

JavaScript

Plan du cours

- 1- Bases et introduction à JavaScript
- 2- Insertion du code JavaScript
- 3- Utilisation des constantes , variables et opérateurs
- 4- Contrôler les scripts avec les structures de contrôles
- 5- Fonctions et évènements
- 6- Initiation à la Programmation Orienté Objet et JavaScript

7- Les formulaires

8- Les principaux objets JavaScript en détail

JavaScript

Qu'est-ce que le Javascript?

Le **Javascript** est un langage de script incorporé dans un document HTML.

Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web.

Javascript est donc une extension du code Html des pages Web. Les scripts, *qui s'ajoutent aux balises Html*, peuvent en quelque sorte être comparés aux macros d'un traitement de texte.

Ces scripts vont être gérés et exécutés par le browser lui-même sans devoir faire appel aux ressources du serveur. Ces instructions seront donc traitées en direct et surtout sans retard par le navigateur.

Ainsi le langage Javascript est fortement dépendant du navigateur appelant la page web dans laquelle le script est incorporé, mais en contrepartie il ne nécessite pas de compilateur, contrairement au langage [Java](#), avec lequel il a longtemps été confondu.

Javascript a été initialement développé par Netscape et s'appelait alors LiveScript. Adopté à la fin de l'année 1995, par la firme Sun (qui a aussi développé Java), il prit alors son nom de Javascript.

Javascript n'est donc pas propre aux navigateurs de Netscape (bien que cette firme en soit un fervent défenseur). Microsoft l'a d'ailleurs aussi adopté à partir de son Internet Explorer 3. On le retrouve, de façon améliorée, dans Explorer 4.

Les versions de Javascript se sont succédées avec les différentes versions de Netscape :

Javascript pour Netscape 2, Javascript 1.1 pour Netscape 3 et Javascript 1.2 pour Netscape 4. Ce qui n'est pas sans poser certains problèmes de compatibilité, selon le browser utilisé, des pages comportant du code Javascript. Mais consolons nous en constatant qu'avec MSIE 3.0 ou 4.0 et la famille Netscape, une très large majorité d'internautes pourra lire les pages comprenant du Javascript.

L'avenir de Javascript est entre les mains des deux grands navigateurs du Web et en partie lié à la guerre que se livrent Microsoft et Netscape. On s'accorde à prédire un avenir prometteur à ce langage surtout de par son indépendance vis à vis des ressources du serveur.

Il importe de savoir que Javascript est totalement différent de Java. Bien que les deux soient utilisés pour créer des pages Web évoluées, bien que les deux reprennent le terme Java , nous avons là deux outils informatiques bien différents.

Le Javascript est *case sensitive* (en français *sensible à la casse*)

comme en langage C, chaque instruction se termine par un point-virgule (;).

Javascript : Code intégré dans la page Html

Java: Module (applet) distinct de la page Html

Javascript : Code interprété par le browser au moment de l'exécution

Java: Code source compilé avant son exécution

Javascript : Codes de programmation simples

Java : Langage de programmation beaucoup plus complexe

Limites et avantage de java script

JavaScript est assez simple à comprendre et il est possible de réaliser rapidement de petits scripts impressionnants. De plus, c'est un langage très riche et qui offre beaucoup de possibilités.

Le principal avantage de Javascript réside dans la sécurité. En effet, les concepteurs ont interdit toutes les opérations qui pouvaient porter atteinte à la sécurité du visiteur.

Le vol d'information, la destruction de fichiers sont impossibles.

Ce qui peut paraître une faiblesse : il est impossible de créer un fichier, d'accéder à une base de données en JavaScript

un gros atout : PRATIQUEMENT TOUS LES VISITEURS
ACCEPTENT LE JAVASCRIPT.

Le véritable inconvénient de JavaScript est sa compatibilité très limitée entre navigateurs Internet Explorer et Netscape ont parfois des visions très différentes du JavaScript. Cela oblige souvent à coder 2 scripts pour la même action.

Outil de conception:

Les outils permettant d'insérer du code JavaScript sont nombreux cela va du simple logiciel éditeur de texte, comme Word Pad de Windows à l'outil spécifique Aptana Studio, on passant par les éditeurs de code HTML tels Dreamweaver ou Frontpage. Avec les quels il est possible d'insérer des blocs Javascreept.

Paramétrage:

Rappelle:

Une page HTML se divise en deux grandes parties:

- La partie head dans laquelle se situent les informations correspondant a la description du contenu;

- La partie body ou figure le code permettant la construction des objets dans la page (Champs de formulaire , zone de texte, image,.....).

Un script est une portion de code qui vient s'insérer dans une page HTML.

Le code du script n'est toutefois pas visible dans la fenêtre du navigateur car il est compris entre des balises (ou tags) spécifiques qui signalent au navigateur qu'il s'agit d'un script écrit en langage JavaScript.

