TUTORIEL N° 23

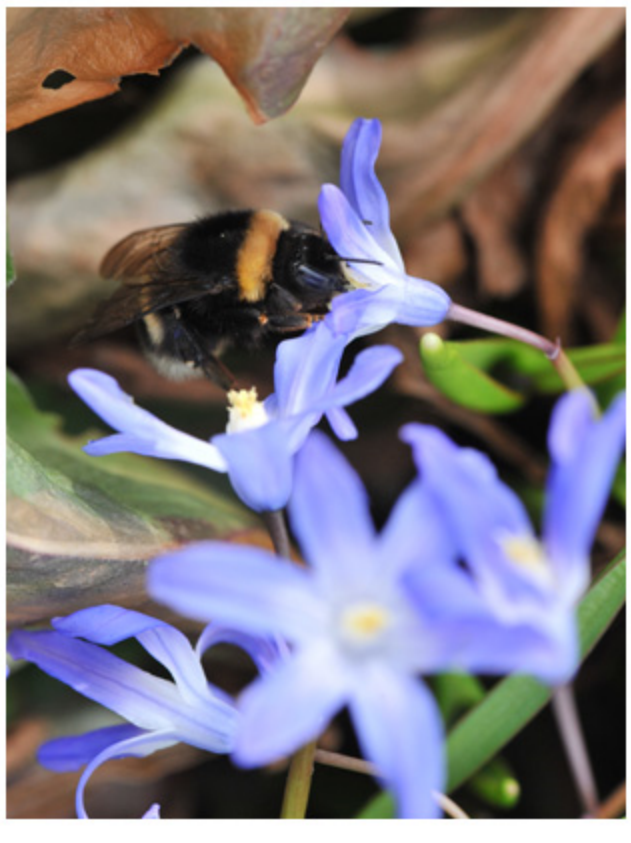
Les scripts : initiation à Javascript pour Photoshop CS 5.0

Préliminaires

Comme cela a été dit dans le tutoriel n° 22, il existe depuis de très nombreuses années une ambiguïté au niveau de la terminologie entre les notions d'«**actions**» et de «**scripts**». Si vous cliquez sur **Fichier...** dans le menu de Photoshop CS 5.0, vous trouverez la fonction «**Scripts**» au bas de la liste. De quoi s'agit-il ?

Photoshop est fourni depuis de nombreuses années avec trois langages de script :
- Applescript pour la plateforme Mac OS,
- VBscript pour Windows (Microsoft) et
- Javascript qui est le seul à être commun aux mondes Mac et Windows.

C'est à ce langage que vous allez être initié. L'objectif de ce tutoriel n'est pas de faire de vous un programmeur de scripts chevronné, mais de vous faire découvrir son existence et la puissance de la programmation dans votre flux de travail. Et tant mieux si vous y prenez goût.



Prérequis

- avoir Notepad++ installé sur la machine
- posséder une assez bonne maîtrise de Photoshop pour la photographie.

Il existe depuis de nombreuses années deux types de programmation : la séquentielle et l'évènementielle. Dans la séquentielle, les instructions sont exécutées dans l'ordre de l'écriture du code, donc à la suite les unes des autres.
Dans la programmation évènementielle, l'utilisateur doit faire des choix et définir des valeurs dans une interface graphique - et ce dans l'ordre de son choix - avant d'appuyer sur une commande d'exécution du type bouton «Valider», «OK» ou la touche {Entrée} du clavier.
Le compte à rebours du lancement d'une fusée est un programme séquentiel. Windows et Photoshop sont des interfaces évènementielles. Et bien il en est de même ici. On peut écrire des scripts avec ou sans interface utilisateur. Si on connaît tous les paramètres à l'avance, on peut écrire un script séquentiel, mais si des valeurs doivent être entrées par l'utilisateur en cours d'exécution, on aura besoin de son intervention donc d'une interface graphique. D'un côté on aura un simple **Script**, de l'autre un **Script UI** (User Interface).

Exemple de besoin ne nécessitant qu'un simple script :
- changer la résolution en 72 dpi de toutes les photos se trouvant dans un dossier,
- leur donner une largeur de 800 pixels quelles que soient leurs dimentions initiales,
- et les sauvegarder au format PSD dans un dossier créé pour la circonstance en début de script.

