

Le langage JAVASCRIPT

I Introduction

Vous vous dites encore un langage alors qu'il y en a déjà un nombre non négligeable. En effet un de plus, mais à chaque langage son utilisation et sa particularité. Javascript propose un certain nombre d'avantages que ce tutorial va s'efforcer de démontrer.

I.1 Historique

Le javascript est un langage normalisé par l'organisme du nom d'ECMA (European Computer Manufacturers Associations) sous le numéro 262 (3^{ème} édition 1999).

Vous retrouvez aussi cette norme sous le standard ISO (International Standard Organization) sous le nom ISO-16262.

La spécification est récupérable à l'adresse suivante :

<http://www.ecma.ch/ecma1/STAND/ecma-262.htm>

Ce langage de script est apparu en 1996 grâce à Brendan Fich. Connu sous le nom de javascript, ce langage va être inclus dans la version 2.0 du navigateur Netscape puis dans la version 3.0 du navigateur Internet Explorer.

Il est normalisé pour la première fois en Juin 1997 par l'ECMA puis en Avril 1998 par l'ISO.

La version actuelle de javascript est 1.3.

Attention

Beaucoup de personnes confondent le langage JAVA et le langage JAVASCRIPT. A part les 4 premières lettres ces deux langages n'ont rien à voir. Le premier à été créé par SUN et est un vrai langage Objets pour le développement.

I.2 Particularités du langage

Tout d'abord c'est un langage de scripts. C'est-à-dire un langage de programmation qui permet d'automatiser des tâches, de manipuler et améliorer le fonctionnement des systèmes.

C'est un langage orienté objet. L'ajout du mot orienté est important. En effet javascript utilise des outils des langages objets, mais il n'en est pas complètement un. Il ne propose pas exactement la même philosophie d'utilisation et de développement des objets notamment au niveau de l'héritage.

Définition

Un langage objet est un langage de programmation qui propose un ensemble de méthodes (fonctions) et de propriétés (caractéristiques) généralistes qui peuvent s'appliquer aux différents objets définis dans le langage. Il existe des classes d'objets, qui hiérarchisent les objets qui la constituent. Chaque objet hérite des propriétés et méthodes de l'objet parent.

Cette notion de classe en javascript n'est pas aussi pointue et bien définie. C'est pour cela que le javascript est un langage **orienté** objet ou **basé** objet selon le terme employé mais pas objet tout cours.

Les objets en javascript sont des entités qui décrivent les éléments contenus dans une page HTML (frame, formulaires, image ...) ainsi que les éléments qui structures cette page (navigateur ...).

On retrouve ces éléments caractéristiques d'un langage objets dans d'autres langages objets comme ADA, C++, PYTHON ou JAVA.

Le javascript est un langage pour le web qui permet de dynamiser les pages HTML en interceptant les actions réalisées par l'internaute (clic droit ...).

A la différence de langages comme PHP, le traitement des parties du document chargé par l'internaute réalisées en javascript, ne sont pas interprétées par le serveur web (celui qui distribue les pages sur l'internet), mais par le client (navigateur). Le navigateur (netscape, mozilla, Internet Explorer ...) possède donc un interpréteur de javascript. Le traitement du langage HTML est identique, l'interprétation est réalisée côté client et non côté serveur.

Attention

Un effet pervers de cette interprétation côté client est d'avoir des réactions différentes selon le navigateur utilisé et notamment avec Internet Explorer de Microsoft qui propose une version de javascript différente de ce que préconise la norme. Microsoft a créé son « propre »

javascript : Jscript qui se démarque de la norme ECMA-262. Ce phénomène se retrouve aussi avec le HTML.

Le javascript est donc un complément du HTML. Il permet de produire du HTML dynamique (DHTML). C'est-à-dire un résultat différent selon les interactions que fait l'internaute ou son environnement. Le javascript va créer du HTML ou se mettre au service de la page HTML pour proposer ce dynamisme. Il permet aussi d'accentuer, d'améliorer et d'enrichir la partie graphique de votre page HTML (image rollover, menu déroulant graphique, changement de couleurs ...). Cette dernière partie est très importante et a permis l'essor du javascript qui a permis de combler un vide du langage HTML.

