Leçon n°2 : Le Langage JavaScript

Objectifs

- Créer des pages Web Statiques en utilisant le langage HTML ;
- Programmer et intégrer des scripts dans une page Web ;

I. Introduction

❖ *Activité :*

Comparer les deux fichiers « login1.html » et « login2.html »

❖ *Réponse :*

……
……
……
……..

❖ *Constatation :*

……
……..

1. Les limites du langage HTML :

Le langage HTML présente plusieurs limites :

- Absence de structures de contrôles algorithmiques (conditionnelles et itératives) ;
- Un langage sans aucune logique de programmation procédurale

Pour passer ces limites deux solutions sont proposées :

- Utilisation d'un langage du côté client (JavaScript, VbScript, …) permettant d'ajouter d'autres fonctionnalités omises par HTML et peuvent être exécutés par le navigateur ;
- Utilisation des langages du côté serveur (ASP, PHP, …) : offrent la possibilité de se connecter à des bases de données et de verrouiller le code source. Le test de ces langages se fait sur un serveur d'hébergement (Apache, Netscape server, …).

2. Le Langage JavaScript :

C'est un langage de script incorporé (introduit, inséré) aux balises HTML permettant d'améliorer la présentation et l'interactivité des pages web.

II. Le formalisme de base du JavaScript

❖ *Activité*

```
<html>
<head> <title>menu</title>
<script language="javascript">
alert("bienvenue dans le langage Javascript");
</script></head>
<body >
<script language="javascript">
document.write("<b><font color='#aa1288'>bienvenue dans le langage Javascript
</font></b>");
</script>
</body>
</html>
```

- *Enregistrer le code html avec le nom « tp1.html » dans votre dossier de travail*
- *Exécuter le code*

❖ *Constatation :*

……
……
……
……
……
……..

III. Les objets JavaScript

1. Les objets JavaScript et leurs hiérarchies

❖ *Activité :*

Ouvrir le fichier « **objet.html** » et déduire La hiérarchie des objets d'interfaces.

❖ *Réponse :*

……
……

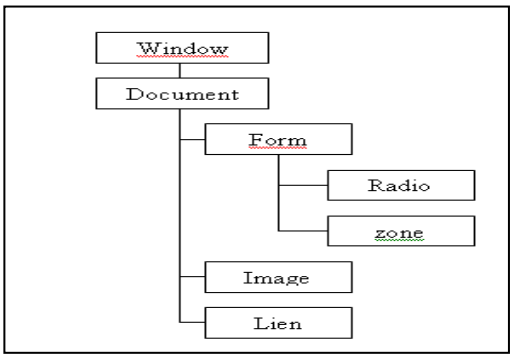
❖ *Constatation :*

JavaScript divise une page web en objets et permet d'accéder à ces objets pour tirer des informations et de les manipuler. On distingue deux catégories d'objets :

- Les objets d'interface :
  ……
  ……..
- Les objets des propriétés et des fonctions prédéfinies :
  ……
  ……...

a) La hiérarchie des objets d'interfaces

Exemple :



b) Les propriétés des objets

Pour accéder à une propriété, il faut donner le chemin complet de l'objet. En JavaScript on utilise la syntaxe : ……

Exemple :

✓ ……
✓ ……

c) Les méthodes des objets

Le langage JavaScript a prévu un ensemble de méthodes (fonctions) pour chaque objet. Par exemple pour l'objet Document on dispose de la méthode write () « Ecrire dans le document ».

❖ *Activité* : Créer un fichier nommé « TPJS2.html » et saisir le code suivant :

```
<html><head><title>methodes</title></head>
<body>
<script language="javascript">
document.write("methode write <br>");
x=2;
document.write(x);
document.write("<br>le contenu de la variable x est: "+x);
document.write("<br><b>le contenu de la bariable x est :</b>");
document.write("<font color='red'>"+x+"</font>");
</script>
</body></html>
```

