

SOMMAIRE

Introduction	3
DECLARATION	3
LA BALISE <SCRIPT>	3
Utiliser script externe :.....	3
LES COMMENTAIRES.....	3
MASQUER LE SCRIPT POUR LES ANCIENS BROWSERS.....	3
METHODE :.....	4
Message d'alerte :.....	4
AFFICHER DU TEXTE :	4
LA METHODE WRITE() :	4
L'INSTRUCTION WRITELN()	4
variable.bold();.....	4
variable.italics();.....	4
variable.fontcolor(color);.....	4
variable.fontSize(x);.....	4
variable.strike();.....	5
INSTRUCTIONS DE FORMATAGE DE DOCUMENT	5
document.bgColor :	5
document.fgColor :	5
document.alinkColor :	5
document.linkColor :	5
document.vlinkColor.....	5
UTILISER DES VARIABLES :	5
LA DECLARATION DE VARIABLE.....	5
LES TYPES DONNEES SOUS JAVASCRIPT.....	5
LES MOTS RESERVES :	6
VARIABLES GLOBALES ET VARIABLES LOCALES.....	6
LES OPERATEURS DE CALCUL	7
LES OPERATEURS DE COMPARAISON.....	7
LES OPERATEURS ASSOCIATIFS	7
LES OPERATEURS LOGIQUES :.....	7
LES OPERATEURS D'INCREMENTATION.....	7
LA PRIORITE DES OPERATEURS JAVASCRIPT	8
LES FONCTIONS	8

DECLARATION DES FONCTIONS :.....	8
L'APPEL D'UNE FONCTION :	8
LES FONCTIONS DANS <HEAD>...</HEAD>	8
RETOURNER UNE VALEUR :	9
LES EVENEMENTS :	9
LES GESTIONNAIRES D'EVENEMENTS :	10
GESTIONNAIRES D'EVENEMENT DISPONIBLES EN JAVASCRIPT :	10
CHANGEMENT D'IMAGES :	11
LES CONDITIONS :	11
IF	11
Ternaire	11
BOUCLE :	11
WHILE :	12
BREAK :	12
CONTINUE :	12
UNE MINUTERIE :	12
Pour arrêter le temporisateur avant l'expiration du délai fixé, il y a :	12
ouvrir une nouvelle fenetre :	12
ouvrir :	12
pour fermer :	12
Ecrivez dans une fenetre :	12
Conversion chaine → nombre.....	12
Fonctions mathématiques :	13

Introduction

Javascript est donc une extension du code Html des pages Web. Les scripts, qui s'ajoutent ici 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.

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.

DECLARATION

LA BALISE <SCRIPT> :

```
<SCRIPT LANGUAGE="Javascript">.  
</SCRIPT>.
```

Utiliser script externe :

```
<SCRIPT LANGUAGE='javascript' SRC='http://site.com/javascript.js'></SCRIPT>
```

LES COMMENTAIRES

```
// commentaire  
/* commentaire */
```

MASQUER LE SCRIPT POUR LES ANCIENS BROWSERS

```
<SCRIPT LANGUAGE="javascript">  
<!-- Masquer le script pour les anciens browsers  
...  
programme Javascript  
...  
// Cesser de masquer le script -->  
</SCRIPT>
```

REMARQUE :

- Javascript est « case sensitive »
- Si vous souhaitez utiliser des guillemets dans vos chaînes de caractères, tapez \" ou \' pour les différencier vis à vis du compilateur.