Remarque

Le fait que ce soit le client qui interprète le code javascript entraîne que votre code est visible par le client. Donc n'incluez rien de « secret » dans vos codes javascript.

I.3 Les bases du langage javascript

Comme pour tous les langages, il existe des règles de construction et d'utilisation.

a. Inclusion du code javascript

Vous avez deux possibilités pour insérer du code javascript dans vos sites Web et dans les pages HTML. Vous pouvez aussi utiliser les deux méthodes simultanément.

- Dans un fichier externe

Ce fichier doit posséder l'extension .js et il doit être accessible par le web. Ce fichier va contenir les fonctions, définitions et tous les éléments javascript nécessaires au fichier HTML auquel ce fichier est rattaché. Il faut alors inclure le fichier .js dans l'entête du fichier HTML de la façon suivante :

```
<HTML>
<HEAD>
<script language="javascript" src="chemin/fichier.js">
</script>
</HEAD>
```

...

Vous devez donc indiquer dans l'attribut **src** l'endroit où se situe le fichier ayant l'extension js ainsi que le nom de ce fichier.

➤ Dans le fichier HTML

Vous pouvez aussi inclure le ou les codes javascript directement dans le fichier HTML. Il faut alors indiquer au moment où vous incluez du code javascript que vous faites cette action afin que l'interpréteur du navigateur client « traduise » cette partie de code et effectue les actions demandées.

A chaque endroit (ou presque) où vous incluez du javascript dans votre document HTML, vous devez utiliser la syntaxe suivante :

```
... ← Codes HTML
<script language="javascript"> ← Début du code javascript
Codes javascript ← Codes javascript
</script> ← Fin du code javascript
... ← Codes HTML
```

Ce schéma peut être répété autant de fois que nécessaire.

Remarque

Parfois en plus du mot clé javascript, la version de ce dernier est indiquée (javascript 1.2).

A contrario, il est aussi possible d'omettre la partie qui suit le mot clé script et se contenter de la syntaxe <script>...</script>, l'interpréteur du navigateur client le comprend.

Dans le deuxième cas, les déclarations de méthodes (fonctions) se fait dans l'entête du document entre les balises <HEAD> et </HEAD>. Mais malgré tout rien n'empêche de le faire dans le document même entre les balises <BODY> et </BODY>.

Par contre, dans les deux méthodes d'inclusion du code, les éléments visibles dans la page HTML se situent dans la partie BODY où sont chargés en même temps que la page en l'indiquant à l'intérieur de la balise BODY.

b. Les commentaires

Comme dans tous les langages de programmations mais aussi comme dans tous les documents, il est impératif de commenter les lignes de ses

programmes pour une meilleure lisibilité et pour une meilleure compréhension pour vous-même mais aussi pour celles et ceux qui devront reprendre ces pages de codes si besoin est. Pour inclure un commentaire vous avez deux possibilités qui sont tirées du C et du C++.

1. Les commentaires

➤ `//`

Ce type de commentaires permet d'insérer un commentaire sur **une seule** ligne.

➤ `/* ... */`

Ce type de commentaires permet d'insérer un commentaire sur **plusieurs** lignes.

Attention

N'imbriguez pas les commentaires.