❖ *Constatation :*

……
……
……
……
……

2. Les différents emplacements du code JavaScript

❖ *Activité* : Définir les différents emplacements du code JavaScript dans un document HTML. Créer un fichier nommé « TPJS3.html » et saisir le code suivant :

```
<html><head><title>emp</title>
<script language="javascript">
function fermer()
{
 window.close();
}
</script></head><body>
<script language="javascript">
function message()
{   document.write("pour fermer cette fenêtre vous pouvez cliquer: ");}
</script>
<script language="javascript">
message();
</script>
<a href ="" onclick="fermer()">ici</a>
</body></html>
```

❖ *Constatations* :

- Les solutions permettant d'insérer du code JavaScript dans une page web sont :

……
……

- La déclaration de fonctions JavaScript peut se faire :

……
……

IV. Les variables

❖ *Activité* :

```
<html><head>
<script language="javascript">
// déclaration des variables globales
var nom = "ali";
notec=12;
notesy=15
function moy()
{ // déclaration des variables locales
 var prenom="ben saleh ";
 moyenne=(notec+(notesy*2))/3;
}
</script></head>
<body>
<script language="javascript">
// la visibilité des variables globales
 document.write(" les notes de " +  nom +  ":" + notec + ","+ notesy );

document.write("et la moyenne = "+ moyenne  );
</script></body></html>
```

1- Enregistrer le fichier sous le nom « variables.html » puis tester le.
2- Modifier le code pour afficher le contenu de la variable moyenne.
3- Déduire les différentes méthode du déclaration d'une variable en javascript .

❖ *Réponse :*

…….

❖ *Constatations* :

- La déclaration des variables :

……
……
……
……
……
……..

- La visibilité des variables

……
……
……
……
……
……..

- Les types utilisé en javascript :

Type	Description
Numérique	
Chaîne de caractère	
booléens	
Le mot Null	

✓ Remarque :

Il ne faut pas déclarer le type de données d'une variable.

❖ *Activité :* écrire un script javascript qui permet de lire deux valeur a et b puis de calculer et d'afficher la somme.

❖ *Réponse :*

……
……
……
……
……
……
……
……
……
……

❖ *Constatations* :

✓ Les méthodes de lectures et d'affichages des données

……
……
……
……

✓ Les fonctions de la conversion des types :

- La fonction Number(chaîne numérique) :

……..

Exemple : var a = Number("12")→ a=……..

var x = Number("12a")→ x=……..

- La fonction String(valeur numérique) :

……

Exemple : var a = String(12.36)→ a=……..

- La fonction isNaN(valeur) :

……

……..

Exemple : var a = isNaN("12")→ a=…….. var a = isNaN("12a")→ a=……..

✓ Les méthodes de l'objet Math :

Nom	syntaxe	rôle	Exemple
abs()	Math.abs(valeur)		
round()	Math.round(valeur)		
random	Math.random ()		
eval	eval (expression)		

NB : voir livre page 168

✓ Les méthodes de l'objets String :

Nom	syntaxe	rôle	Exemple
length	Chaine.length		
charAt	Caractère=chaine.charAt(valeur)		
indexOf	P=chaine.indexOf(ch2,pos)		
substr	Sch=chaine.substr(pos,nbc)		
toUpperCase	Ch=chaine.toUpperCase()		

NB : voir livre page 162

V. Les opérateurs prédéfinis :

1. Les opérateurs de calcul

Signe	Signification
+	Addition
-	Soustraction
*	Multiplication
/	Division
%	Modulo (reste de la division)
=	Affectation

2. Les opérateurs de comparaison

Signe	Signification
==	Egal
<	Inférieur
<=	Inférieur ou égale
>	Supérieur
>=	Supérieur ou égal
!=	Différent

3. Les opérateurs logiques

Signe	Nom	Exemple
&&	ET	(Condition1) && (Condition2)
||	OU	(Condition1) || (Condition2)
!	NON	! (Condition)

4. Les opérateurs d'incrémentation

Ces opérateurs vont augmenter ou diminuer la valeur d'une variable d'une unité.
Dans notre exemple x vaut initialement 3.

