

## Table des matières

Introduction :	2
Insertion d'un Script JavaScript dans une page HTML.	2
1. À l'intérieur de la balises <head></head>	3
2. À l'intérieur de la balise <body></body>	4
3. Le code JavaScript se trouve à l'extérieur du document HTML	5
4. Insertion en utilisant un lien <a href> </a>	5
Éléments du langage JavaScript	6
1- Insertion de commentaires :	6
2- Les types et la déclaration de variables	6
Les instructions de contrôles	9
1- L'alternative	9
2- Les itératives	11
Les fonctions	13
Déclaration d'une fonction	13
Quelques objets du langage javascript	14
1. L'objet Window	16
2. L'objet document	17
3. L'objet location	19
4. L'objet navigator	20
5. Javascript et les formulaires	20
Quelques événements du langage JavaScript	22

# Introduction au langage JavaScript

## Introduction :

Le JavaScript est un langage de script incorporé au langage HTML. Il est créé par Netscape en 1995.

Un Script est une portion de code inséré dans une page HTML et qui permet d'exécuter une tâche.

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. Le JavaScript est un langage interprété (besoin du navigateur pour interpréter le code).

Contrairement à HTML c'est un langage **sensible à la CASSE** c'est-à-dire qu'il fait une différence entre les variables contenant des majuscules ou pas.

MaVariable, MaVariable et MavariabLe sont des variables différentes pour JavaScript

## Pré-requis :

- 1- Pour écrire des scripts javascript il faut avoir une bonne connaissance du langage HTML.
- 2- JavaScript est un langage de programmation, l'écriture d'un code JavaScript doit respecter les normes de programmation départementale

## Insertion d'un Script JavaScript dans une page HTML.

Il existe plusieurs façons d'incorporer un Script Javascript dans une page HTML. Toute fois, peu importe la manière ou l'endroit où vous allez l'intégrer il faut préciser au navigateur qu'un script JavaScript va être exécuté en plaçant le code JavaScript entre les balises `<script>` et `</script>` comme suit:

```
<script language="javascript"> votre code javascript ici</script >  
ou  
<script type="text/javascript"> votre code javascript ici</script >  
ou  
<script src ="MonScript.js"></script > (appel d'un fichier externe)
```

La méthode la plus utilisée pour insérer un script JavaScript et celle utilisée par la plus grande majorité des programmeurs consiste à insérer directement le code dans une page HTML aux endroits suivants :

## 1. À l'intérieur de la balises <head></head>

L'insertion d'un script JavaScript à l'intérieur des balises <head></head> permet d'exécuter le code JavaScript d'une **manière différée**, c'est-à-dire que le code est d'abord lu par le navigateur, stocké en mémoire et exécuté à la demande de l'utilisateur par un code correspondant à un événement.

**Le code de cet événement est directement écrit dans le corps du document HTML.**

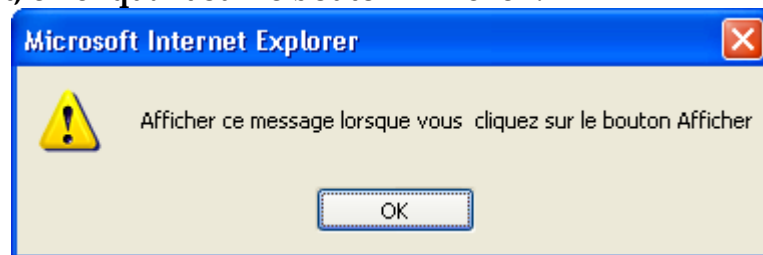
**Exemple de code**

```
<html>
<head>
<title>Page JavaScript </title>

<script type="text/javascript">
// ecrire le code JavaScript Ici
function Message()
    {
        alert("Afficher ce message lorsque vous cliquez sur le bouton Afficher")
    }
</script>
</head>

<body>
Exemple de code JavaScript <br /><br />
<input type = "button" value = "Afficher" Name = "Bouton1" onClick="Message()" />
</body>
</html>
```

**Voici le résultat, en cliquant sur le bouton Afficher :**



**Explications :**

- Remarquer que le code JavaScript est placé entre les balises <head></head>
- Le code JavaScript correspond à une fonction qui permet d'afficher un message à l'aide de la méthode **alert** de JavaScript

