<Pratique de JavaScript

Pour étendre les possibilités des pages HTML, pour les rendre plus interactives et pour effectuer certains traitements et contrôles au niveau du client avant de soumettre la page web à un serveur, le développeur utilise des scripts. Deux langages de scripts sont très utilisés actuellement sur le marché: JavaScript(de Netscape) et VBScript(de Microsoft).

Par rapport à VBScript, JavaScript présente l'avantage d'être supporté par les deux navigateurs les plus connus au monde Netscape Navigator et Internet Explorer alors que VBScript n'est pas supporté par Netscape Navigator. En plus il est léger et orienté objet.

Remarque : Malgré quelques ressemblances apparente JavaScript est très différent de Java.

Utilisation

Les Scripts JavaScript sont intégrés dans une page web de plusieurs manières :

* En écrivant le script entre une balise <Script> et </Script> :

<Script Language="JavaScript">

<!--

Ecriture Script

-->

<NoScript>

Traitement si le navigateur n'accepte pas les scripts

</NoScript>

</Script>

* En associant le script à un événement :

<NomBalise…..onEvénement="Ecriture Script">

* En associant le script à un lien :

<a href="JavaScript:Ecriture Script"…>…</a>

Remarque : Pour empêcher que le script dans href ne remplace le document courant, on applique l'opérateur void, qui neutralise toute valeur ou tout effet de retour : <a href="JavaScript:void(Ecriture Script)">…</a>

* En écrivant le script dans un fichier ayant l'extension .js et en l'appelant dans la page web à l'aide de l'instruction :

<Script Language="JavaScript" src="FichierScript.js> </Script>

Remarque:

* Les Scripts peuvent être placés dans le Head ou dans le Body. Dans le Head ils seront reconnus dés le chargement de la page.
* Un gestionnaire d'événement est la procédure particulière attachée à un événement dans une balise HTML
* Il est possible d'utiliser plusieurs gestionnaires d'événements dans une même balise

Concepts de programmation

***Insertion de commentaires***

* Sur une ligne :

//Commentaire

* Sur plusieurs lignes :

/\*Commentaire

Commentaire\*/

***utilisation des variables***

Pour déclarer une variable :

var Variable1 [=Valeur\_initiale1], Variable2 [=Valeur\_initiale2], …

Remarque : La déclaration se fait de manière implicite si une variable est utilisée dans le script mais qu'elle n'a pas été déclarée à l'aide de var.

***Les types de données :***

String, Number, Boolean, Array, Object, Null, Undefined.

|  |
| --- |
| **var a=10;**  **alert(typeof a); -----> Number**  **var b="Karim";**  **alert(typeof b); -----> String**  **var c=true;**  **alert(typeof c); -----> Boolean**  **var d;**  **alert(typeof d); -----> Undefined**  **var e=undefined;**  **alert(typeof e); -----> Undefined**  **var f= new Date();**  **alert(typeof f); -----> Object**  **var g= new String();**  **alert(typeof g); -----> Object**  **g=null;**  **alert(typeof g); -----> Object** |

***utilisation des tableaux (objet Array)***

* Déclaration:
  + Tableaux à une dimension :
* Le nombre d'éléments du tableau est inconnu :

var tableau=new Array()

* Le nombre d'éléments du tableau est connu mais le contenu est non connu :

var tableau=new Array(NombreEléments)

* Le nombre d'éléments du tableau est connu et le contenu est connu :

var Tableaux=new Array(ElémentTableau1, ElémentTableau2,…)

ou

var Tableaux=[ElémentTableau1, ElémentTableau2,…]

* + Tableaux à plusieurs dimensions : Ils sont gérés comme des tableaux de tableaux à une dimension

var Tableau =new Array(Vide|Nombre\_Elément|ListeEléments)

Tableau[0]=new array(Vide|Nombre\_Eléments|ListeEléments)

Tableau[1]=new array(Vide|Nombre\_Eléments|ListeEléments)

…

* Remplissage
  + Tableaux à une dimension :

