

1. JAVASCRIPT SYNTAXE

JavaScript est sensible à la casse. Ainsi pour afficher une boîte de dialogue d'alerte, il faudra écrire **alert()** et non **Alert()**.

Pour l'écriture des instructions JavaScript, on utilisera l'alphabet ASCII classique (à 128 caractères) comme en Html. Les caractères accentués comme é ou à ne peuvent être employés que dans les chaînes de caractères.

Pour déclarer une chaîne de caractères, les guillemets " et l'apostrophe ' peuvent être utilisés à condition de ne pas les mélanger. Si vous souhaitez utiliser des guillemets dans vos chaînes de caractères, tapez \" ou \' pour les différencier vis à vis du compilateur.

JavaScript ignore les espaces, les tabulations et les sauts de lignes.

Les points-virgules terminent les instructions. Cependant, JavaScript accepte les instructions isolées sans terminaison en point-virgule.

Pensez toujours à finir vos instructions par un point-virgule afin d'éviter les erreurs !

1.1. LA BALISE <SCRIPT>

Dans la logique du langage Html, il faut signaler au navigateur par une balise, que ce qui suit est du code JavaScript (et non du VBScript par exemple). C'est la balise **<script>**.

Depuis la version HTML 4.01 et le XHTML¹, le type MIME² est indiqué dans la balise :

`<script type="text/JavaScript">`

Auparavant, les spécifications utilisaient l'attribut "language" :

¹ XHTML: Extensible HyperText Markup Language

² MIME : Multipurpose Internet Mail Extension

```
<script language="JavaScript">
```

Maintenant en HTML5, il suffit simplement de placer une balise script

```
<script>
```

De même, il faudra informer le navigateur de la fin du script.

```
</script>
```

1.2. LES COMMENTAIRES

Il vous sera utile d'inclure des commentaires personnels dans vos codes JavaScript. C'est même vivement recommandé comme pour tous les langages de programmation.

JavaScript utilise les conventions utilisées en C et C++ soit

```
// Commentaire sur une seule ligne
```

Tout ce qui est écrit entre le // et la fin de la ligne sera ignoré.

Il sera aussi possible d'inclure des commentaires sur plusieurs lignes avec le code

```
/* commentaire sur plusieurs      lignes */
```

Ne confondez pas les commentaires JavaScript et les commentaires Html (pour rappel <!-- ...-->) !

1.3. MASQUER LE SCRIPT POUR LES ANCIENS NAVIGATEURS

Les navigateurs qui ne comprennent pas le JavaScript (il y en a encore ?) ignorent la balise **<script>** et vont essayer d'afficher le code du script sans pouvoir l'exécuter. Pour éviter l'affichage peu esthétique de ses inscriptions cabalistiques, on utilisera les balises de commentaire du langage Html <!-- ... -->.

Votre premier JavaScript ressemblera à ceci :

```
<script type="text/JavaScript" language="JavaScript">  
  <!-- Masquer le script pour les anciens navigateurs  
        code JavaScript  
  // Cesser de masquer le script -->
```

```
</script>
```

Cette technique paraît maintenant désuète, les applications HTML5 utilisant de façon intensive le code Javascript.

1.4. LES SPECIFICATIONS XHTML STRICT

Le code XHTML Strict interprète certains caractères comme des marqueurs même s'ils sont dans une section **<script> ... </script>**. (Notamment les caractères <, >, &).0

Pour y remédier, il faut insérer le code JavaScript dans un bloc CDATA qui n'interprète pas les caractères comme des marqueurs.

```
< script type="text/JavaScript" language="JavaScript">
  // <![CDATA[
    code JavaScript
  // ]]>
</script>
```

CDATA est issu des spécifications XML.

Pour éviter cette mixture dans le but de masquer le code ou de le rendre valide pour les validateurs, mieux vaut toujours appeler le code JavaScript dans un fichier externe.

1.5. OU INCLURE LE CODE EN JAVASCRIPT ?

Le principe est simple. Il suffit de respecter les deux principes suivants :

- n'importe où ; mais là où il le faut.

Dans une page Html, le code JavaScript peut-être intégré de 2 façons :

- entre les balises **<script> ... </script>** ;
- directement dans les balises via les attributs de gestion des événements.

Le navigateur traite votre page Html de haut en bas (y compris vos ajouts en JavaScript). Par conséquent, toute instruction ne pourra être exécutée que si le navigateur possède à ce moment précis tous les éléments

nécessaires à son exécution. Ceux-ci doivent donc être déclarés avant ou au plus tard lors de l'instruction.

Pour s'assurer que le programme script est chargé dans la page et prêt à fonctionner à toute intervention de votre visiteur (il y a des impatients), il faut déclarer systématiquement (lorsque cela est possible) un maximum d'éléments dans les balises d'en-tête **<head> ... </head>**. Ce sera le cas par exemple pour les fonctions.

Les objets issus du corps **<body> ... </body>** sont affichés avant d'être complètement chargés. En revanche, la section d'en-tête **<head> ... </head>** est toujours chargée en premier.

- ➔ Si on appelle une fonction JavaScript qui n'est pas encore chargée, cela génère une erreur !

1.6. L'EXTENSION .JS POUR LES SCRIPTS EXTERNES

Il est possible d'utiliser des fichiers externes pour les programmes JavaScript. On peut ainsi stocker les scripts dans des fichiers distincts (avec l'extension .js) et les appeler à partir d'un fichier Html. Le concepteur peut de cette manière se constituer une bibliothèque de script. La balise devient

<code><script src="http://site.com/javascript.js"></script></code>
--

Attention : Quand on appelle un fichier externe via l'attribut *src*, tout code se trouvant entre les balises `<script> ... </script>` est ignoré.

Exercice

Le but est d'externaliser le code JavaScript dans un fichier .js puis de tester l'intégration de code Html dans les fonctions JavaScript.

1. Copier le code suivant dans votre éditeur de texte préféré.

```
<html>
  <head>
    <title>Mon 1er Script</title>

    <script>

      alert("C'est mon 1er script");
      console.log("coucou");
      document.write("coucou");

    </script>
  </head>

  <body>
    <!-- Ajouter du code Javascript ici pour le titre -->

    Rien a ajouter !
  </body>
</html>
```

Enregistrer le fichier puis l'ouvrir avec votre navigateur

2. Créer un fichier JavaScript nommé «Exercice1.js »
3. Déplacer le code JavaScript du fichier Html dans le fichier externe JavaScript et importer ce fichier dans le fichier Html.
4. Répéter le titre de la page dans le <body> avec l'instruction document.write(), consultez la documentation sur le site W3School http://www.w3schools.com/jsref/met_doc_write.asp

- o Ajouter du code Html dans la chaîne de caractères passée en paramètre.

5. Modifier la fonction alert() pour y passer en paramètre également du code Html.

- o Que remarquez-vous ?