Les balises annonçant un code Javascript sont les suivantes :

```
<SCRIPT language="Javascript">
```

Placez ici le code de votre script

```
</SCRIPT>
```

Le code HTML sert généralement de conteneur au bloc d'instruction JavaScript, suite au chargement du code HTML. Le navigateur exécute les blocs d'instructions JavaScript.

Type d'exécution du code JavaScript:

- JavaScript Interne. (Entre deux balises `<script>` /`</script>`)
- JavaScript externe.(Fichier javascript en mode `.js`)

1er Programme utilisant JavaScript

script simple qui doit vous saluer en écrivant votre nom à l'écran.
N'oublions pas que JavaScript est toujours inséré dans une Page HTML .

Créons une Page HTML (la plus simple possible , ou presque..!)

```
<HTML>
  <HEAD>
    <TITLE> Mon 1er Programme en JavaScript</TITLE>

    <SCRIPT LANGUAGE="JavaScript">
document.write("Bonjour ..votre nom.! ");

  </SCRIPT>

</HEAD>

</HTML>
```

- La balise **<SCRIPT>** est obligatoire pour prévenir le navigateur que ce qui suit est du langage JavaScript.
- Le mot **language** est le mot anglais..ce qui explique le **u** .
- La balise **SCRIPT** se trouve insérée dans la balise **HEAD**.. C'est habituel mais pas obligatoire..Il est possible de la placer n'importe où ..même dans la balise **BODY**.

Mettre le code JavaScript dans un fichier

Pour l'exemple précédent, il suffit de créer un fichier nommé `bonjour.js` contenant la ligne :

```
document.write("<B>Bonjour tout le monde !</B>");
```

Dans le fichier html, l'appel à JavaScript se fera simplement avec la ligne :

```
<SCRIPT LANGUAGE="JavaScript" SRC="bonjour.js"></SCRIPT>
```

Cette possibilité se révèle utile lorsqu'on a des portions de code qui se répètent souvent.

Différents types d'expressions

Nous avons utilisé l'instruction `document.write` pour afficher une expression écrite entre guillemets. Celles-ci indiquent que ce qu'elles entourent est du code HTML que le navigateur va interpréter. Il est aussi possible d'utiliser des expressions sans guillemets. Dans ce cas JavaScript va essayer de les interpréter directement avant de les transmettre au navigateur. Les expressions mathématiques permettent de comprendre ce processus.

L'instruction **`document.write("2+5*3");`** provoque l'affichage du texte `2+5*3`. Par contre, l'instruction **`document.write(2+5*3);`** provoque l'affichage du texte `17`. Dans le 1er cas JavaScript a interprété `"2+5*3"` comme un texte quelconque envoyé sans changement au navigateur. Dans le 2ème cas JavaScript a interprété `2+5*3` comme un calcul à effectuer, c'est le résultat de ce calcul qui a été envoyé au navigateur.

Pour programmer on est souvent amené à utiliser un autre type d'expression appelé expression booléenne. Celle-ci ne peut prendre que deux valeurs, **true** (vrai) ou **false** (faux)

Insérez le script suivant et observez:

```
document.write("5*4=12 ?? "); document.write(5*4==12);
```

```
document.write("5*4=20?? "); document.write(5*4==20);
```

l'instruction **document.write** permet d'afficher différents types d'expressions :

des **chaînes de caractères** : elles sont écrites entre guillemets et transmises directement au navigateur

des **expressions mathématiques** : ce sont des calculs numériques, c'est le résultat qui est transmis au navigateur

des **expressions booléennes** : ce sont des expressions dont la valeur est true ou false

L'instruction `document.write()` peut contenir plusieurs arguments qui seront affichés les uns derrière les autres.

Lisibilité du code

Il est très important d'avoir un code lisible : ça permet de s'y retrouver, et donc d'éviter bon nombre d'erreurs.

De plus, si vous commettez une erreur il est beaucoup plus facile de la localiser dans un code bien organisé que dans un code illisible .C'est donc un point à ne pas négliger...

Les commentaires:

Il existe deux manières d'écrire des commentaires en JS : en fin de ligne, après `//`
n'importe où (peut s'étendre sur plusieurs lignes), entre `/*` et `*/`.

Remarque:

Leur rôle est de rendre la page plus facilement lisible. Mais noyer le code entre des commentaires n'est pas forcément mieux que de ne pas le commenter. Il faut donc qu'ils soient clairs, et utilisés là où ils sont nécessaires.

Utiliser des variables

On utilise des variables pour inscrire des informations dans la mémoire de l'ordinateur et pouvoir facilement les retrouver et les modifier (les faire "varier"). Chaque variable est donc caractérisée par un nom qui permet de la désigner et par un contenu qui est ce qui est inscrit dans la mémoire.

Les types de variables

On peut classer les variables selon le type de leur contenu :

chaîne de caractères

- valeur numérique
- valeur booléenne
- objet

Pour utiliser une variable, il est conseillé de la déclarer et de lui donner un contenu initial. Cela se fait en utilisant le mot **var** et le signe **=**.