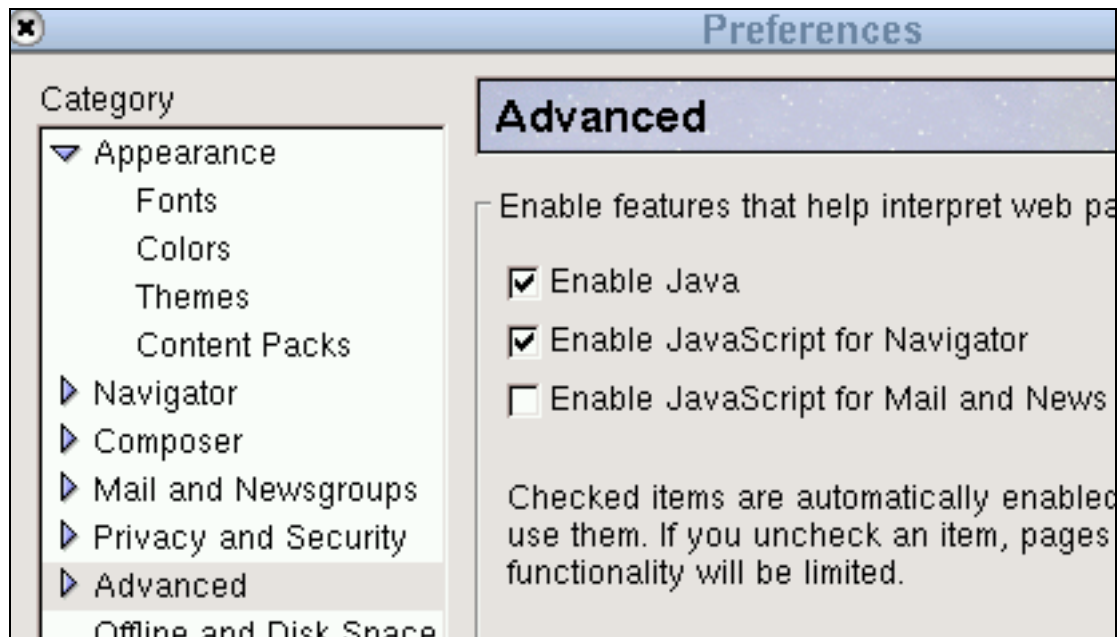
La syntaxe suivante n'est pas valide :

```
/* ... /* ... */ ... */
```

2. Particularités

➤ `<!-- ... // -->`

Il existe un cas particulier d'utilisation des commentaires. Tous les navigateurs ne supportent pas le javascript. De plus, l'utilisateur peut lui-même de son propre chef désactiver cette option (Mozilla -> Edit -> Preferences -> Advanced -> décochez la case « enable javascript for navigator »).



Dans ce cas, vos codes en javascript risquent de polluer l'affichage de la page HTML.

Vous pouvez alors encadrer votre code javascript par le commentaire utilisé en HTML :

```

... ← Codes HTML
<script language="javascript"> ← Début du code javascript
<!--
Codes javascript ← Codes javascript
// Fin du code javascript -->
</script> ← Fin du code javascript
... ← Codes HTML

```

Ainsi, ce qui se trouve à l'intérieur de ce commentaire HTML, ne sera pas montré dans le cas où l'interpréteur HTML est seul activé. Par contre dans le cas où l'interpréteur javascript est activé, il prendra en compte les codes javascript, les commentaires HTML sont alors ignorés.

➤ `<noscript> ... </noscript>`

La balise noscript est une balise HTML et non javascript. Elle permet, dans le cas où le navigateur ne supporte pas le javascript où si il est désactiver d'afficher un message dans la page HTML chargée par l'utilisateur. Ce message doit être situé entre le début et la fin de la balise noscript.

c. Aide à la programmation Javascript : le débogger

Il est toujours intéressant, surtout au début de l'apprentissage d'un nouveau langage, de connaître les différents outils et les astuces qui peuvent aider à la création de vos codes.

Un des éléments importants est souvent de trouver d'où provient une erreur.