METHODE :

Message d'alerte :

```
alert("hello wordl !");
```

Notation classique : window.alert() ;

AFFICHER DU TEXTE :

Appler une méthode :

L'appel de la méthode se fait selon la notation :

nom_de_l'objet.nom_de_la_méthode

Pour appeler la méthode write() du document, on notera

LA METHODE WRITE() :

```
document.write("hello world!")
```

L'INSTRUCTION WRITELN()

Write avec retour à la ligne

```
<PRE>
<SCRIPT LANGUAGE="Javascript">
<--
document.writeln("Ligne 1");
document.writeln("Ligne 2");
//-->
</SCRIPT>
</PRE>
```

variable.bold();

affichera la variable comme si elle était comprise entre les balises Html .

variable.italics();

affichera la variable comme si elle était comprise entre les balises <I></I>.

variable.fontcolor(color);

```
str1="Some words";
str2="red";
document.write("<FONT COLOR='red'>" +str1+"</FONT>");
document.write("<FONT COLOR='red'>" +"Something</FONT>");
document.write(str1.fontcolor(str2));
document.write(str1.fontcolor("red"));
```

variable.fontSize(x);

 où x est un nombre de 1 à 7 ou exprimé en plus ou en moins par rapport à 0
par exemple -2, -1, +1, +2

```
str="Something";
x=3;
document.write("<FONT SIZE=3>" +str+"</FONT>");
document.write("<FONT SIZE=3>" + "Something</FONT>");
document.write(str.fontSize(3));
document.write(str.fontSize(x));
```

variable.strike();

Texte barré

```
str="Something";
document.write("<STRIKE>" +str + "</STRIKE>");
document.write("<STRIKE>Something" + "</STRIKE>");
document.write(str.strike());
document.write("Something".strike());
```

INSTRUCTIONS DE FORMATAGE DE DOCUMENT

document.bgColor :

Cette instruction permet de spécifier la couleur d'arrière-plan d'un objet document

document.fgColor :

Cette instruction permet de spécifier la couleur d'avant-plan

document.alinkColor :

Cette instruction permet de spécifier la couleur d'un lien actif (après le clic de la souris mais avant de quitter le lien)

document.linkColor :

Cette instruction permet de spécifier la couleur d'un hyperlien d'un objet document.

document.vlinkColor

Cette instruction permet de spécifier la couleur d'un hyperlien déjà visité

UTILISER DES VARIABLES :

LA DECLARATION DE VARIABLE

- *façon Explicite :*
var Numero = 1
var Prenom = "Luc"
- *Façon implicite :*
Numero = 1
Prenom = "Luc"

LES TYPES DONNEES SOUS JAVASCRIPT

Type	Description
Des nombres	Tout nombre entier ou avec virgule tel que 22 ou 3.1416
Des chaînes de caractères	Toute suite de caractères comprise entre guillemets telle que "suite de caractères"
Des booléens	Les mots true pour vrai et false pour faux
Le mot null	Mot spécial qui représente pas de valeur

REMARQUE :

Notons aussi que contrairement au langage C ou C++, Il ne faut pas déclarer le type de données d'une variable. On n'a donc pas besoin de int, float, double, char et autres long en Javascript.

Exemple :

