

JAVASCRIPT: CHAPITRE 1

Introduction

- JavaScript est un langage de scripts incorporé aux balises Html, permet d'améliorer la présentation et l'interactivité des pages Web.
- JavaScript est un langage de programmation interprété par le navigateur. Le JavaScript est un langage « client », c'est-à-dire exécuté chez l'utilisateur lorsque la page Web est chargée.
- JavaScript est un langage orienté objet : chaque objet possède des méthodes (ou fonctions), des propriétés.
- JavaScript ne doit pas être confondu avec Java, qui est un langage beaucoup plus complexe permettant de créer des applications autonomes.

Introduction

3

Les principales applications de JavaScript sont :

- la validation de formulaires avant envoi au serveur,
- la détection de type et de version de navigateur,
- la gestion de fenêtres (*pop-ups, cadres...*),
- la préservation d'informations contextuelles,
- la génération automatique de date,
- la détection de plug-in,
- les calculs côté client,
- les effets d'images...

Insérer du code JavaScript

4

- Pour pouvoir fonctionner, le code JavaScript d'une page HTML doit en principe être placé à l'intérieur d'une balise *<SCRIPT>*.
- Une page HTML peut contenir plusieurs balises *<SCRIPT>*, mais elles ne doivent pas être imbriquées.

Inserer du code jsript

5

Methode 1 : Dans l'en-tête **<head>** du fichier HTML
la balise **<script>** **</script>**

Exemple :

```
<head>
  <meta charset="utf-8" />
  <script>
    alert("My First JavaScript");
  </script>
  <title>Premiers tests du jsript</title>
</head>
```

Exemple 2 : fichier externe

```
<head>
  <meta charset="utf-8" />
  <script src="myScript.js"></script>
  <title>Premiers tests du jsript</title>
</head>
```

Inserer du code jscript

6

Methode 2 : Dans le **<body>** du fichier HTML
la balise **<script>** **</script>**

Exemple :

```
<body>
<h1>My Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>
  <script>
    function myFunction()
    {
      document.getElementById("demo").innerHTML="My First JavaScript Function";
    }
  </script>
</body>
```

Exemple 2 : fichier externe

```
<!DOCTYPE html>
<html>
<body>
  <script src="myScript.js"></script>
</body>
</html>
```

Quelques applications du Jscript

7

- Ecrire du code html :

```
document.write("<h1>This is a heading</h1>");  
document.write("<p>This is a paragraph</p>");
```

- Agir aux événements :

```
<button type="button" onclick="alert('Welcome!')">Click  
Me!</button>
```

- Changer le contenu HTML

```
x=document.getElementById("demo") //Find the element  
x.innerHTML="Hello JavaScript"; //Change the content
```

Quelques applications du Jscript

8

- Changer une image html

```
element=document.getElementById('myimage')  
element.src="pic_bulboff.gif";
```

- Changer les styles css

```
x=document.getElementById("demo")  
x.style.color="#ff0000";
```

- Validation des formulaires

```
var x=document.getElementById("demo").value;  
if(x=="" || isNaN(x))  
{  
  alert("Not Numeric");  
}
```


La syntaxe JavaScript

9

- une instruction JScript se termine toujours par un point-virgule
- En JavaScript les commentaires peuvent prendre deux formes correspondant au langage c.

//Ceci est un commentaire monoligne

/* Ceci est un commentaire

Sur plusieurs lignes */

- C'est un langage sensible à la casse Concrètement en JavaScript, Monobjet n'est pas identique à monobjet.

Les variables

10

- **Déclaration:** `var nom_de_la_variable ;`
Exemple: `var mavariable1, mavariable2, mavariable3 ;`
- **Déclaration contextuelle:** `mavariable=1;`
- **Affectation:**
Exemple: `var mavariable = 10 ;`
 `var somme;`
 `somme=mavariable1+mavariable2 ;`
- **Affichage des variables :**
 - ✓ dans une boite de dialogue
 `alert("le résultat de l'addition est : "+somme);`
 - ✓ Dans la page html
 `document.write("le résultat de la multiplication est : "+produit);`

Les variables

11

JavaScript utilise 4 types de données :

- 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

Les variables

12

- Conversion: convertir une variable d'un type à un autre
`var variablenumerique=parseInt(variabletexte) ;`
`var variablenumerique=parseFloat(variabletexte) ;`
`Var variabletexte=variablenumerique.toString() ;`
- `isNaN`
détermine si le parametre n'est pas un nombre (NaN : Not a Number).

```
isNaN("un nombre") //retourne true  
isNaN(20) //retourne false
```

- Règles de nommage :
 - ✓ Il faut pas utiliser des espaces dans les noms des variables utiliser underscore '_';
 - ✓ Utiliser des noms explicites dependent du problème.
 - ✓ Interdit d'utiliser les mots réservés

Les Variables

13

- Une variable correctement déclarée mais dont aucune valeur n'est affectée, est indéfinie (undefined).
- les variables peuvent être globales ou locales.
- Une variable globale est déclarée en début de script et est accessible à n'importe quel endroit du programme.
- Une variable locale est déclarée à l'intérieur d'une fonction et n'est utilisable que dans la fonction elle-même.

