

Chapitre 2

Production électronique avancée

Partie A: Pages Web Statique

Le langage JavaScript

I. Introduction

Limites du langage HTML

- ✓ Absence des structures de contrôles (itératives et conditionnelles)
- ✓ Absence de prise en charge d'événements.
- ✓ Pas de communication avec le système utilisé

Solution proposé : Langage JavaScript

- Le JavaScript est un langage de script incorporé dans un document HTML.
- Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML.

II- Structure d'une page Web Contenant un code JavaScript

```
<HTML>
<HEAD>
<TITLE> titre </TITLE>
<script language="javascript">
  script javascript
</script>
</HEAD>
<BODY>
<script language="javascript">
  script javascript
</BODY>
</HTML>
```

III- Les instructions d'affichage de JavaScript

Activité 1:

Afficher les phrases suivantes : «4SI» et «Bac 2014» dans une page Web en utilisant un code JavaScript et la commande.

```
<html>
<head>
<script language="javascript">
//ceci un commentaire
document.write('4SI Bac 2014');
alert ('4SI Bac 2014');
</script>
</head> <body> </body> </html>
```

Constatation:

- ✓ //: pour insérer les commentaires (facultatif)
- ✓ **document.write ()** : c'est une méthode qui permet d'afficher un texte dans une page Web. Cette méthode appartient à l'objet document.
- ✓ **alert ()** : permet d'afficher le texte dans une fenêtre message

IV- Les instructions de saisie dans JavaScript**Activité 2**

Saisir dans une boîte de saisie votre « date de naissance » puis l'afficher dans une boîte de dialogue en utilisant un code JavaScript.

```
<html>
<head>
</head>
<body>
<script language="JavaScript">
dn=prompt('Donner votre date de naissance');
alert('La date de naissance saisie='+dn);
</script>
</body>
</html>
```

Constatation:

- ❑ **Pour entrer (saisir) des données, on utilise:**
 - Soit la fonction **prompt()** : Affiche une boîte de dialogue avec un message et un champ de saisie `Nom_variable=prompt(Message, [valeurDéfaut]);`
 - Soit à l'aide des objets graphiques du formulaire HTML
- ❑ **pour afficher des données, on utilise**
 - Soit la fonction `alert("message texte"+ nom_variable)`
 - Soit la fonction `write: document.write ("message texte"+ nom_variable)`
 - + est un operateur de concaténation

V- Les structures de données dans JavaScript**1- Les variables**

Une variable est un objet repéré par son nom, pouvant contenir des données, qui pourront être modifiées lors de l'exécution du programme.

Une variable peut être déclarée de deux façons :

Explicitement : `var x = 3;`

Implicitement : `x = 3;`

2- Types de données

En JavaScript, on peut utiliser 4 types de données :

- Les nombres : entiers et réels
- Les chaînes de caractères

- Les booléens : true ou false
- Le mot null : indique l'absence d'une valeur

Activité 3

Créer un fichier nommé « tpjs3.html » et saisir le code suivant

```
<html>
<body>
<script language="javascript">
b=prompt ("saisissez une valeur");
if(isNaN(b))
{
  alert("conversion impossible");
}
else
{
  b=Number (b);
  b=b+1;
  document.write("la valeur de b est :"+b);
}
</script>
</body>
</html>
```

Remarques :

- ✓ La Fonction « **isNaN** : is Not a Number » est une fonction booléenne permettant de vérifier si le contenu d'une variable donnée est numérique ou non.
- ✓ Il existe des fonctions de conversion de type : String et Number
 Var a = String (21.34) ; → a = "21.34"
 Var a = Number ("10.5") ; → a = 10.5
- ✓ La fonction « eval » évalue une chaîne de caractères sous forme de valeur numérique :
 Exemple : ch="5+10" ; x=eval (ch) ; → x=15 ;

3- Les opérateurs prédéfinis

a- Les opérateurs de calcul

Signe	Signification
+	Addition
-	Soustraction
*	Multiplication
/	Division

%	Modulo (reste de la division)
=	Affectation

b- Les opérateurs de comparaison

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

c- Les opérateurs associatifs

Sont des opérateurs qui réalisent un calcul dans lequel une variable intervient des deux cotés du signe (dans notre exemple $x = 11$ et $y = 5$)