Exemple de besoin nécessitant une interface utilisateur :
- mettre un passe-partout autour de toutes les photos se trouvant dans un dossier en laissant à l'utilisateur le choix :
1) de sa couleur et
2) de la largeur des bandes

Vous trouverez dans la rubrique «**Autres partages**» de ce site quelques scripts qui vous aurez tout intérêt à étudier.

Vos deux premiers programmes

Pour illustrer ce qui vient d'être dit, nous allons créer deux petits programmes en Javascript pour vous faire une idée de ce qu'est un programme avec ou sans interface utilisateur (nous parlerons maintenant de programmes UI).

Si votre version de Notepad++ est vieille, commencez par la mettre à jour. Lancez ce programme et sélectionnez dans le menu **Langage...J...Javascript**. La structure de votre programme sera plus facile à lire et les mots réservés du langage seront mis en couleur.

Tapez le code suivant :

```
1   /*
2       Mon premier programme
3   */
4   //Ce programme affiche le prompt "Hello world !"
5
6   // On fait passer Photoshop au 1er plan.
7   #target photoshop
8   app.bringToFront();
9
10  // Le programme proprement dit
11  alert("Hello world !");
```

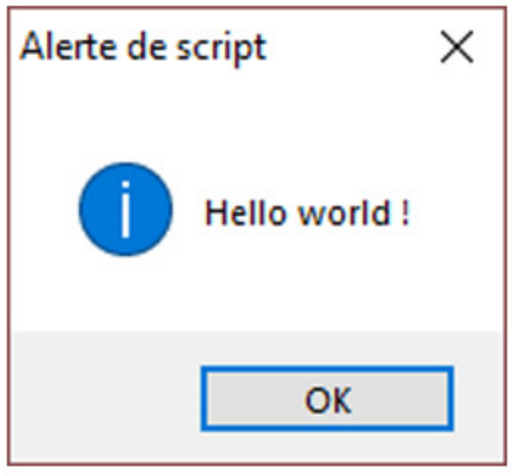
Tout cela mérite quelques explications :
Ligne 1 : **/*** est une **balise de remarques ouvrante**. On peut utiliser autant de lignes que l'on veut ; elles ne sont pas compilées et n'occupent donc pas de mémoire lors de l'exécution du programme. A la ligne 3, on a la balise fermante ***/** qui met fin à ces remarques. Ce système est très pratique lorsqu'on a besoin de bien renseigner son programme.

Ligne 4 : on a une remarque commencant par un double slash **//**. Il s'agit là d'une balise ouvrante signalant une remarque sur une seule ligne. La balise fermante n'est pas obligatoire ; comme les remarques sont en vert sous Notepad++, on ne peut pas confondre avec une ligne d'instruction.
Ligne 5 : on a un saut de ligne pour rendre le code plus facile à lire. Comme les remarques, les lignes vides de ce type ne sont pas compilées.
Ligne 6 : on a une ligne de remarques.
Lignes 7 et 8 : elles font passer Photoshop au premier plan si une autre application est en cours. Vous pouvez copier ces deux lignes d'instructions au début de tous vos scripts, que vous soyez sous Windows ou Mac OS.
Ligne 11 : on a notre première ligne d'instruction.

```
alert("Hello world !");
```

alert() est une instruction qui signifie «*afficher un message destiné à l'utilisateur sous la forme d'un prompt*».

Le message à afficher est entre guillemets, le tout entre parenthèses. A noter un point très important : **les lignes d'instruction se terminent toujours par un point-virgule**. Nous reviendrons là-dessus plus tard. Maintenant, sauvegardez votre fichier dans un dossier où vous mettrez tous vos scripts et donnez-lui l'extension **.JSX**. C'est à cela qu'on fait la différence entre un Javascript pour **Photoshop** et un Javascript pour le HTML qui lui portera l'extension **.JS**.



Allez maintenant sous Photoshop et recherchez votre fichier par «**Fichier...Scripts...Parcourir**». Si vous n'avez pas fait d'erreur vous devriez obtenir ce message. Cliquez sur «OK» pour refermer cette boîte et quitter le programme.