- La fonction de l'utilisateur `Message()` est appelée en cliquant sur le bouton « Afficher ». Cette portion du code qui permet d'appeler la fonction `Message()` est directement écrite dans le corps (entre `<body>` et `</body>`) du code HTML

## 2. À l'intérieur de la balise `<body></body>`

On l'appelle exécution directe, car le code s'exécute automatiquement lors du chargement de la page HTML dans le navigateur. Le code JavaScript est placé dans le corps même de la page HTML, entre les balises `<body></body>`

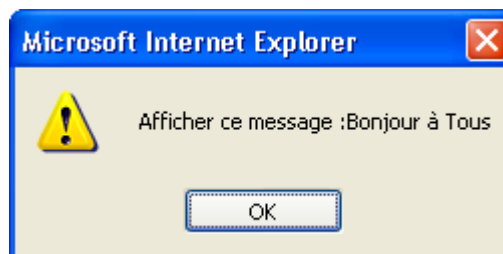
Dans ce cas, le code s'exécutera automatiquement lors du chargement de la page HTML dans le navigateur du client, avant même que le contenu de la page HTML ne s'affiche à l'écran.

### Exemple de code :

```
<html>
<head>
<title>Page javascript </title>
</head>
<body>
Exemple de code JavaScript <br /><br />

<script type="text/javascript">
alert("Afficher ce message :Bonjour à Tous ")
</Script>
</body>
</html>
```

### Résultat de l'exécution



### Explications :

- remarquez que le code JavaScript est placé entre les balises `<body></body>`
- aucun événement n'est nécessaire pour exécuter le code JavaScript

### 3. Le code JavaScript se trouve à l'extérieur du document HTML

Nous pouvons insérer un code JavaScript en appelant un fichier externe dont l'extension est **js** ou **txt**.

Dans ce cas, nous pouvons appeler le fichier en question soit dans le corps du code HTML entre **<body></body>** ou à l'intérieur des balises **<head></head>**.

Pour appeler un fichier externe, il suffit d'utiliser la balise :

```
<script src ="adresse_du_fichier_externe"> </script>
```

#### Exemple

```
<script src ="visualiser.js"> ou <script src ="visualiser.txt">
```

Le fichier Visualiser.txt contient le code suivant :

```
alert("si vous voyez ce message c'est que votre code a été exécuté")
```

#### Contenu du fichier script2.html

```
<html>
<head>
<title>Page JavaScript </title>
<body>
Exemple de code JavaScript <br /><br />
<script src ="Visualiser.js"></script >
</body>
</html>
```

### 4. Insertion en utilisant un lien <a href> </a>

Nous pouvons insérer du code JavaScript en utilisant la balise **<a href> </a>**, soit directement dans le corps du code HTML, ou en faisant appel à une fonction JavaScript, écrite par l'utilisateur et qui est entre les balises **<head></head>**.

Cette manière d'insérer du JavaScript est surtout utilisée lors de la conception des aides en lignes.

#### Syntaxe

```
<a href="javascript:NomFonction()" >cliquer ici pour le résultat </a>
```

#### Exemple

```
<a href ="javascript:Afficher_Note()"> cliquer pour résultat </a>
```

- La fonction Afficher\_Note() est définie dans le **head** un peu plus loin.
- Remarquez les deux points qui suivent le mot réservé javascript

## Éléments du langage JavaScript

### 1- Insertion de commentaires :

Comme tout programme, un code JavaScript doit être documenté. L'insertion de commentaires aide à expliquer certaines instructions du code JavaScript :

- Pour insérer une ligne de commentaire, utiliser // suivi du commentaire
- Pour insérer un commentaire sur plusieurs lignes (consécutives) placer le entre /\* et \*/ (**/\* commentaires \*/**)

### 2- Les types et la déclaration de variables

Il existe principalement trois types de variables JavaScript :

- Le type **number** qui définit un nombre ou un numérique
  - Le type **string** qui définit une chaîne de caractères
  - Le type **boolean** qui définit une variable logique (qui prend la valeur true( vrai) ou la valeur false(fausse)) ;
- 
- Le mot réservé **var** (écrit en minuscule) permet de déclarer une variable.
  - Les variables doivent être déclarées avant leur utilisation.
  - Aucun mot réservé ne doit être utilisé pour les noms de variables
  - Un nom de variable doit commencer par une lettre (majuscule ou minuscule) ou un "\_"
  - Un nom de variables peut comporter des lettres, des chiffres et les caractères \_ et & (les espaces ne sont pas autorisés!)

