmp[1]]);
22.     }
23. }
24. iniFile.close();
25. //スウォッチの存在確認
26. tmp = "";
27. for (i = 0; i < cset.length; i++) {
28.     if (!sws.itemByName(cset[i][1]).isValid) {
29.         tmp += cset[i][1] + "\r";
```

```

30.     }
31. }
32. if (tmp != "") {
33.     alert("以下のスウォッチが見つかりません\r\r" + tmp +
34.         "\rドキュメントを確認するか、" + iniFile.name +
35.         "\rファイル中のスウォッチ名を確認してください");
36.     exit();
37. }
38. //選択状態の確認
39. var sel = doc.selection;
40. if (sel.length != 1) {
41.     alert("表のあるテキストフレームを1つ選択するか、\r" +
42.         "表内に文字カーソルを立ててください");
43.     exit();
44. }
45. //選択されているオブジェクト別に対象を特定して実行
46. switch (sel[0].constructor.name) {
47.     case "TextFrame":
48.         //ストーリー中の表全てを対象。テキストフレーム内に限定する場合はsel[0].table
49.         var tbls = sel[0].parentStory.tables;
50.         if (tbls.length > 0) {
51.             for (i = 0; i < tbls.length; i++){
52.                 cellcolor(tbls[i]);
53.             }
54.         } else {
55.             tmp = "notfound";
56.         }
57.         break;
58.     case "Table":
59.         cellcolor(sel[0]);
60.         break;
61.     case "Cell":
62.         cellcolor(sel[0].parent);
63.         break;
64.     case "Text":
65.     case "TextStyleRange":
66.     case "TextColumn":
67.     case "Paragraph":
68.     case "Line":
69.     case "Word":
70.     case "Character":
71.     case "InsertionPoint":
72.         if (sel[0].parent.constructor.name == "Cell") {
73.             cellcolor(sel[0].parent.parent);
74.         } else {
75.             tmp = "notfound";
76.         }
77.         break;
78.     default:
79.         tmp = "notfound";
80. }
81. //終了処理
82. if (tmp == "notfound") {
83.     alert("表は見つかりませんでした");
84. }
85.
86. //セルに色を付ける
87. function cellcolor(targetTable) {
88.     var j, k;
89.     var ac = targetTable.cells;
90.     for (j = 0; j < ac.length; j++) {

```

```

91.         for (k = 0; k < cset.length; k++) {
92.             if (ac[j].contents == cset[k][0]) {
93.                 ac[j].fillColor = sws.itemByName(cset[k][1]);
94.                 break;
95.             }
96.         }
97.     }
98. }

01. // セル内の文字が特定の文字列の場合に、セルの塗りを設定するスクリプト
02.
03. if (app.documents.length < 1) {
04.     alert("ドキュメントを開いてください");
05.     exit();
06. }
07.
08. var cset = [];
09. var tmp, i;
10. var doc = app.activeDocument;
11. var sws = doc.swatches;
12.
13. //定義ファイルの読み込み
14. var iniFile = File.openDialog("設定ファイルを指定してください");
15. iniFile.open("r");
16. while (!iniFile.eof){
17.     tmp = iniFile.readln().split("\t");
18.     tmp[0] = tmp[0].replace(/^\s+/, "").replace(/\s+$/, "");
19.     if (tmp[0] != "" && tmp[0].slice(0, 1) != "#") {
20.         tmp[1] = tmp[1].replace(/^\s+/, "").replace(/\s+$/, "");
21.         cset.push([tmp[0], tmp[1]]);
22.     }
23. }
24. iniFile.close();
25. //スウォッチの存在確認
26. tmp = "";
27. for (i = 0; i < cset.length; i++) {
28.     if (!sws.itemByName(cset[i][1]).isValid) {
29.         tmp += cset[i][1] + "\r";
30.     }
31. }
32. if (tmp != "") {
33.     alert("以下のスウォッチが見つかりません\r\r" + tmp +
34.         "\rドキュメントを確認するか、" + iniFile.name +
35.         "\rファイル中のスウォッチ名を確認してください");
36.     exit();
37. }
38. //選択状態の確認
39. var sel = doc.selection;
40. if (sel.length != 1) {
41.     alert("表のあるテキストフレームを1つ選択するか、\r" +
42.         "表内に文字カーソルを立ててください");
43.     exit();
44. }
45. //選択されているオブジェクト別に対象を特定して実行
46. switch (sel[0].constructor.name) {
47.     case "TextFrame":
48.         //ストーリー中の表全てを対象。テキストフレーム内に限定する場合はsel[0].table
49.         var tbls = sel[0].parentStory.tables;
50.         if (tbls.length > 0) {
51.             for (i = 0; i < tbls.length; i++){
52.                 cellcolor(tbls[i]);
53.             }

```