Tableau[position]=valeur

ou

Tableau[position]= new array(Vide|Nombre\_Eléments|ListeEléments)

* + Tableaux à plusieurs dimensions :

Tableau[positionLigne,PositionColonne]=valeur

ou

Tableau[positionLigne,PositionColonne]= new array(Vide|Nombre\_Eléments|ListeEléments)

Remarque : Le premier élément du tableau commence à l'indice 0

* Quelques propriétés
  + length : Retourne le nombre d'éléments du tableau (Tableau.length)
* Quelques méthodes
  + join() : Regroupe tous les éléments avec un caractère de séparation aléatoire (Tableau.join())

|  |
| --- |
| [**var**](http://www.toutjavascript.com/reference/reference.php?ref=var)**tab=**[**new**](http://www.toutjavascript.com/reference/reference.php?ref=new)[**Array**](http://www.toutjavascript.com/reference/reference.php?ref=Array)**("Pommes", "Poires", "Ananas", "Cerise");** [**document**](http://www.toutjavascript.com/reference/reference.php?ref=document)**.**[**write**](http://www.toutjavascript.com/reference/reference.php?ref=write)**(tab.**[**join**](http://www.toutjavascript.com/reference/reference.php?ref=join)**());** |

* + join(car) : Regroupe tous les éléments avec un caractère de séparation car (Tableau.join(car))

|  |
| --- |
| [**var**](http://www.toutjavascript.com/reference/reference.php?ref=var)**tab=**[**new**](http://www.toutjavascript.com/reference/reference.php?ref=new)[**Array**](http://www.toutjavascript.com/reference/reference.php?ref=Array)**("Pommes", "Poires", "Ananas", "Cerise");** [**document**](http://www.toutjavascript.com/reference/reference.php?ref=document)**.**[**write**](http://www.toutjavascript.com/reference/reference.php?ref=write)**(tab.**[**join**](http://www.toutjavascript.com/reference/reference.php?ref=join)**(", "));** |

* + reverse() : Inverse l'ordre des éléments contenus dans le tableau (Tableau.reverse())

|  |
| --- |
| [**var**](http://www.toutjavascript.com/reference/reference.php?ref=var)**tab=**[**new**](http://www.toutjavascript.com/reference/reference.php?ref=new)[**Array**](http://www.toutjavascript.com/reference/reference.php?ref=Array)**("Pommes", "Poires", "Ananas", "Cerise");** [**document**](http://www.toutjavascript.com/reference/reference.php?ref=document)**.**[**write**](http://www.toutjavascript.com/reference/reference.php?ref=write)**("Tableau d'origine : " + tab.**[**join**](http://www.toutjavascript.com/reference/reference.php?ref=join)**(", ") + "<BR>"); tab.**[**reverse**](http://www.toutjavascript.com/reference/reference.php?ref=reverse)**()** [**document**](http://www.toutjavascript.com/reference/reference.php?ref=document)**.**[**write**](http://www.toutjavascript.com/reference/reference.php?ref=write)**("Tableau inversé : " + tab.**[**join**](http://www.toutjavascript.com/reference/reference.php?ref=join)**(", "));** |

* + sort() : Trie le suivant l'ordre croissant (Tableau.sort())

|  |
| --- |
| [**var**](http://www.toutjavascript.com/reference/reference.php?ref=var)**tab=**[**new**](http://www.toutjavascript.com/reference/reference.php?ref=new)[**Array**](http://www.toutjavascript.com/reference/reference.php?ref=Array)**("Pommes", "Poires", "Ananas", "Cerise", "Abricot");** [**document**](http://www.toutjavascript.com/reference/reference.php?ref=document)**.**[**write**](http://www.toutjavascript.com/reference/reference.php?ref=write)**("Tableau d'origine : " + tab.**[**join**](http://www.toutjavascript.com/reference/reference.php?ref=join)**(", ") + "<BR>"); tab.**[**sort**](http://www.toutjavascript.com/reference/reference.php?ref=sort)**()** [**document**](http://www.toutjavascript.com/reference/reference.php?ref=document)**.**[**write**](http://www.toutjavascript.com/reference/reference.php?ref=write)**("Tri croissant : " + tab.**[**join**](http://www.toutjavascript.com/reference/reference.php?ref=join)**(", ") + "<BR>"); tab.**[**reverse**](http://www.toutjavascript.com/reference/reference.php?ref=reverse)**()** [**document**](http://www.toutjavascript.com/reference/reference.php?ref=document)**.**[**write**](http://www.toutjavascript.com/reference/reference.php?ref=write)**("Tri décroissant : " + tab.**[**join**](http://www.toutjavascript.com/reference/reference.php?ref=join)**(", "));** |