#### Exemple de mots réservés

abstract, boolean, break, byte, case, catch, char, class, const, continue , debugger  
default, delete, do, double, else, export, extends, false, final, finally, float, for,  
function, etc

Exemple de déclaration de variable.

```
<html>
<head>
<title>Page JavaScript </title>
<script type="text/javascript">
// ecrire le code JavaScript Ici

var Nom ="Emile Zola";

function Afficher();
{
document.write(Nom);
}
</Script>
</head>
<body>
Exemple de code JavaScript<br /><br />
<input type = "button" value ="information" name = "Bouton2" onClick="Afficher()">
</body>
</html>
```

2- **Les opérateurs** : Dans la plus part des cas les opérateurs utilisés en C++ sont les mêmes qu'en javascript

**Exemple :**

```
var Note =85;
var Bonus =5
function Afficher_Note()
{
Note = Note + Bonus;
Note --;
document.write(Note);
}
```

**Ce qui affiche 89**

### Exemple :

Opérateurs	Rôle ou signification
<b>Arithmétique</b>	
+	Addition de nombres ou concaténation de chaînes de caractères
-	Soustraction
*	Multiplication
/	Division
%	Modulo
++	Incrémementation de 1
--	Décrémementation de 1
<b>Logique</b>	
&&	Opérateur logique qui veut dire ET
	Opérateur logique qui veut dire OU
!	Non
<b>Comparaison</b>	
==	Égal (vérifier l'égalité de deux nombre, s'utilise dans un test d'une condition)
!=	Inégal ou différent
>	Supérieur
>=	Supérieur ou égal
<	Inférieur
<=	Inférieur ou égal
<b>Affectation</b>	
=	var1=var2 veut dire var1 reçoit le contenu de var2
+=	var1+=var2 identique à var1 =var1+var2
-=	var1-=var2 identique à var1 =var1-var2
*=	var1*=var2 identique à var1 =var1*var2
/=	var1/=var2 identique à var1 =var1/var2

Exemple de concaténation de chaînes de caractère

### 3- Les instructions

- Les instructions javascript sont placées à l'intérieur d'une fonction ou entre les balises <script> </script> dans le corps du document HTML
- Les instructions doivent se terminées par (;) (un point virgule)
- Les instructions doivent être placées entre { et } (entre deux accolades)



# Les instructions de contrôles

## 1- L'alternative

Exemple : afficher Good morning, si l'heure est plus petite que 10

### Le if

```
<html>
<body>
<script type="text/javascript">
var d = new Date();
var time = d.getHours();

if (time < 10)
{
document.write("<b>Good morning</b>");
}
</script>
</body>
</html>
```

### Le if ...else

```
<html>
<body>

<script type="text/javascript">
var d = new Date();
var time = d.getHours();

if (time < 10)
{
document.write("<b>Good morning</b>");
}
else
{
document.write("<b>Good day</b>");
}
</script>
</body>
</html>
```

## Le Switch

```
<html>
<body>
<script type="text/javascript">
//You will receive a different greeting based
//on what day it is. Note that Sunday=0,
//Monday=1, Tuesday=2, etc.
var d=new Date();
theDay=d.getDay();
switch (theDay)
{
case 5:
  document.write("Finally Friday");
  break;
case 6:
  document.write("Super Saturday");
  break;
case 0:
  document.write("Sleepy Sunday");
  break;
default:
  document.write("I'm looking forward to this weekend!");
}
</script>
</body>
</html>
```

L'instruction *break* permet d'arrêter une boucle tandis que l'instruction *continue* permet de passer immédiatement à la prochaine itération

## 2- Les itératives

### Le for

```
html>
<body>

<script type="text/javascript">
for (i = 0; i <= 5; i++)
{
document.write("The number is " + i);
document.write("<br />");
}
</script>

</body>
</html>
```

### Le while

```
<html>
<body>

<script type="text/javascript">
var i=0;
while (i<=5)
{
document.write("The number is " + i);
document.write("<br />");
i++;
}
</script>

</body>
</html>
```

## do while

```
<html>
<body>

<script type="text/javascript">
var i = 0;
do
{
document.write("The number is " + i);
document.write("<br />");
i++;
}
while (i <= 5)
</script>
</body>
</html>
```

## Les fonctions

Une fonction contient des instructions qui seront exécutées lors d'un évènement ou d'un appel explicite à cette fonction.