Les operateurs

14

- Operateurs arithmétiques : les memes du langage C
+ , - , * , / , % , ++ , -- , += ...
- Opérateurs de comparaison : > , >= , <= , < , == , !=
- Opérateurs logiques : && , || , !
- Opérateur de concatenation : +
alert("Votre nom complet est : "+nomcomplet);

Les instructions conditionnelle

15

Formulation générale :

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

Exemple :

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

Les instructions conditionnelle

16

Else if :

```
if (test) {  
  Ligne 1 d'instruction ;  
  Ligne 2 d'instruction ;  
}  
else if (test) {  
  Ligne 1 d'instruction ;  
  Ligne 2 d'instruction ;}  
}  
else  
{  
  Ligne 1 d'instruction ;  
  Ligne 2 d'instruction ;  
}
```


Les instructions conditionnelle

17

switch:

```
switch (reponse)
{
case "A":alert("Pas tout-à-fait, ré-essayez");
        break;
case "B":alert("Bravo ! C'est exact");
        break;
case "C":alert("Pas tout-à-fait, ré-essayez");
        break;
default:alert("Votre réponse ne correspond pas aux propositions");
        break;
}
```

Les boucles

18

For :

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

Exemple:

```
for (compteur=0; compteur<10;compteur++)  
{  
document.write("la valeur du compteur est de : " +compteur+"<BR>");  
}
```

Les boucles

19

While :

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

Exemple:

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

Les boucles

20

■ Do..while :

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

Exemple:

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

Entrée et sortie de données avec JavaScript

21

- 3 types de boîtes de messages peuvent être affichés en utilisant JavaScript : **Alerte, Confirmation et Invite**
 - **Méthode alert()**
sert à afficher à l'utilisateur des informations simples de type texte. Une fois que ce dernier a lu le message, il doit cliquer sur OK pour faire disparaître la boîte
 - **Méthode confirm()**

permet à l'utilisateur de choisir entre les boutons OK et Annuler.
 - **Méthode prompt()**

La méthode prompt() permet à l'utilisateur de taper son propre message en réponse à la question posée
- La méthode **document.write** permet d'écrire du code HTML dans la page WEB

Entrée et sortie de données avec JavaScript

22

- **alert()** : methode de l'objet window
l'affichage d'une boîte de dialogue comprenant un message d'avertissement



Entrée et sortie de données avec JavaScript

23

- **confirm()** : La méthode `confirm()` de l'objet `window` affiche une boîte de dialogue avec un message suivi des 2 boutons **OK** et **Annuler**.

```
var reponse=window.confirm("ok = true, Annuler = false");  
alert("la valeur de la variable réponse est : "+reponse);
```



Entrée et sortie de données avec JavaScript

24

- **prompt()**: permet, en plus, d'afficher un message pour saisir une valeur dans un champ, appelé invite.

```
<script language="javascript">
```

```
nom=prompt("Quel est votre nom ?", "Saisir votre nom ici");
```

```
alert("Votre nom complet est : \r"+nom);
```

```
</script>
```



Les Fonctions

25

- Une fonction est un groupe de ligne(s) de code de programmation destiné à exécuter une tâche bien spécifique et que l'on pourra utiliser à plusieurs reprises. De plus, l'usage des fonctions améliorera grandement la lisibilité de votre script.
- 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.

Les Fonctions

26

Déclaration d'une fonction:

La syntaxe d'une déclaration de fonction est la suivante :

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

Exemple:

```
function carre(nombre)
{
resultat = nombre * nombre ;
return(resultat)
}
```

Les Fonctions

27

L'appel d'une fonction:

L'appel d'une fonction se fait le plus simplement du monde par le nom de la fonction (avec les parenthèses).

Exemple:

```
Var c;
```

```
c=carre(3);
```

Les Fonctions

28

Variables locales et variables globales

- Une variable déclarée dans une fonction par le mot clé `var` aura une portée limitée à cette seule fonction. On ne pourra donc pas l'exploiter ailleurs dans le script. On l'appelle donc variable locale.
- Si la variable est déclarée contextuellement (sans utiliser le mot `var`), sa portée sera globale -- et pour être tout à fait précis, une fois que la fonction aura été exécutée--.