



```

with (aiOptions) {
    pdfCompatible = true ; //PDF 互換ファイルを作成→チェックオン
    embedLinkedFiles = true ; //配置した画像を含む→チェックオン
    embedICCPProfile = true ; //ICC プロファイルを埋め込む→チェックオン
    compressed = true ; //圧縮を使用する→チェックオン
}

//-----//
var abNum = 1; // 100 ページ
var pageW = 1714.11; // サイズ横幅
var pageH = 1735.99; // サイズ縦幅
//-----//
var docObj = app.documents.add(
    DocumentColorSpace.RGB,
    pageW, // A4 サイズ
    pageH,
    abNum,
    DocumentArtboardLayout.GridByRow, // 左から右、上から下
    0, // アートボード間の余白
    10 // アートボードサイズの平方根。Math.SQRT(abNum)
);

//foldername2 = Folder.selectDialog("読み込みフォルダを指定してください");

var folderPath="D:/_tori/Set/"+folderPathIndex;//作業フォルダパス
var setFolderPath="D:/_tori/Set";//保存フォルダパス
var foldername = (new Folder(folderPath));//作業フォルダ
var foldername2 = (new Folder(setFolderPath));//保存フォルダ

if (foldername)
{
    //保存先を選択
    var saveObjNameFolder = foldername2;
    //末尾に区切りのスラッシュが抜けているので足しておく
    var saveObj = saveObjNameFolder + "/";

    //フォルダ名用に変数を用意する
    var saveObjNameFolder2 = foldername;
    //末尾に区切りのスラッシュが抜けているので足しておく
    var saveObj2 = saveObjNameFolder2 + "/";

    // 保存先のフォルダ名を抽出

```

```

var folderName = saveObj2.split("/");

// ファイルネームの先頭にフォルダ名を持ってくる
setFolderName=folderName[4]; //作業フォルダの位置

// alert(""+ folderName[2]);

// AI ファイル 配置
ext = ["*.ai"];
for (var j=0; j<ext.length; j++)
{
    fileList = foldername.GetFiles(ext[j]);
    layoutImage(fileList,docObj);
}
}

//-----//
//すべてを選択
//-----//
app.doScript("すべてを選択", "全てを選択");
//alert("1 ok");

//-----//
//アートボードセンター合流
//-----//
app.doScript("アートボードの中央に配置", "アートボード中央→合流");
app.doScript("合流", "アートボード中央→合流");

app.activeDocument.selection = null; //選択解除
//alert("2 ok");
//-----//

arrayFileName.reverse(); // 配列の順番を反転
number = arrayFileName.join("x"); // 区切り文字でつなげる

//ファイルネームの表示
//alert(""+ arrayFileName);
//alert(""+ number);

//ドキュメント名を取得
//var fsName = docObj.name ;
//var fsArray = fsName.split(".") ;