Une fonction peut être définie aussi bien dans la section <body > que dans la section <head>, mais si l'on veut que la fonction soit lue avant d'être exécutée alors il faut la placer dans la section <head>.

Il est recommandé de placer les fonctions dans <head> ou dans un fichier externe.

### Déclaration d'une fonction

```
function Mafonction(var1,var2,...,varX)  
{  
code de la fonction  
}
```

*var1, var2, etc* sont des variables (paramètres) passées à la fonction.

Une fonction sans paramètres doit se terminer par ()

### Exemple

```
<html>  
<head> <title> Exemple 9</title>  
  <script type="text/javascript">  
    function somme_N_entiers (nb)  
    {  
      var somme=0;  
      for (i=1; i <=nb ; i++)  
      {  
        somme=somme + i ;  
        document.write("Pour i = " + i + " , somme = " + somme + "<br />");  
      }  
      return somme;  
    }  
  </script>  
</head>  
  
<body>  
<!-- appel de la fonction -->  
Entrer un nombre <br />  
<input type="text" name="nbre"/>  
<input type = "button" value ="calculer" name = "Bouton2"  
onClick="somme_N_entiers (nbre.value)" />  
</body>  
</html>
```

L'instruction *return* est utilisée lorsque la fonction doit retourner une valeur.

Dans l'exemple de la page 2, la fonction Message() ne retourne aucune valeur.

## Quelques objets du langage javascript

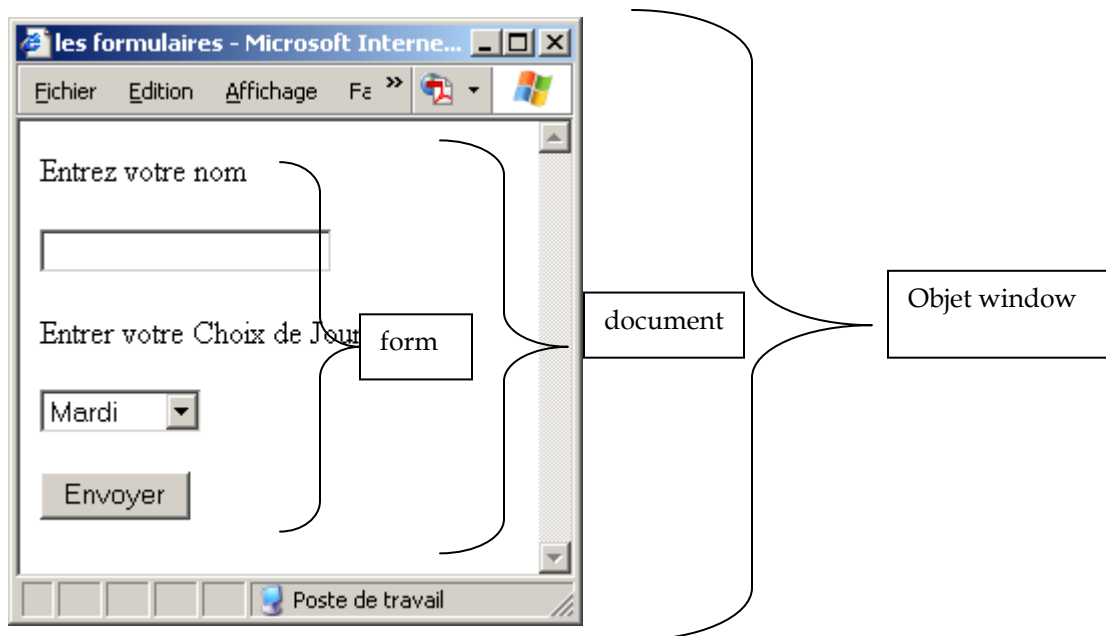
Un objet est défini par un ensemble d'attributs (variables) et d'opérations (actions, fonction ou méthodes)

Exemple :

L'objet Personne est caractérisé par les attributs : Nom, Prénom, Adresse et par les fonctions : Manger (), Marcher() et courir ().

javascript manipule différents objets qui sont hiérarchisés

L'objet window : Toute page Web s'affiche dans une fenêtre (window), cette fenêtre est appelé objet window de javascript



Quelques opérations (méthodes) de l'objet window sont alert(), confirm(), prompt()

### Objet document

L'ensemble des informations inscrites dans la fenêtre sont écrites dans un objet document

Une des opérations de l'objet document est write()

### Objet formulaire

Il arrive que notre document contienne des informations structurées dans un formulaire, dans ce cas, nous dirons qu'elles sont contenues dans une forme ou dans un objet form.