Passons maintenant aux programmes avec interface..

PROGRAMME N° 2.
Tapez le code ci-dessous sous Notepad++ et sauvegardez-le dans votre bibliothèque sous le nom «tuto02.jsx».

```
#target photoshop
app.bringToFront();

var fichier = File.openDialog("Ouvrez le document photo de votre choix :");
if(fichier != null) app.open(fichier);
var dlgBox = new Window('dialog', 'Fermeture', {x:0, y:0, width:180, height:100});
bouton1 = dlgBox.add ("button",  {x:30, y:30, width:50, height:25}, 'OK');
bouton1.onClick = bouton1_click;
bouton2 = dlgBox.add ("button",  {x:100, y:30, width:50, height:25}, 'Annuler');
bouton2.onClick = bouton2_click;
dlgBox.center();
dlgBox.show();

function bouton1_click()
{
    app.activeDocument.close(SaveOptions.DONOTSAVECHANGES);
    dlgBox.hide();
}

function bouton2_click()
{
    alert("On laisse ouvert");
}
```

Il n'y a pas de remarques dans le programme que vous avez tapé parce que nous allons donner ici toutes les explications qui s'imposent. Lorsque vous préparerez vos propres programmes, notez les deux règles suivantes :

- un programme mal renseigné (avec trop peu de remarques) ne vaut pas grand-chose
- un programme trop renseigné ne vaut pas mieux, car les remarques en excès tuent les remarques utiles.

Lignes 1 et 2 : on fait passer Photoshop au 1er plan (déjà vu)

Ligne 4 : **Var** est un mot réservé qui permet de déclarer une variable. Contrairement aux constantes dont la valeur est fixe et qu'on doit écrire en majuscules (exemple PI=3.1416), une variable peut prendre n'importe quelle valeur pendant le déroulement du programme. Les variables doivent être obligatoirement déclarées avant d'être utilisées. Il est fortement conseillé de les initialiser en leur donnant une valeur par défaut au moment de leur déclaration (valeur nulle par exemple). Cela s'appelle une «instanciation». Exemple : **Var** nbMois = 12;

File.openDialog**()** est une fonction système qui permet de rechercher un fichier à l'aide du navigateur

Ligne 5 : **if(.....)** pose une condition.
La condition peut être :
- une égalité : if(age == 18)
- une inégalité : if (age != 18) signifie si âge est différent de 18
- une infériorité ou supériorité : if (age>=18)

if(fichier **!=** null**)**, ici la condition signfie "Si le nom du fichier n'est pas vide", autrement dit "si on a sélectionné un fichier". L'instruction à exécuter suit la condition. Ici app.**open(**fichier**)** commande le chargement de la photo sélectionnée.
Enfin notez la présence du point-virgule à la fin de la ligne d'instruction. Il ne faudra jamais l'oublier .

Ligne 6 : Voici enfin notre première interface. On crée une boîte de dialogue de 180 pixels de large sur 100 pixels de haut. Le mot réservé **New** permet d'instancier un objet à partir d'un modèle (on dit «classe»). Ici on crée une boîte (ou fenêtre) à partir de la classe «Window» du système et une fenêtre qui sera de dialogue du fait de la sous-classe **«dialog»** qui est un mot réservé comme **«New»** et **«Window»**. Cette boîte aura pour nom «*dlgBox*» et pour titre «*Fermeture du fichier*» ; elle ne sera pas visible tant qu'on ne l'aura pas décidé. Le choix du nom et du titre est libre.

Lignes 7 : on ajoute à la boîte de dialogue un bouton sur lequel l'utilisateur devra cliquer s'il veut refermer la photo qu'il a chargée. La syntaxe qui permet de placer le bouton dans la boîte de dialogue est la suivante (abscisse du bouton par rapport au bord gauche de la fenêtre, ordonnée par rapport au sommet, largeur et hauteur du bouton, le tout en pixels). Le zéro du repère se trouve donc en haut à gauche

Lignes 8 : on ajoute une commande qui exécute la fonction «bouton1_click()» si l'utilisateur veut refermer la photo

Lignes 9 et 10 : idem pour le deuxième bouton si l'utilisateur ne veut pas la refermer