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="Javascript">
<!--
var texte = "Mon chiffre préféré est le "
var variable = 7
document.write(texte + variable);
//-->
</SCRIPT>
</BODY>
</HTML>
```

LES MOTS RESERVES :

A	abstract
B	boolean break byte
C	case catch char class const continue
D	default do double
E	else extends
F	false final finally float for function
G	goto
I	if implements import in instanceof int interface
L	long
N	native new null
P	package private protected public
R	return
S	short static super switch synchronized
T	this throw throws transient true try
V	var void
W	while with

VARIABLES GLOBALES ET VARIABLES LOCALES

Dans une fonction, une variable déclarée par le mot clé var aura une portée limitée à cette seule fonction. On ne pourra donc pas l'exploiter ailleurs dans le script. D'où son nom de locale. Par contre, toujours dans une fonction, si la variable est déclarée contextuellement (sans utiliser le mot var), sa portée sera globale.

LES OPERATEURS DE CALCUL

Signe	Nom	Signification	Exemple	Résultat
+	plus	addition	$x + 3$	14
-	moins	soustraction	$x - 3$	8
*	multiplié par	multiplication	$x * 2$	22
/	divisé	par division	$x / 2$	5.5
%	modulo	reste de la division par	$x \% 5$	1
=	a la valeur	affectation	$x = 5$	5

LES OPERATEURS DE COMPARAISON

Signe	Nom	Exemple	Résultat
==	égal	$x == 11$	true
<	inférieur	$x < 11$	false
<=	inférieur ou égal	$x <= 11$	true
>	supérieur	$x > 11$	false
>=	supérieur ou égal	$x >= 11$	true
!=	différent	$x != 11$	false

LES OPERATEURS ASSOCIATIFS

Signe	Description	Exemple	Signification	Résultat
+=	plus égal	$x += y$	$x = x + y$	16
-=	moins égal	$x -= y$	$x = x - y$	6
*=	multiplié égal	$x *= y$	$x = x * y$	55
/=	divisé égal	$x /= y$	$x = x / y$	2.2

LES OPERATEURS LOGIQUES :

Signe	Nom	Exemple	Signification
&&	et	$(condition1) \&\& (condition2)$	condition1 et condition2
	ou	$(condition1) (condition2)$	condition1 ou condition2

LES OPERATEURS D'INCREMENTATION

Signe	Description	Exemple	Signification	Résultat
x++	incrémenter (x++ est le même que x=x+1)	$y = x++$	3 puis plus 1	4
x--	décrémenter (x-- est le même que x=x-1)	$y = x--$	3 puis moins 1	2

LA PRIORITE DES OPERATEURS JAVASCRIPT

Opération	Opérateur
,	virgule ou séparateur de liste
= += -= *= /= %/=	affectation
? :	opérateur conditionnel
	ou logique
&&	et logique
== !=	égalité
< <= >= >	relationnel
+ -	addition soustraction
* /	multiplier diviser
! - ++ --	unaire
()	parenthèses

LES FONCTIONS

En Javascript, il existe deux types de fonctions :

- les fonctions propres à Javascript. On les appelle des "méthodes". Elles sont associées à un objet bien particulier comme c'était le cas de la méthode Alert() avec l'objet window.
- les fonctions écrites par vous-même pour les besoins de votre script. C'est à celleslà que nous nous intéressons maintenant.

DECLARATION DES FONCTIONS :

```
function nom_de_la_fonction(arguments)
{
... code des instructions ...
}
```

L'APPEL D'UNE FONCTION :

```
nom_de_la_fonction();
```

LES FONCTIONS DANS <HEAD>...</HEAD>


```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="Javascript">
<--
function message() {
document.write("Bienvenue à ma page");
}
//-->
</SCRIPT>
</HEAD>
<BODY onLoad="message()">
</BODY>
</HTML>

```

RETOURNER UNE VALEUR :

```

function cube(nombre)
{
var cube = nombre*nombre*nombre
return cube;
}

```

LES EVENEMENTS :

Description	Evénement
Lorsque l'utilisateur clique sur un bouton, un lien ou tout autre élément.	Clik
Lorsque la page est chargée par le browser ou le navigateur.	Load
Lorsque l'utilisateur quitte la page.	Unload
Lorsque l'utilisateur place le pointeur de la souris sur un lien ou tout autre élément.	MouseOver
Lorsque le pointeur de la souris quitte un lien ou tout autre élément. Attention : Javascript 1.1 (donc pas sous MSIE 3.0 et Netscape 2).	MouseOut
Lorsque un élément de formulaire a le focus c-à-d devient la zone d'entrée active.	Focus
Lorsque 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.	Blur
Lorsque la valeur d'un champ de formulaire est modifiée.	Change
Lorsque l'utilisateur sélectionne un champ dans un élément de formulaire.	Select
Lorsque l'utilisateur clique sur le bouton Submit pour envoyer un formulaire.	Submit

LES GESTIONNAIRES D'EVENEMENTS :

Pour être efficace, il faut qu'à ces événements soient associées les actions prévues par vous. C'est le rôle des gestionnaires d'événements. La syntaxe est :