### Les autres objets

Le formulaire tel que montré par la figure précédente contient d'autres objets comme l'objet Liste de choix, l'objet texte et l'objet bouton.

### Accès aux objets.

Pour accéder aux objets d'une fenêtre il est très important :

- de connaître le nom (name) ou l'identifiant (id) de l'objet
- et le chemin d'accès à l'objet.

Exemple.

Pour accéder à la ligne de texte de la figure précédente et chercher sa valeur on dira :

document.forme1.texte1.value :

où forme1 désigne le nom de la forme dans laquelle se trouve la ligne de texte :

texte1 désigne le nom de la ligne de texte.

value est une opération de l'objet texte.

**Exemple de code :**

```
<html>
<head> <title> Exemple 9</title>
  <script type="text/javascript">
    function somme_N_entiers (nb)
    {
      var somme=0;
      for (i=1; i <=nb ; i++)
      {
        somme=somme + i ;
        document.write("Pour i = " + i + " , somme = " + somme + "<br />");
      }
      return somme;
    }
  </script>
</head>
<body>
<!-- appel de la fonction -->
Entrer un nombre <br />
<input type="text" name="nbre"/>
<!-- Attention l'instruction qui suit est sur une seule ligne -->
<input type = "button" value ="calculer" name = "Bouton2"
onClick="somme_N_entiers (nbre.value)"/>
</body>
</html>
```

## 1. L'objet Window

C'est l'objet qui représente une fenêtre de navigation dans le navigateur. C'est l'objet qui contient entre autre l'objet **document** et l'objet **location**

Quelques méthodes de l'objet window.

Méthodes	Description
alert()	La méthode alert() permet d'afficher dans une boîte composée d'une fenêtre et d'un bouton OK le texte qu'on lui fournit en paramètre
close()	Permet de fermer la fenêtre en cours
confirm()	La méthode confirm() est similaire à la méthode alert(), si ce n'est qu'elle permet un choix entre "OK" et "Annuler".
open()	Permet d'ouvrir une nouvelle fenêtre
prompt()	Elle permet à un utilisateur de rentrer une valeur
resizeTo()	Redimensionne une fenêtre : window.resizeTo(500,300). Les valeurs doivent être en pixels



## 2. L'objet document

L'objet document contient toutes les informations d'une page html en cours. Les méthodes reliées à cet objet sont :

Méthodes	Description
<u>close()</u>	Permet de fermer un document. Utilisée automatiquement après un document.write()
<u>getElementById()</u>	Retourne le premier element dont l'id est spécifié. Document.getElementById(id)
<u>getElementsByName()</u>	Renvoie une liste (tableau) des éléments portant le nom de balise donné.
<u>open()</u>	Ouvre un document pour écriture. Cette méthode est appelée automatiquement lors d'un document.write()
<u>write()</u>	Écrit du texte html ou javascript dans un document (une nouvelle page html)
<u>writeln()</u>	Identique à la méthode write(), avec des sauts de lignes après chaque expression

### Exemple 1 :

```
<html>
<head>
<script type="text/javascript">
function Nbre_images()
{
var x=document.getElementsByTagName("img");
alert(x.length);
}
</script>
</head>
<body>

<br />
<br />
<br />
<br />
<input type="button" onclick="Nbre_images()"
value="Combien d'images y'a t-il dans ce documents?" />
</body>
</html>
```

## Autre exemple

```
<html>
<head>
<script type="text/javascript">
function obtenirValeur()
{
  var x=document.getElementById("titre")
  alert(x.innerHTML)
}
</script>
</head>
<body>

<h1 id="titre" onclick="obtenirValeur()">c'est un titre de niveau 1</h1>
<p>appuyer sur le titre pour affichage </p>

</body>
</html>
```

Remarquer l'instruction `alert(x.innerHTML)`.