```

54.     } else {
55.         tmp = "notfound";
56.     }
57.     break;
58. case "Table":
59.     cellcolor(sel[0]);
60.     break;
61. case "Cell":
62.     cellcolor(sel[0].parent);
63.     break;
64. case "Text":
65. case "TextStyleRange":
66. case "TextColumn":
67. case "Paragraph":
68. case "Line":
69. case "Word":
70. case "Character":
71. case "InsertionPoint":
72.     if (sel[0].parent.constructor.name == "Cell") {
73.         cellcolor(sel[0].parent.parent);
74.     } else {
75.         tmp = "notfound";
76.     }
77.     break;
78. default:
79.     tmp = "notfound";
80. }
81. //終了処理
82. if (tmp == "notfound") {
83.     alert("表は見つかりませんでした");
84. }
85.
86. //セルに色を付ける
87. function cellcolor(targetTable) {
88.     var j, k;
89.     var ac = targetTable.cells;
90.     for (j = 0; j < ac.length; j++) {
91.         for (k = 0; k < cset.length; k++) {
92.             if (ac[j].contents == cset[k][0]) {
93.                 ac[j].fillColor = sws.itemByName(cset[k][1]);
94.                 break;
95.             }
96.         }
97.     }
98. }

```

入力中に自動で色を付ける場合

スクリプト実行中は、別のファイルを開かないでください。スウォッチの一致しないドキュメントでたまたま条件にあった場合にエラーが出るはずです。また、スクリプト実行中に別のドキュメントを開いて閉じると、追加したイベントが削除されるので、何らかのエラーやメッセージが出る可能性があります。その場合、もう一度スクリプトを実行してみてください。

監視動作を開始するスクリプト

cellcolor_start.jsx

```

01. #targetengine "cellColorSession"

```

```
02.
03.   if (app.documents.length < 1) {
04.       alert("ドキュメントを開いてください");
05.       exit();
06.   }
07.
08.   var cset = [];
09.   var tmp, i;
10.   var doc = app.activeDocument;
11.   var sws = doc.swatches;
12.   var sws0 = sws.itemByName("None");
13.
14.   //定義ファイルの読み込み
15.   var iniFile = File.openDialog("設定ファイルを指定してください");
16.   iniFile.open("r");
17.   while (!iniFile.eof){
18.       tmp = iniFile.readln().split("\t");
19.       tmp[0] = tmp[0].replace(/^\s+/, "").replace(/\s+$/, "");
20.       if (tmp[0] != "" && tmp[0].slice(0, 1) != "#") {
21.           tmp[1] = tmp[1].replace(/^\s+/, "").replace(/\s+$/, "");
22.           cset.push([tmp[0], tmp[1]]);
23.       }
24.   }
25.   iniFile.close();
26.   //スウォッチの存在確認
27.   tmp = "";
28.   for (i = 0; i < cset.length; i++) {
29.       if (!sws.itemByName(cset[i][1]).isValid) {
30.           tmp += cset[i][1] + "\r";
31.       }
32.   }
33.   if (tmp != "") {
34.       alert("以下のスウォッチが見つかりません\r\r" + tmp +
35.           "\rドキュメントを確認するか、" + iniFile.name +
36.           "\rファイル中のスウォッチ名を確認してください");
37.       exit();
38.   }
39.
40.   //イベントを追加
41.   app.addEventListener(Event.AFTER_SELECTION_CHANGED, fillCellColor).name = "cel
42.
43.   function fillCellColor(ev){
44.       if (app.selection.length == 0) return;
45.       var parentObject = app.selection[0].parent;
46.       if (parentObject.constructor.name != "Cell") return;
47.
48.       var filled = false;
49.       for (var k = 0; k < cset.length; k++) {
50.           if (parentObject.contents == cset[k][0]) {
51.               parentObject.fillColor = sws.itemByName(cset[k][1]);
52.               filled = true;
53.               break;
54.           }
55.       }
56.       if (!filled) parentObject.fillColor = sws0;
57.   }
58.
59.   //ドキュメントを閉じるときに追加したイベントを削除（これをしないと閉じるときにメッセー
60.   app.addEventListener(Event.BEFORE_CLOSE, function (ev) {
61.       if (app.eventListeners.itemByName("cellColorEvent").isValid) {
62.           app.eventListeners.itemByName("cellColorEvent").remove();
```