Signe	Signification	Exemple	Résultat
x++, ++x	Incrémentation (x++ est le même que x=x+1)	Y=x++ ; Z=++x ;	Y=3 et x=4 Y=4 et x=4
x--, --x	Décrémentation (x-- est le même que x=x-1)	Y=x-- ; Y=--x ;	Y=3 et x=2 Y=2 et x=2

VI. Les Entrées/Sorties en JavaScript :

- *Activité :* créer un fichier html nommée « permutation.html » permettant l'échange du contenu de deux variables a et b.
- *Réponse :*

……
……
……
……
……
……
……
……
……
……

- *Constatations* :

  - L'entrée (lecture) :
    ……
    ……
  - La sortie :
    ……
    ……
  - L'affectation : …….

Syntaxe :
L'entrée : ……..
La sortie : ……
……

VII. Les Structures de contrôle :

1. Les structures conditionnelles

a) La structure IF

La Forme réduite	La Forme complète

if (condition vraie)
{Une ou plusieurs instructions ;}
	if(condition vraie)
	{Instructions 1 ;}
	else
	{Instructions 2 ;}

❖ *Activité 1 :* créer un fichier HTML nommée « comparaison.html » qui détermine le maximum de deux entiers (forme réduite).

❖ *Réponse :*

……
……
……
……
……
……
……
……
……
……

❖ *Activité 2 :* créer un fichier HTML nommée « parité.html » qui permet d'afficher la parité d'un nombre

❖ *Réponse :*

……
……
……
……
……
……
……
……
……
……

b) La forme généralisé :

syntaxe :

```
 if(condition1 )
      {Instructions 1 ;}
else if (condition2 )
       {Instructions 2 ;}
else if (condition3 )
       {Instructions 3 ;}
      …..
else
      {Instructions n ;}
```

❖ *Activité :* créer un fichier HTML nommée « temps .html » qui permet de lire une période T (hh :mm :ss) puis de vérifier s'elle valide ou non valide .

❖ *Réponse :*

……
……

……
……
……
……
……
……
……
……
……
……

Remarques:

- Dans le cas d'une seule instruction, les accolades sont facultatives;
- Il est possible d'imbriquer des structures conditionnelles (forme généralisée) ;

c) La structure switch

❖ Syntaxe :

```
switch (expression)
{
Case v1: bloc 1;
                Break;
Case v2: bloc 2;
                 Break ;
   …
default: bloc n;
Break ;
```

❖ *Activité :* Ecrire un code javascript permettant de lire le numéro du mois et la valeur d'une année puis de déterminer le nombre de jours de ce mois.

❖ *Réponse :*

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

- Remarques:

……
……
……
……
……
……
……
……

2. Les structures itératives

a) La structure For

Permet de répéter l'exécution d'un bloc d'instructions un certain nombre de fois connu d'avance :

- Syntaxe :

```
for (initialisation ; condition ; progression)
{
 Instructions ;
}
```

  - ✓ Initialisation : les instructions d'initialisations nécessaires ;
  - ✓ Condition : la condition de continuité (bouclage) ;
  - ✓ Progression : définit le pas du compteur

- *Activité :* Ecrire un code JavaScript permettant de lire entier puis d'afficher ces diviseurs dans un tableau html.
- *Réponse :*

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

b) La structure do…while

- Syntaxe :

```
do
{
Instructions;
} while (condition(s));
```

- *Activité :* Ecrire un code JavaScript permettant de lire un nombre (10 <= n<=100).
- *Réponse :*

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

- Remarques:

……
……
……
……

c) La structure while

Lorsque le nombre de répétitions n'est pas connu d'avance on utilise la structure itérative while.

- Syntaxe :

```
while (condition(s))
{
 Instructions ;
}
```

- *Activité* : Ecrire un code JavaScript permettant d'afficher le PGCD de deux entiers donnés a et b tels que (a>1 et b>1) en utilisant la méthode de différences.
- *Réponse :*

……
……
……
……
……
……

……
……
……
……
……
……
……
……

VIII. Les Fonctions en JavaScript :

En JavaScript, il existe deux types de fonctions :

- Les fonctions prédéfinies, on les appelle « méthodes ». Elles sont associées à un objet particulier comme par exemple la méthode Alert () de l'objet window.
- Les fonctions déclarées par le programmeur selon les besoins de l'application.