`x.innerHTML` retourne la valeur de la variable `x`

### 3. L'objet location

Contient les informations concernant l'url de la page en cours. La propriété href permet de renvoyer l'url de la page en cours ou de d'affecter une url.

Exemple (protéger une page web par mot de passe)

```
<html>
<head>

<script type="text/javascript">

var password;
var mdp = "motdepasse";

password=prompt("entrer votre mot de passe:", "");

if (password==mdp) {
  window.location = "interets.html";
} else {
  window.location = "erreur.html";
}

</script>
</head>
</html>
```

#### Autre exemple : liste déroulante

```
<html>
<head>
<title>Nouvelle page 1</title>
<head>
<script type="text/javascript">
function Menu(x)
{
switch(x)
{
case 1 : window.location='programmeur.html';break;
case 2 : window.location='outils.html';break;
case 3 : window.location='interets.html';break;
case 4 : window.location='formulaire.html';break;
}
}
}

</script>
</head>
```

```

<body>
<form name="forme1">
<select name="selection" onChange="Menu(selection.selectedIndex)">
<option>Choisissez une option :</option>
<option >La fonction de programmeur</option>
<option >Nos outils</option>
<option >Nos intérêts</option>
<option >Soumettre sa candidature</option>
</select>
</form>
</body>
</html>

```

#### 4. L'objet navigator

Contient les informations concernant votre navigateur

navigator.appName	retourne le nom du navigateur
navigator.appVersion	retourne la version de votre navigateur

#### 5. Javascript et les formulaires

```

<html>
<head>
<title>Page exemple</title>
<script type="text/javascript">
function Tester(Forme1)
{
if (Forme1.test[0].checked)
{alert("Vous avez choisi:" + " " + Forme1.test[0].value)};
if (Forme1.test[1].checked)
{alert("Vous avez choisi" + " " + Forme1.test[1].value)};
}
}
</script>

</head>
<body>
<form name = "Forme">
<input type = "radio" name = "test" value = "Mathématiques" /> Mathématiques <br />
<input type = "radio" name = "test" value = "Informatique" /> Informatique <br />
<input type = "button" value = "Essayer" Name = "Bouton1" onClick="Tester(Forme)" />

</form>
</body>
</html>

```

#### Autre exemple

```

<html>
<head>
<title>Page exemple</title>
<script type="text/javascript">
function Choix_Reponse(Forme2)
{
if ((Forme2.Case1.checked) == true && (Forme2.Case2.checked) ==true)
{
alert("Vous avez choisi la bonne réponse" )
}

else
{
alert("Vous n'avez pas la bonne réponse");
}
}
</script>
</head>
<body>

<form name = "Forme">
Citer deux cours obligatoires en informatique
<br />
<input type = "checkbox" name ="Case1" value ="Algorithmie"> Algorithmie <br />
<input type = "checkbox" name ="Case2" value = "Mathématiques"> Mathématiques
<br />
<input type = "checkbox" name ="Case3" value ="Espagnol"> Espagnol<br />
<input type = "button" value ="Essayer" Name = "Bouton2"
onClick="Choix_Reponse(Forme)" >
</form>
</body>
</html>

```

## Quelques événements du langage JavaScript

Un événement est une action réalisée par un utilisateur en vue d'accomplir une interactivité comme par exemple un clic sur un bouton.

Javascript manipule différents événements. En voici quelques uns.

Principaux événements :

Événements	Rôle	Exemple de code
onClick	Lorsque l'utilisateur clique sur un bouton, un lien ou tout autre élément	<pre>&lt;input type = "button" name = "Cliquer" value = "Cliquer" onClick = "alert('Ceci est un Click')"&gt;</pre> en général, la syntaxe est : onClick = " NomFonction()"
onLoad	Lorsque la page est chargée par le browser ou le navigateur	Est utilisé comme un attribut de la balise <b>&lt;body&gt;</b> Exemple <pre>&lt;body onLoad = "Bienvenue()" onUnload = "AuRevoir()"&gt;</pre> Dans <b>&lt;head&gt;</b> <pre>function Bienvenue() { var Message = "Bienvenue sur mon site: Bonne navigation"; alert(Message); }</pre>
onUnload	Lorsque l'utilisateur quitte la page.	Est utilisé comme un attribut de la balise <b>&lt;body&gt;</b> Exemple (voir onLoad)
onmouseover	Lorsque l'utilisateur place le pointeur de la souris sur un lien, une image ou tout autre élément.	<pre>&lt;html&gt; &lt;head&gt; &lt;script type="text/javascript"&gt; function mouseOver() { document.getElementById("b1").src = "Flash.jpg"; } function mouseOut() { document.getElementById("b1").src = "photoshop.jpg"; } &lt;/script&gt;</pre>