* + push() : Ajoute des éléments en fin de tableau (tableau.push(String element1 [, ..., String element\_n]))

|  |
| --- |
| [**var**](http://www.toutjavascript.com/reference/reference.php?ref=var)**tab=**[**new**](http://www.toutjavascript.com/reference/reference.php?ref=new)[**Array**](http://www.toutjavascript.com/reference/reference.php?ref=Array)**("Pommes", "Poires", "Ananas", "Cerise"); tab.**[**push**](http://www.toutjavascript.com/reference/reference.php?ref=push)**("Banane", "Fraise")** [**document**](http://www.toutjavascript.com/reference/reference.php?ref=document)**.**[**write**](http://www.toutjavascript.com/reference/reference.php?ref=write)**(tab.**[**join**](http://www.toutjavascript.com/reference/reference.php?ref=join)**(", "));** |

* + pop() : Supprime le dernier élément du tableau (tableau.pop())

|  |
| --- |
| [**var**](http://www.toutjavascript.com/reference/reference.php?ref=var)**tab=**[**new**](http://www.toutjavascript.com/reference/reference.php?ref=new)[**Array**](http://www.toutjavascript.com/reference/reference.php?ref=Array)**("Pommes", "Poires", "Ananas", "Cerise"); tab.**[**pop**](http://www.toutjavascript.com/reference/reference.php?ref=pop)**() ;** [**document**](http://www.toutjavascript.com/reference/reference.php?ref=document)**.**[**write**](http://www.toutjavascript.com/reference/reference.php?ref=write)**(tab.**[**join**](http://www.toutjavascript.com/reference/reference.php?ref=join)**(", "));** |

***Structures conditionnelles***

* if

if (condition)

{Instructions}

else

{Instructions}

**if (time<20)  
  {  
  x="Good day";  
  }  
else  
  {  
  x="Good evening";  
  }**

**if (time<10)  
  {  
  x="Good morning";  
  }  
else if (time<20)  
  {  
  x="Good day";  
  }  
else  
  {  
  x="Good evening";  
  }**

* switch

switch(expression)

{

case const1:

instructions

break

case const2:

Instructions

break

...

default:

Instructions

}

|  |
| --- |
| **var day=new Date().getDay(); switch (day) { case 0:   x="Today it's Sunday";   break; case 1:   x="Today it's Monday";   break; case 2:   x="Today it's Tuesday";   break; case 3:   x="Today it's Wednesday";   break; case 4:   x="Today it's Thursday";   break; case 5:   x="Today it's Friday";   break; case 6:   x="Today it's Saturday";   break; }** |

* ?

(Condition)?Instructions si condition réalisée : Instructions si condition non réalisée

|  |
| --- |
| **var now = new Date();**  **var greeting = "Good" + ((now.getHours() > 17) ? " evening." : " day.");**  **alert (greeting);** |

C’est équivanlent à écrire :

|  |
| --- |
| **Var now = new Date();**  **var greeting = "Good";**  **if (now.getHours() > 17)**  **greeting += " evening.";**  **else**  **greeting += " day.";** |

***Structures répétitives***

* for
  + for (Initialisation de la boucle, Condition de répétition, Variation des compteurs de la boucle)

{Instructions}

* + for (variable in tableau)

{Instructions}

* + for (variable in objet)

{Instructions}

* while

while (condition)