Pour déclarer une fonction, on utilise la syntaxe suivante :

```
function nom_de_la_fonction(arguments)
{
  Instructions ;
  [return nom_variable]
}
```

❖ *Activité :* modifier l'exemple précédent en utilisant une fonction qui retourne le PGCD de deux entiers.

❖ *Réponse :*

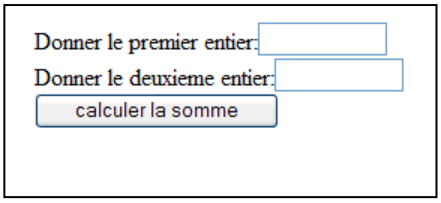
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

Remarques :

……
……
……
……
……
……

IX. La gestion des évènements en Javascript :

❖ *Activité* 1: Créer un fichier nommé « evenement1.html » composé de deux zones saisi et Ecrire un code Javascript permettant de calculer la somme de deux entiers saisis dans les zones de texte en cliquant sur un lien hypertexte ou un bouton.



❖ *Réponse :*

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

❖ *Constatations* :

- OnClick : peut être utilisé avec les boutons (type = "button") et les liens hypertextes.

OnClick = " Nom_fonction() "

- Exemple :……..
- OnSubmit="return Nom_fonction()" : utilisé avec les boutons (type = "submit") dans la balise <form OnSubmit="return Nom_fonction">
- Exemple :…….

Activité 2:

```
<html><head><script language="javascript">
function msg()
{alert("saisir une chaine");}
function nbc()
{var x = document.f1.ch1.value;
document.f1.nbcar.value=x.length;
}
function majus()
{var x = document.f1.ch1.value;
document.f1.maj.value=x.toUpperCase();
}
</script></head><body>
<form  name ="f1" method ="get">
   Chaine : <input name="ch1" type="text" size="20" OnChange="majus()" OnKeyUp="nbc()" OnFocus="msg()">
          nombre de caractere : <input name="nbcar" type="text" size="20" maxlength="30">
           Chaine en majuscule : <input name="maj" type="text" size="20" maxlength="30">
```

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

1- Enregistrer le fichier sous le nom « test14.html »
2- Déduire les différents événements utilisés en javascript.

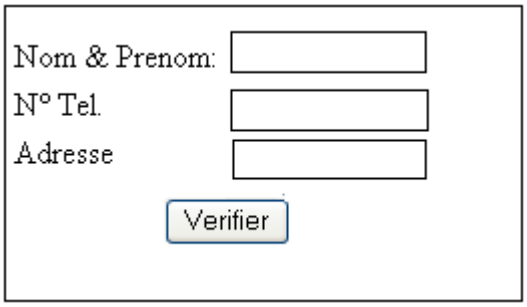
Les évènements :

- OnFocus : survient lorsqu'un champ de saisi est prêt à recevoir ce que l'utilisateur à l'intention de taper au clavier (c à d quand la zone a le focus) ;
- OnChange : ressemble à l'évènement OnBlur avec une différence (la zone de texte a perdu le focus et son contenu est modifié par l'utilisateur) ;
- OnKeyUp : est déclanché au cours de la saisi d'une zone de texte lorsque l'utilisateur relâche une touche.

X. Les formulaires en Javascript :

1. Le contrôle zone de texte

Activité : Soit le formulaire suivant :



1- ouvrir le fichier « saisie .html »
2- Ajouter une fonction javascript qui permet de :
   - ✓ Vérifier le contenu la zone Nom et prénom n'est pas vide
   - ✓ Vérifier le numéro de téléphone (8 chiffre numérique)
   - ✓ Vérifier si l'adresse email contient le caractère @

❖ *Réponse :*

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

❖ *Constatations* :

✓ Pour affecter le contenu d'une zone de saisie à une variable :

……

✓ Pour modifier la valeur d'une zone de saisie :

……

2. Les boutons radio

❖ *Activité* : A l'aide de logiciel Notepad , ouvrir le fichier « radio.html » puis ajouter un script qui détermine le nombre de solution d'une équation de second degré pour cas.

❖ *Réponse :*

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

❖ *Constatations* :

- ✓ Pour vérifier l'état d'une case d'option (cochée ou pas), on utilise la propriété « checked »qu'on peut affecter à une variable booléenne :

……..

