

Découvrir le JavaScript

Introduction:

. JavaScript est un **langage de programmation de scripts** principalement utilisé dans les pages web interactives **côté client**.

Créer des scripts – Les scripts JavaScript:

- . Sont des scripts **fichiers «texte»** (extension conseillée .js) à créer avec un **éditeur de texte**
- . Sont **intégrés au sein des pages web**.
- . Sont **exécutés côté client par le navigateur web**.

JavaScript sert:

- . à **contrôler les données saisies dans des formulaires HTML**
- . à **interagir avec le document HTML via l'interface DOM** (**Document Object Model**) fournie par le navigateur (On parle alors parfois de HTML dynamique ou DHTML)
- . à modifier le contenu des pages web par programmation avec la méthode Ajax (Asynchronous JavaScript And XML)

Exemple 1:

Le code JavaScript est généralement inséré entre les balises <HEAD> et </HEAD>.

```
<script TYPE= «text/javascript»>  
//Du code JavaScript  
alert(«Hello World !»); //affiche une boîte de dialogue modale  
</script>  
</head>
```

Exemple 2: Sur page web. Écriture en JavaScript

```
document.write(«<p>Votre navigateur accepte le JavaScript.</p>»);
```

Fichier script à intégrer dans sa page web:

```
<script TYPE=«text/javascript» SRC=«monscript.js»></script>
```

DOM:

Une interface DOM (**Document Object Model**) pour accéder aux objets le composant:

- . Ces objets sont classés de manière hiérarchique (notion d'arbre).
- . L'objet le plus haut dans la hiérarchie est l'objet de la **classe window** (fenêtre).
- . Dans cette fenêtre, il y a un document HTML: **c'est l'objet document**. Donc, l'objet window contient l'objet document.

Exemple d'utilisation du DOM:

Dans ce document HTML suivant, on pourrait donc accéder aux objets de la manière suivante:

```
// Accès à une propriété de l'objet button  
window.document.form.button.value = «Déterminer»;
```

```
// Accès à une méthode de l'objet button  
window.document.form.button.focus();
```

```
// Accès à une méthode de l'objet window  
window.alert(«Hello world»);
```

Les événements:

Quelques événements classiques gérés en JavaScript:

- . L'utilisateur clique sur le bouton, un lien ou tout autre élément: **click**
- . La page est chargée par le navigateur: **Load**
- . L'utilisateur quitte la page: **Unload**
- . L'utilisateur place le pointeur de la souris sur un élément: **MouseOver**
- . Le pointeur de la souris quitte un lien ou tout autre élément: **MouseOut**
- . Un élément de formulaire a le focus (devient la zone d'entrée active): **Focus**
- . Un élément de formulaire perd le focus: **Blur**
- . La valeur d'un champ de formulaire est modifiée: **Change**
- . L'utilisateur sélectionne un champ dans un élément de formulaire: **Select**
- . L'utilisateur clique sur le bouton submit pour envoyer un formulaire: **Submit**
- . L'utilisateur appuie sur une touche: **Keydown**

Exemple d'événement:

```
<form>
  <input TYPE=«button» VALUE=«Cliquez-ici» onClick=«alert('Vous
avez bien cliqué ici !')»>
</form>
```

Les variables:

// Déclaration donc Local (uniquement dans le script ou la fonction) :

```
var vloc= 0;
```

// Pas déclarée donc Global (en tout point du document) :

```
vglob= 0;
```

Les fonctions:

// Définition:

```
function mafonction(param1, ..., paramN)
{
//Code JavaScript
//...
return variable_ou_valeur ;
}
```

//Appel:

```
var res=mafonction(var1, var2, varN);
```

La structure conditionnelle:

Les choix (sélecteur)

```
if (<condition>)
{
  <action>;
  <action>;
}
```

Exemple:

```
if (j==5)
{
somme = somme+1;
}
else
{
somme=somme - 1;
```

Switch

```
case valeur1:
    //Instructions
[break;]
```

```
case valeurN:
    //Instructions
[break;]
```

```
[default:
    // Instructions
[break;]]
```

Les structures répétitives: (Les boucles)

while

Formulation générale:

```
while(<condition>)
{
<action>;
...
}
```

Exemple:

```
Chiffre=0;
Somme=0;
while(chiffre<5)
{
chiffre=chiffre+1;
somme=somme+chiffre
```

⑩ Do...while

Formulation générale:

```
do
{
<action>
...
}
While (<condition>)
```

⑩ FOR

Formulation générale:

```
for(<expression de depart>;<condition de continuation>;<incrementation>)
{
...
}
```

Exemple:

```
somme=0;
for(chiffre=1;chiffre<=5;chiffre=chiffre+1)
{
    somme=somme+chiffre;
}
```

Signification: La boucle for se déroule de la manière suivante:

Tant que la condition de continuation est vraie:

- ⑩ En partant de l'expression de départ (chiffre=1) on exécute le contenu des accolades;
- ⑩ La variable (chiffre) est incrémentée (chiffre=chiffre+1) et on exécute le contenu des accolades autant de fois que nécessaire;
- ⑩ La boucle ne prend fin que lorsque la condition de continuation devient fausse