{Instructions}

* do

{Instructions}

while (condition)

***fonctions***

* Déclaration

function nom\_Fonction(argument1, argument2…)

{

instructions

[return valeur]

}

* Appel

L'appel se fait :

A partir d'une autre fonction : NomFonction(…)

ou

Suite à un événement : <NomBalise …..onEvénement="NomFonction(….)">

ou

A partir d'un lien : <a href="JavaScript:NomFonction(….)" …>…</a>

Remarque : Pour empêcher que le script dans href ne remplace le document courant, on applique l'opérateur void, qui neutralise toute valeur ou tout effet de retour : <a href="JavaScript:void(NomFonction(….))">…</a>

ou

Si la fonction retourne une valeur, à partir d'une variable : Variable=NomFonction(….)

Remarque :

* Les variables déclarées au niveau du Script sont considérées comme globales.
* Les variables déclarées dans des fonctions sont considérées comme locales si l'instruction **var** est utilisée dans leur déclaration (déclaration explicite) sinon elles sont considérées comme globales.

Exemples :

|  |  |
| --- | --- |
| **<!DOCTYPE html> <html> <head> <script> function myFunction() { alert("Hello World!"); } </script> </head>  <body> <button onclick="myFunction()">Try it</button> </body> </html>** | <button onclick="myFunction('Harry Potter','Wizard')">Try it</button>  <script> function myFunction(name,job) { alert("Welcome " + name + ", the " + job); } </script> |
| <button onclick="javascript:var k=myFunction(); alert(k);">Try it</button>  <script>  function myFunction()  {  var x=5;  return x;  }  </script> |  |

***Manipulation des chaînes de caractères(objet String)***

Une chaîne de caractères est composée d'une suite de caractères quelconques, y compris des balises HTML. Des caractères spéciaux peuvent aussi être insérés dans les chaînes :\n (nouvelle ligne), \r (Entrée), \t (tabulation), \f (saut de page), \b (retour arrière), \' pour une apostrophe…

Pour concaténer deux chaînes de caractères on utilise le symbole +

* Quelques propriétés
  + length : Retourne la longueur d'une chaîne de caractère (chaîne.length)
* Quelques méthodes
  + toLowerCase() : Convertit une chaîne de caractères en minuscules(chaîne.toLowerCase())
  + toUpperCase() : Convertit une chaîne de caractères en majuscules(chaîne.toUpperCase())
  + charAt(p) : Retourne le caractère à la position p (chaîne.charAt(p)
  + indexOf(Ch) : Retourne la position de ch dans une chaîne de caractères (Chaine1.indexOf(Chaîne2))
  + indexOf(Ch, p ) : Retourne la position de ch dans une chaîne de caractères en commençant la recherche par la position p (Chaine1.indexOf(Chaîne2, p))
  + lastIndexOf(Ch) : Retourne la position de ch dans une chaîne de caractères en commençant la recherche par la fin(Chaine1.lastIndexOf(Chaîne2))
  + lastIndexOf(Ch, p ) : Retourne la position de ch dans une chaîne de caractères en commençant la recherche par la fin et en s'arrêtant à la position p (Chaine1.lastIndexOf(Chaîne2, p))
  + substring(p1,p2) : Retourne une sous-chaîne allant de la position p1 à la position p2-1 (chaîne.substring(p1,p2))
  + split(séparateur) : Fractionne une chaîne de caractères en se basant sur le séparateur et retourne les différentes parties dans un tableau de chaînes de caractères (Tabelau=Chaîne.split(séparateur))
  + isNaN(élément) : Revoie true si élément est un nombre et false sinon (variable=isNaN(élément))

…

***Utilisation des fonctions date et heure (objet Date)***

* Pour manipuler la date en cours

Variable =new Date()

Remarque : La date qui sera stockée dans variable est sous la forme suivante :

"NomJour Mois jourMois Heure:Minutes:Secondes Année"

* Pour manipuler une autre date

Variable =new Date(date)

date peut être donnée sous les formes :