```

63.     }
64. });
01. #targetengine "cellColorSession"
02.
03. if (app.documents.length < 1) {
04.     alert("ドキュメントを開いてください");
05.     exit();
06. }
07.
08. var cset = [];
09. var tmp, i;
10. var doc = app.activeDocument;
11. var sws = doc.swatches;
12. var sws0 = sws.itemByName("None");
13.
14. //定義ファイルの読み込み
15. var iniFile = File.openDialog("設定ファイルを指定してください");
16. iniFile.open("r");
17. while (!iniFile.eof){
18.     tmp = iniFile.readln().split("\t");
19.     tmp[0] = tmp[0].replace(/^\\s+/, "").replace(/\\s+$/, "");
20.     if (tmp[0] != "" && tmp[0].slice(0, 1) != "#") {
21.         tmp[1] = tmp[1].replace(/^\\s+/, "").replace(/\\s+$/, "");
22.         cset.push([tmp[0], tmp[1]]);
23.     }
24. }
25. iniFile.close();
26. //スウォッチの存在確認
27. tmp = "";
28. for (i = 0; i < cset.length; i++) {
29.     if (!sws.itemByName(cset[i][1]).isValid) {
30.         tmp += cset[i][1] + "\\r";
31.     }
32. }
33. if (tmp != "") {
34.     alert("以下のスウォッチが見つかりません\\r\\r" + tmp +
35.         "\\rドキュメントを確認するか、" + iniFile.name +
36.         "\\rファイル中のスウォッチ名を確認してください");
37.     exit();
38. }
39.
40. //イベントを追加
41. app.addEventListener(Event.AFTER_SELECTION_CHANGED, fillCellColor).name = "cel
42.
43. function fillCellColor(ev){
44.     if (app.selection.length == 0) return;
45.     var parentObject = app.selection[0].parent;
46.     if (parentObject.constructor.name != "Cell") return;
47.
48.     var filled = false;
49.     for (var k = 0; k < cset.length; k++) {
50.         if (parentObject.contents == cset[k][0]) {
51.             parentObject.fillColor = sws.itemByName(cset[k][1]);
52.             filled = true;
53.             break;
54.         }
55.     }
56.     if (!filled) parentObject.fillColor = sws0;
57. }
58.
59. //ドキュメントを閉じるときに追加したイベントを削除（これをしないと閉じるときにメッセー

```

```

60. app.addListener(Event.BEFORE_CLOSE, function (ev) {
61.     if (app.eventListeners.itemByName("cellColorEvent").isValid) {
62.         app.eventListeners.itemByName("cellColorEvent").remove();
63.     }
64. });

```

監視動作を終了するスクリプト
途中でやめる場合に使用します。

cellcolor_stop.jsx

```

01. if (app.eventListeners.itemByName("cellColorEvent").isValid) {
02.     app.eventListeners.itemByName("cellColorEvent").remove();
03. }
01. if (app.eventListeners.itemByName("cellColorEvent").isValid) {
02.     app.eventListeners.itemByName("cellColorEvent").remove();
03. }

```

Actions ▾

 Edit

 Delete

 Report Abuse

 Like (0)



Omachi Dec 3, 2017 6:36 PM (in response to Omachi)

Re: InDesignの表で、セルのコンテンツが特定の文字列の場合に、セルの塗りを設定するスクリプトください (JavaScript)

MVP

12月4日 追記

cellcolor_start.jsxの44行目に

```

01. if (app.selection.length == 0) return;
01. if (app.selection.length == 0) return;

```

を追加しました。何も選択しない場合にエラーが出るのを修正しました。

Actions ▾

 Edit

 Delete

 Report Abuse

 Liked (1)



Omachi Dec 14, 2017 12:43 AM (in response to Omachi)

Re: InDesignの表で、セルのコンテンツが特定の文字列の場合に、セルの塗りを設定するスクリプトください (JavaScript)

MVP

追記です。ファイルを開くダイアログで「キャンセル」された場合の処理が抜けていますので、

```

01. var iniFile = File.openDialog("設定ファイルを指定してください");
01. var iniFile = File.openDialog("設定ファイルを指定してください");

```

のあとに、次を書き加えてください。

```

01. if (iniFile == null) {
02.     alert("キャンセルされました");
03.     exit();
04. }
01. if (iniFile == null) {
02.     alert("キャンセルされました");
03.     exit();

```

04. }

Actions

 Edit

 Delete

 Report Abuse

 Liked (1)