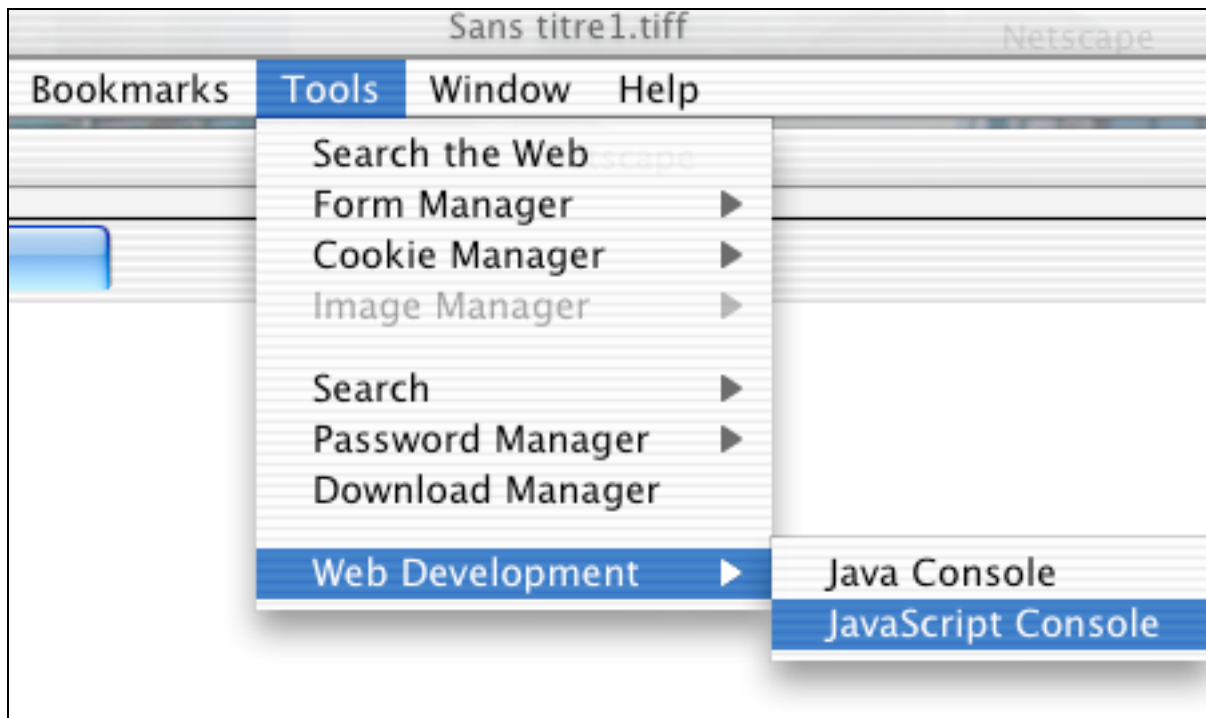
Les navigateurs web (Mozilla, Netscape) mettent à votre disposition un débogueur intégré. Ce débogueur va vous informer des éventuelles erreurs qui ont eu lieu lors du chargement et l'interprétation d'une page contenant du javascript.

Pour obtenir ce débogueur appelé aussi Console Javascript :

- tapez comme URL **javascript:** après une erreur par exemple. Une fenêtre s'ouvre en indiquant le type et la raison de ou des erreurs.



- Vous pouvez aussi obtenir cette console Javascript au niveau du menu des navigateurs.



➤ Autres possibilités

Vous pouvez aussi installer un débogueur externe.

- Chez Microsoft vous pouvez le télécharger à l'adresse suivante : <http://www.microsoft.com/scripting/debugger>
- Mozilla et Netscape proposent aussi un certain nombre d'outils de débogage aux adresses suivantes :
<http://www.mozilla.org/projects/venkman>
<http://developer.netscape.com/software/jsdebug.html>

d. Syntaxes

Voici quelques règles simples à respecter pour éviter certaines surprises.

- A la fin d'une ligne, il n'est pas nécessaire de mettre un point virgule (;). Si vous l'oubliez, l'interpréteur l'ajoute si besoin est.
- Les lignes vides ne comptent pas en Javascript.
- Evitez, comme toujours, les caractères accentués dans la construction des éléments de vos programmes notamment au niveau des variables. Par contre au niveau de tout ce qui est message d'affichage pas de restriction.
- La casse des éléments (variables, constantes, objets ...) est importante. Numero et numero ne représenta pas le même élément.

- La virgule dans un nombre est symbolisée par le caractère . (point).