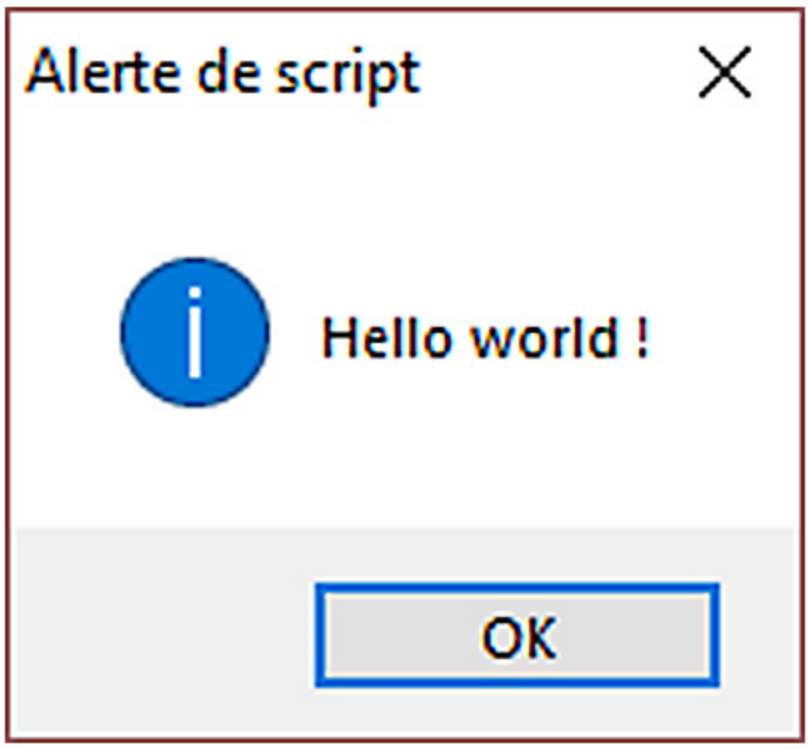
Lignes 11 et 12 : on centre la boîte de dialogue sur l'écran et on la rend visible

Lignes 14 à 18 : instructions pour refermer la photo ET la boîte de dialogue

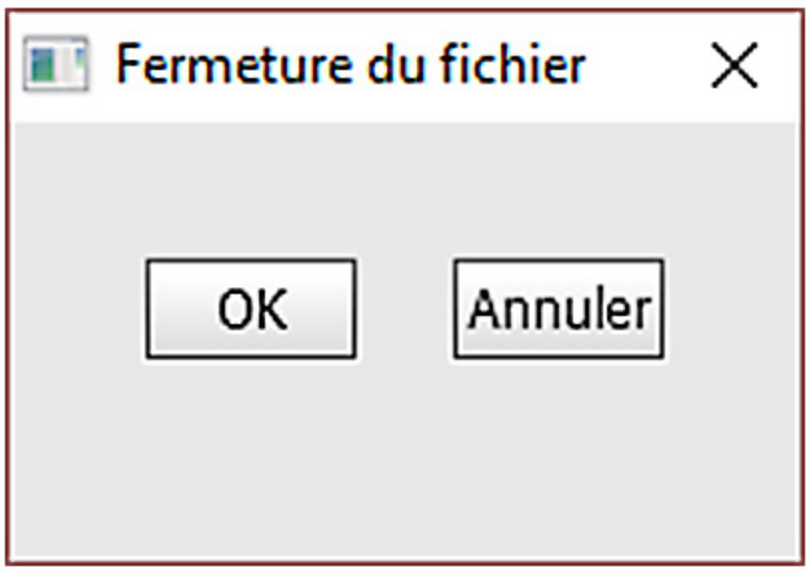
Lignes 22 : on affiche le message «On laisse ouvert», mais il n'y aura aucune action. La boîte de dialogue restera ouverte.

Si vous n'avez pas fait d'erreur, vous devez obtenir les messages suivants :

Programme 1 :



Programme 2 :



Vous pourrez télécharger ces deux scripts en :
- http://www.planete-bleue.net/javascript/tuto01.jsx et
- http://www.planete-bleue.net/javascript/tuto02.jsx

Fin.