```

```
//var aiName = fsArray[0]; //ドキュメント名取得できた

//ファイル生成
var saveFile = new File(saveObj + "s"+setFolderName + "ss" + number + ".ai");

//保存
docObj.saveAs(saveFile, aiOptions);

//ドキュメントを閉じる
//app.activeDocument.close();

/*****/

//アラートを無効にする
app.userInteractionLevel=UserInteractionLevel.DONTDISPLAYALERTS;

var color = new CMYKColor;

//12
color.cyan = 35;
color.magenta = 100;
color.yellow = 35;
color.black = 10;
app.activeDocument.defaultFillColor = color;

app.doScript("選択 共通 塗り", "共通塗り→ロック");
app.doScript("ロック", "共通塗り→ロック");

//10
color.cyan = 75;
color.magenta = 0;
color.yellow = 75;
color.black = 0;
app.activeDocument.defaultFillColor = color;

app.doScript("選択 共通 塗り", "共通塗り→ロック");
app.doScript("ロック", "共通塗り→ロック");

//9
color.cyan = 0;
color.magenta = 90;
color.yellow = 85;
```

```
color.black = 0;
app.activeDocument.defaultFillColor = color;

app.doScript("選択 共通 塗り", "共通塗り→ロック");
app.doScript("ロック", "共通塗り→ロック");

//8
color.cyan = 0;
color.magenta = 95;
color.yellow = 20;
color.black = 0;
app.activeDocument.defaultFillColor = color;

app.doScript("選択 共通 塗り", "共通塗り→ロック");
app.doScript("ロック", "共通塗り→ロック");

//7
color.cyan = 35;
color.magenta = 60;
color.yellow = 80;
color.black = 25;
app.activeDocument.defaultFillColor = color;

app.doScript("選択 共通 塗り", "共通塗り→ロック");
app.doScript("ロック", "共通塗り→ロック");

//6
color.cyan = 100;
color.magenta = 100;
color.yellow = 0;
color.black = 0;
app.activeDocument.defaultFillColor = color;

app.doScript("選択 共通 塗り", "共通塗り→ロック");
app.doScript("ロック", "共通塗り→ロック");

//5
color.cyan = 80;
color.magenta = 10;
color.yellow = 45;
color.black = 0;
app.activeDocument.defaultFillColor = color;
```

```
app.doScript("選択 共通 塗り", "共通塗り→ロック");  
app.doScript("ロック", "共通塗り→ロック");
```

```
//4
```

```
color.cyan = 20;  
color.magenta = 0;  
color.yellow = 100;  
color.black = 0;  
app.activeDocument.defaultFillColor = color;
```

```
app.doScript("選択 共通 塗り", "共通塗り→ロック");  
app.doScript("ロック", "共通塗り→ロック");
```

```
//3
```

```
color.cyan = 0;  
color.magenta = 80;  
color.yellow = 95;  
color.black = 0;  
app.activeDocument.defaultFillColor = color;
```

```
app.doScript("選択 共通 塗り", "共通塗り→ロック");  
app.doScript("ロック", "共通塗り→ロック");
```

```
//2
```

```
color.cyan = 85;  
color.magenta = 50;  
color.yellow = 0;  
color.black = 0;  
app.activeDocument.defaultFillColor = color;
```

```
app.doScript("選択 共通 塗り", "共通塗り→ロック");  
app.doScript("ロック", "共通塗り→ロック");
```

```
//1
```

```
color.cyan = 40;  
color.magenta = 45;  
color.yellow = 50;  
color.black = 5;  
app.activeDocument.defaultFillColor = color;
```

```
app.doScript("選択 共通 塗り", "共通塗り→ロック");
```

```
app.doScript("ロック", "共通塗り→ロック");

//11
/*
var color2 = new RGBColor;
color2.red = 111;
color2.green = 111;
color2.blue = 112;
app.activeDocument.defaultFillColor = color2;
*/
color.cyan = 60;
color.magenta = 50;
color.yellow = 50;
color.black = 5;
app.activeDocument.defaultFillColor = color;

app.doScript("選択 共通 塗り", "共通塗り→ロック");
app.doScript("ロック", "共通塗り→ロック");

/*****/

//選択を解除する
app.activeDocument.selection = null;

//塗り潰しカラーを「なし」に指定する
var color0 = new NoColor;
activeDocument.defaultFillColor = color0;

app.doScript("カラーを線にする", "線を選択する");

//塗り潰しカラーを「なし」に指定する
activeDocument.defaultFillColor = color0;

//塗りと線を選択して消去する
app.doScript("塗りと線の選択と消去", "塗りと線を選択して消去する");

//ロックを解除
mydocument = app.activeDocument;

// レイヤーをカウントする。
layercount = mydocument.layers.length;
```

```

var visibleflag = [];

// レイヤーの表示状態を記憶し、レイヤーの表示、ロック解除を行う。
for( i=0; i<layercount; i++ ){
    mylayer = mydocument.layers[i];
    visibleflag[i] = mylayer.visible;
    mylayer.visible = true;
    mylayer.locked = false;
}

// テキストフレームをカウントする。
for (n=0; n<mydocument.pageItems.length; n++){
    mydocument.pageItems[n].locked = false;
}

// レイヤーの表示状態を元に戻し、ロックする。
for( m=0; m<layercount; m++){
    mylayer = mydocument.layers[m];
    if( visibleflag[m] != true ){
        mylayer.visible = false;
    }
}

//ドキュメントの保存
activeDocument.save();

//ドキュメントを閉じる
activeDocument.close(SaveOptions.DONOTSAVECHANGES);
////////////////////////////////////(1)
}
////////////////////////////////////(1)

//=====
//レイアウトイメージ
//=====
function layoutImage(fileList,docObj)
////////////////////////////////////(2)
{
////////////////////////////////////(2)
    for (i=0; i<fileList.length; i++)
    {
        pItem = docObj.placedItems.add();

```



```

fs = new File(fileList[i].fsName);

var objNumWk = fileList[i].fsName.split("-");

//alert(""+ objNumWk);

var objNum = objNumWk[1].split(".");

//alert(""+ objNum);

// ファイルネームにファイル番号を保存していく
arrayFileName[i]=objNum[0];

//alert(""+ objNum[0]);

pItem.file = fs;
//pItem.width = w;
//pItem.height = h;
pItem.embed();          //埋め込み画像にする
//pItem.position = new Array(baseX + x,baseY - y);
x = x + w;
cntX++;
if (cntX >= maxCountX)
{
    cntX = 0;
    x = startX;
    y = y + h + boxHeight;
    cntY++;
    if (cntY >= maxCountY)
    {
        cntY = 0;
        y = startY;
        docObj = app.documents.add();
    }
}
}
}
////////////////////(2)
}
////////////////////(2)

```