Par exemple :

`var v1=« karim »;` La variable nommée **v1** contient la chaîne de caractères « **karim** ».

`var v2=13;` La variable nommée **v2** contient la valeur numérique **13**.

`var v3=true;` La variable nommée **v3** contient la valeur booléenne **true**.

- Nous verrons plus tard comment utiliser des variables contenant des objets.

Lorsqu'une variable est interprétée dans une instruction JavaScript, elle est automatiquement remplacée par son contenu. Ainsi, l'instruction `document.write(v1)` provoquera l'affichage du mot karim. Attention, l'instruction `document.write("v1")` provoquera l'affichage du mot v1 et non Henri entre guillemets, dans ce cas JavaScript considère qu'il faut afficher une chaîne de caractères et n'essaie pas d'interpréter son contenu.

Opérations sur les variables

Les variables numériques peuvent être utilisées dans des expressions mathématiques faisant intervenir les quatre opérations élémentaires.

Les variables contenant des chaînes de caractères peuvent être *ajoutées*, cette opération se nomme la concaténation, elle consiste à construire une nouvelle chaîne de caractères en en assemblant deux. Ainsi, si $v1 = \text{"bonjour "}$ et $v2 = \text{"karim "}$, alors $v1+v2$ est la chaîne de caractères contenant "bonjour karim".

Les variables booléennes peuvent être combinées à l'aide des opérations logiques **non**, **et**, **ou** respectivement notées **!**, **&&**, **||**.

Inserez le script suivant et observez:

```
<SCRIPT LANGUAGE="JavaScript">  
var v1="12";  
var v2=5;  
document.write("La variable v1 contient : ",v1,"<BR>");  
document.write("La variable v2 contient : ",v2,"<BR>");  
document.write("Avec v1+v2 on obtient : ",v1+v2,"<BR>");  
document.write("Avec v2+v1 on obtient : ",v2+v1,"<BR>");  
</SCRIPT>
```

La fonction eval

Il arrive souvent que l'on doive interpréter une chaîne de caractères (par exemple une saisie de l'utilisateur) comme étant une valeur numérique. JavaScript permet de réaliser cette opération avec la fonction **eval**.

Observons l'effet du [script](#) suivant.

```
<SCRIPT LANGUAGE="JavaScript">
var v1="12";
var v2=5;
document.write("La variable v1 contient : ",v1,"<BR>");
document.write("La variable v2 contient : ",v2,"<BR>");
document.write("Avec v1+v2 on obtient : ",v1+v2,"<BR>");
document.write("Avec eval(v1)+v2 on obtient : ",eval(v1)+v2,"<BR>");
</SCRIPT>
```

La fonction **eval** de JavaScript est très puissante, elle permet aussi d'évaluer des chaînes de caractères contenant des opérations. Par exemple `eval("12-8")` fournira 4 comme résultat.

La fonction **prompt**

Il arrive souvent que le contenu d'une variable doive être choisi par l'utilisateur du programme, et non par le programmeur. La fonction **prompt()** fournit une façon de réaliser cette opération avec JavaScript. Elle attend en paramètre deux chaînes de caractères: la première servira de message d'invite et la seconde de réponse proposée par défaut.

```
<html>
<head>
<title>Mon premier programme en js</TITLE>

<SCRIPT LANGUAGE="JavaScript">
var nom;
nom=prompt("Quel est ton nom ?", "");
document.write("<BIG><B>B</B></BIG>onjour "+nom+" !<BR>");
</SCRIPT>

</HEAD>
</html>
```

Exercice 1:

Calculatrice

Utiliser les fonctions `eval` et `prompt` pour écrire une petite calculatrice.

```
l><head>
  <title>Ma calculette JavaScript</title>
</head>
<body>
<h1>Ma calculette JavaScript</h1>
<hr>
<script language="JavaScript">
var expr;
expr=prompt("Entrer un calcul :", "");
document.write(expr, " = ", eval(expr)+"<BR>");
</script>null = null<br>
<p><br></p>
<a
href="file:///C:/Documents%20and%20Settings/hadbi/Mes%20docume
nts/TD%20Javascript/td-depart/eval.html">Nouveau calcul...</a>

</body></html>
```

Fonction Alert

Affichage du resultat d'une addition de deux variable dans une boite de dialogue:

```
<html>
<head>
<title>Mon premier programme en js</TITLE>

<SCRIPT LANGUAGE="JavaScript">
var somme, nombre1= 10; nombre2= 5;
somme=nombre1+nombre2;
alert("le resultat de l'addition est : " +somme);
</script>

</HEAD>
</html>
```

Exercice 2:

Affichez le résultat d'un calcul de cout du trajet d'un véhicule.

L'utilisateur doit renseigner plusieurs boites de dialogue

La marque

Le cout du litre de carburant

Le kilométrage du véhicule

En utilisant la formule suivante:

$\text{cout_total} = \text{nb_kilometres} / 100 * \text{consommation} * \text{prix_litre};$

En utilisant les fonctions

Prompt et alert

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.