- ✓ On peut récupérer la valeur d'un bouton radio à l'aide de la propriété « value »

……..

- ✓ On peut récupérer le nombre d'option dans un groupe de cases d'options à l'aide de la propriété « length » : …….
- ✓ L'indice des boutons radio commence à partir de la valeur 0.

3. Les boutons case à cocher (checkbox)

<u>*Activité*</u> : A l'aide de logiciel Notepad , ouvrir le fichier « case.html » puis ajouter un script qui détermine les nombres premiers parmi la liste des nombres suivantes : 11 ; 24 ; 37 .

❖ *Réponse :*

……
……
……
……
……
……
……
……
……
……
……
……
……
……

❖ *Constatations* :

- ✓ Pour vérifier si une case est cochée ou non on utilise la syntaxe suivante :

……

- ✓ Pour récupérer la valeur d'une case à cocher on utilise la syntaxe suivante :

……

4. Liste de sélection

- Activité : ouvrir le fichier « eleve.html » puis ajouter les fonctions suivantes :
  - ✓ Ajouter() :permet d'ajouter les coordonner de l'élevé à la liste.
  - ✓ Modifier cet fonction pour éviter la redondance des donnes da la liste
  - ✓ Afficher () : permet d'afficher les coordonnées de l'élevé sélectionner a partir de la liste dans les zones de textes correspondantes
  - ✓ Supp() : permet de supprimer un élevé de la liste.
- *Réponse :*

……
……
……

❖ REMARQUES

✓ Modification de la fonction ajouter (contrôle sur l'ajout)

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

✓ L'utilisation des types du bouton

- Le type button :

Evenement utilise : OnClick="nom_fonction()"

Utilisation : <input type="button" name="b1" OnClick="nom_fonction()">

- Le type submit :

Evenement utilise : Onsubmit="return nom_fonction()"

<Form name=" f1" action="nom_page.html" methode="post/get" Onsubmit="return nom_fonction()">

Type de fonction : booleen

exemple:

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

X. Application :

Créer un fichier HTML qui permet de remplir un formulaire par le nom, le prénom, l'adresse Email, l'age et les deux boutons de confirmation et d'annulation. Le contrôle de saisie est obligatoire :

- Le nom et le prénom doivent être non vides ;
- L'adresse Email doit être non vide et comportant le caractère @ ;
- L'age doit être un nombre >0.

La validation doit appeler le fichier « sauvegarde.html » (existant dans le même répertoire du formulaire) qui affiche le message « merci pour votre visite ».

```
<html>
<head>
<title>saisie formulaire</title>
<script language="javascript">
function verification()
{
if(document.f1.nom.value=="")
{
alert("veuillez entrer votre nom SVP");
return false;
}
if(document.f1.pre.value=="")
{
alert("veuillez entrer votre prénom SVP");
return false;
}
if(document.f1.mail.value=="")
{
alert("veuillez entrer votre adresse Email SVP");
return false;
}
if(document.f1.mail.value.indexOf("@")==-1)
{
alert("ce n'est pas une adresse électronique");
document.f1.mail.value="";
return false;
}
n=document.f1.age.value;
```

```
if(isNaN(n)||n=="")
{
alert("cette information n'est pas un nombre");
return false
}
}
</script>
</head>
<body>
<center><h1>Remplissage du formulaire</h1></center>
<form name="f1" action="sauvegarde.html" method="post" OnSubmit="return verification()">
Nom:<input type="text" name="nom" size="40"><br>
Prénom:<input type="text" name="pre" size="40"><br>
Email:<input type="text" name="mail" size="40"><br>
Age:<input type="text" name="age" size="3"><br>
<center><input type ="submit" value="valider"><input type="reset" value="annuler"></center>
</form>
</body>
</html>
```

Remarque : La méthode indexOf() : permet de rechercher (de gauche à droite) la première position d'une sous chaîne dans une chaîne de caractères donnée si elle existe ou -1 dans le cas contraire.
P=ch.indexOf(sch,pos) ;
La position pos permet de déterminer la position du caractère à partir duquel la recherche est effectuée.