* + "année, mois, jour"
  + "année, mois, jour, heures, minutes , secondes"
  + "NomMois JourMois , Année Heures:Minutes:Secondes"
* Méthodes
  + getYear() : Retourne l'année d'une date (>1900) (UneDate.getYear())
  + setYear(année) : Modifie l'année d'une date (UneDate.setYear(année))
  + getMonth : Retourne le mois (entre 0 et 11) d'une date (UneDate.getMonth())
  + setMonth(mois) : Modifie le mois d'une date (UneDate.setMonth(mois))
  + getDate() : Retourne le numéro du jour du mois (entre 1 et 31) d'une date (UneDate.getDate())
  + setDate(jourMois) : Modifie le jour du mois d'une date (UneDate.setDate(jourMois))
  + getDay() : Retourne le jour de la semaine (entre 0 et 6) d'une date (UneDate.getDay())
  + setDay(jourSemaine) : Modifie le jour de semaine d'une date (UneDate.setDay(jourSemaine))
  + getHours() : Retourne l'heure (entre 0 et 23) d'une date (UneDate.getHours())
  + setHours(heure) : Modifie l'heure d'une date (UneDate.setHours(heure))
  + getMinutes() : Retourne les minutes( entre 0 et 59) d'une date (UneDate.getMinutes())
  + setMinutes(minutes) : Modifie les minutes d'une date (UneDate.setMinutes(minutes))
  + getSeconds() : Retourne les secondes (entre 0 et 59) d'une date (UneDate.getSeconds())
  + setSeconds(secondes) : Modifie les secondes d'une date (UneDate.setSeconds(secondes))
  + getTime() : Retourne le nombre de Millisecondes écoulées depuis le 1 janvier 1970 00:00:00 (UneDate.getTime())
  + setTime (nombreMillisecondes) : Modifie le nombre de Millisecondes écoulées depuis le 1 janvier 1970 00:00:00 (UneDate.setTime(nombreMillisecondes))

…

***Utilisation des fonctions mathématiques (objet Math)***

* Quelques propriétés
  + PI : constante PI (Math.PI)
* Quelques méthodes
  + abs(nombre) : Retourne la valeur absolue d'un nombre (Math.abs(nombre))
  + round(nombre) : Arrondit le nombre à l'entier le plus proche (Math.round(nombre))
  + max(nombre1,nombre2) : Renvoie le plus grand des deux nombres (Math.max(nombre1, nombre2))
  + min(nombre1,nombre2) : Renvoie le plus petit des deux nombres (Math.min(nombre1, nombre2))
  + pow(nombre1,nombre2) : Renvoie le résultat de nombre1 puissance nombre2 (Math.pow(nombre1, nombre2))
  + sqrt(nombre) : Retourne la racine carrée de nombre (Math.sqrt(nombre))
  + parseInt(chaîne) : Evalue une chaîne de caractère en un nombre entier (parseInt(chaîne))
  + parseFloat(chaîne) : Evalue une chaîne de caractère en un nombre à virgule flottante (parseFloat(Chaîne))
  + eval(chaîne) : Evalue une chaîne de caractère en un nombre (eval(Chaîne))
  + random() : Retourne un nombre aléatoire entre 0 et 1 (1 non compris) (variable=Math.random())

…

Les boîtes de dialogues :

* Les alertes :
* Alert(‘Votre messages’)
* Alert(‘‘Votre messages’’)
* Message de confirmation

**var r=confirm("Press a button");  
if (r==true)  
  {  
  x="You pressed OK!";  
  }  
else  
  {  
  x="You pressed Cancel!";  
  }**

* Boites de dialogue de saisie

|  |
| --- |
| [**var**](http://www.toutjavascript.com/reference/reference.php?ref=var)**saisie =**[**prompt**](http://www.toutjavascript.com/reference/reference.php?ref=prompt)**("Saisissez votre texte :", "Texte par défaut")** |

**Var name=prompt("Please enter your name","Harry Potter");  
if (name!=null && name!="")  
  {  
  x="Hello " + name + "! How are you today?";  
  }**