```
onévénement="fonction()
```

exemple :

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="style.css">
  <title></title>
  <script type="text/javascript">

    function bonjour(){
      alert("bonjour");
    }

    function au_revoir(){
      alert("au revoir");
    }

  </script>
</head>
<body onLoad='bonjour()'onUnload='au_revoir()>

</body>
</html>
```

GESTIONNAIRES D'EVENEMENT DISPONIBLES EN JAVASCRIPT :

Objets	Gestionnaires d'événement disponibles
Fenêtre	onLoad, onUnload
Lien hypertexte	onClick, onMouseOver, on mouseOut
Élément de texte	onBlur, onChange, onFocus, onSelect
Élément de zone de texte	onBlur, onChange, onFocus, onSelect
Élément bouton	onClick
Case à cocher	onClick
Bouton Radio	onClick
Liste de sélectionon	Blur, onChange, onFocus
Bouton Submit	onClick
Bouton Reset	onClick

Exemple :

```
<a href="#" onMouseOver='alert("bonjour");'>cliquer ici</a>
```

CHANGEMENT D'IMAGES :

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="style.css">
  <title></title>
  <script type="text/javascript">

    function showUp(){
      document.images["imageTest"].src="2.jpg";
    }

    function showDown(){
      document.images["imageTest"].src="1.jpg";
    }

  </script>
</head>
<body>
  <a href="#" onMouseOver="showUp();" onMouseOut="showDown();">cliquer ici</a>
  
</body>
</html>
```

LES CONDITIONS :

IF

```
if (condition vraie) {
  instructions1;
}
else {
  instructions2;
}
```

Ternaire

```
(expression) ? instruction a : instruction b
```

BOUCLE :

```
for (valeur initiale ; condition ; progression) {
  instructions;
}
```

Exemple :

```
for (var i = 0; i < 10; i++) {
  document.write("hello\t");
}
```

WHILE :

```
while (condition vraie){  
    continuer à faire quelque chose  
}
```

BREAK :

L'instruction break permet d'interrompre prématurément une boucle for ou while.

CONTINUE :

L'instruction continue permet de sauter une instruction dans une boucle for ou while et de continuer ensuite le bouclage (sans sortir de celui-ci comme le fait break).

UNE MINUTERIE :

```
nom_du_compteur = setTimeout("fonction_appelée()", temps en milliseconde)
```

Ainsi, setTimeout("demarrer()",5000) va lancer la fonction demarer() après 5 secondes.

Pour arrêter le temporisateur avant l'expiration du délai fixé, il y a :

```
clearTimeout(nom_du_compteur)
```

ouvrir une nouvelle fenetre :

ouvrir :

```
open("test.html","test","width=100px,height=100px");
```

pour fermer :

```
<input type="button" value="close" onClick="self.close();"/>
```

Ecrivez dans une fenetre :

```
msg=open("", "test", "width=100px,height=100px");  
msg.document.write("hello");
```

Conversion chaine → nombre

Int parseInt(string chaine) : conversion d'une chaine en entier

Float parseFloat(string chaine) : conversion d'une chaine en réel

Fonctions mathématiques :

@ Principe général

- Appel des méthodes de l'objet Math

@ Listes des méthodes

- $\text{abs}(x)$, $\text{acos}(x)$, $\text{asin}(x)$, $\text{atan}(x)$, $\text{cos}(x)$, $\text{ln}(x)$, $\text{log}(x)$, $\text{round}(x)$, $\text{sin}(x)$, $\text{sqrt}(x)$, $\text{tan}(x)$: applique la fonction appropriée à x
- $\text{ceil}(x)$: renvoie le plus petit entier supérieur ou égal à x
- $\text{exp}(x)$: renvoie e (exponentielle) à la puissance x
- $\text{floor}(x)$: renvoie le plus grand entier inférieur ou égal à x
- $\text{max}(x,y)$: renvoie la plus grande des valeurs de x et y
- $\text{min}(x,y)$: renvoie la plus petite des valeurs de x et y
- $\text{pow}(x,y)$: renvoie x à la puissance y
- $\text{random}(x)$: renvoie un nombre aléatoire compris entre 0 et 1