onmouseout	Lorsque le pointeur de la souris quitte un lien ou tout autre élément.	<pre> &lt;/head&gt;  &lt;body&gt; &lt;a href="http://www.clg.qc.ca" target="_blank"&gt; &lt;img border="0" alt="image animée!" src="photoshop.jpg" id="b1" onmouseover="mouseOver()" onmouseout="mouseOut()" /&gt;&lt;/a&gt; &lt;/body&gt; &lt;/html&gt; </pre>
onChange	Lorsque la valeur d'un champ de formulaire est modifiée	<pre> &lt;body&gt; &lt;form name="forme1"&gt; &lt;select name="selection" onChange="Menu(selection.selectedIndex)"&gt; &lt;option&gt;Choisissez une option :&lt;/option&gt; &lt;option &gt;La fonction de programmeur&lt;/option&gt; &lt;option &gt;Nos outils&lt;/option&gt; &lt;option &gt;Nos intérêts&lt;/option&gt; &lt;option &gt;Soumettre sa candidature&lt;/option&gt; &lt;/select&gt; &lt;/form&gt; &lt;/body&gt;&lt;/html&gt; </pre> <p>La fonction menu() est définie dans le head</p>
onFocus	Lorsqu'un élément de formulaire a le focus c-à-d devient la zone d'entrée active	<pre> &lt;html&gt; &lt;head&gt; &lt;script type="text/javascript"&gt; function setStyle(x) { y=document.getElementById(x); y.style.background="yellow"; } &lt;/script&gt; &lt;/head&gt; &lt;body&gt; First name: &lt;input type="text" onfocus="setStyle(id)" id="fname" /&gt; &lt;br /&gt; Last name: &lt;input type="text" onfocus="setStyle(id)" id="lname" /&gt; &lt;/body&gt; </pre>

onBlur	Lorsqu'un élément de formulaire perd le focus c-à-d que l'utilisateur clique hors du champs et que la zone d'entrée n'est plus active	
--------	---	--

## Événements et balises HTML

(Pour plus de détail allez sur [http://www.w3schools.com/jsref/dom\\_obj\\_event.asp](http://www.w3schools.com/jsref/dom_obj_event.asp)Source:

Événements	Balises HTML	Objets javascript
<b>onclick</b>	<a>, <address>, <area>, <b>, <bdo>, <big>, <blockquote>, <body>, <button>, <caption>, <cite>, <code>, <dd>, <dfn>, <div>, <dl>, <dt>, <em>, <fieldset>, <form>, <h1> to <h6>, <hr>, <i>, <img>, <input>, <kbd>, <label>, <legend>, <li>, <map>, <object>, <ol>, <p>, <pre>, <samp>, <select>, <small>, <span>, <strong>, <sub>, <sup>, <table>, <tbody>, <td>, <textarea>, <tfoot>, <th>, <thead>, <tr>, <tt>, <ul>, <var> >	button, document, checkbox, link, radio, reset, submit
<b>onchange</b>	<input type="text">, <select>, <textarea>	fileUpload, select, text, textarea
<b>onmouseover</b> <b>onmouseout</b>	<a>, <address>, <area>, <b>, <bdo>, <big>, <blockquote>, <body>, <button>, <caption>, <cite>, <code>, <dd>, <dfn>, <div>, <dl>, <dt>, <em>, <fieldset>, <form>, <h1> to <h6>, <hr>, <i>, <img>, <input>, <kbd>, <label>, <legend>, <li>, <map>, <ol>, <p>, <pre>, <samp>, <select>, <small>, <span>, <strong>, <sub>, <sup>, <table>, <tbody>, <td>, <textarea>, <tfoot>, <th>, <thead>, <tr>, <tt>, <ul>, <var>	layer, link
<b>onload</b>	<body>, <frame>, <frameset>, <iframe>, <img>, <link>	image, layer, window
<b>onunload</b>	<body>, <frameset>	window