Signe	Nom	Signification	Exemple	Résultat
+=	Plus égal	$x = x+y$	$x +=y$	16
-=	Moins égal	$x = x-y$	$x -=y$	6
*=	Multiplié égal	$x = x*y$	$x *=y$	55
/=	Divisé égal	$x = x/y$	$x /=y$	2.2

d- Les opérateurs logiques

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

e- Les opérateurs d'incrémentatation

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	Incrémentatation (x++ est le même que x=x+1)	Y=x++ ; Z=++x ;	Y=3 et x=4 Y=4 et x=4
x--, --x	Décrémentatation (x-- est le même que x=x-1)	Y=x-- ; Y=--x ;	Y=3 et x=2 Y=2 et x=2

VI- 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 4

Déterminer le maximum de deux entiers (forme réduite)

```
<html>
<head>
<title>forme reduite</title>
</head>
<body>
<script language="javascript">
var a=Number(prompt("donner un premier entier:", ""));
var b=Number(prompt("donner un deuxieme entier:", ""));
max=a;
if (a<b)
{max=b;}
alert("le maximum est:"+max);
</script> </body></html>
```

Activité 5

Déterminer le maximum de deux entiers (forme généralisée)

```

<html>
<head>
<title>forme reduite</title>
</head>
<body>
<script language="javascript">
var a=Number(prompt("donner un premier entier:", ""));
var b=Number(prompt("donner un deuxieme entier:", ""));
if (a<b)
{max=b;}
else
{max=a;}
alert("le maximum est:"+max);
</script>
</body>
</html>

```

Remarques:

Dans le cas d'une seule instruction, les accolades sont facultatives;

b) La structure switch

La syntaxe est la suivante :

```

Switch (expression)
{
Case v1: bloc 1;Break;
Case v2: bloc 2;Break ;
...
Default: bloc n;Break ;

```

Activité 6

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.

```

<script language="javascript">
var m=Number(prompt("donner le numéro du mois", ""));
var a=Number(prompt("donner la valeur de l'année", ""));
switch(m)
{
case 1:case 3:case 5:case 7:case 8:case 10:case 12:j=31;break;
case 4:case 6:case 9:case 11:j=30;break;
case 2:if(a%4==0)j=29; else j=28;break;}
alert("le nombre de jours du mois n°"+m+" est: "+j);
</script>

```

Remarques:

L'instruction break permet de quitter la structure switch après l'exécution du bloc convenable ;

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 :

For (initialisation ; condition ; progression)

```
{
  Instructions ;
}
```

- ✓ Initialisation : les instructions d'initialisations nécessaires ;
- ✓ Condition : la condition de continuité (bouclage) ;
- ✓ Progression : définit le pas du compteur (pas forcément + ou - 1)

Activité 7

Ecrire un code JavaScript permettant d'afficher tous les nombres parfaits compris entre 2 et 1000 ; sachant qu'un nombre N est dit parfait s'il est égal à la somme de ses diviseurs sauf lui même. Exemple pour N=6 → 1+2+3=6

```
<script language="javascript">
for (n=2;n<=1000;n++)
{
  s=0;
  for(i=1;i<=n/2;i++)
    if(n%i==0) s+=i;
  if (s==n) document.write(n+"<br>");
}
</script>
```

b) La structure do...while

Permet de répéter l'exécution des instructions tant que la condition est vérifiée. L'équivalent en langage pascal est repeat...until (à la seule différence que la condition utilisée après until est une condition d'arrêt alors que la condition utilisée après while est une condition de continuité)

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

Activité 8

Ecrire un code JavaScript permettant d'afficher tous les nombres premiers compris entre deux entiers ($a > 1$ et $a < b$ et $b < 1001$). Un nombre est dit premier s'il n'est divisible que par 1 et par lui-même.

```
<script language="javascript">
do
{
a=Number(window.prompt("donner un entier: ",""));
b=Number(window.prompt("donner un autre entier: ",""));
}while(a<=1||a>=b||b>=1001);
for (n=a;n<=b;n++)
{i=1;
do
{i=i+1;
} while (n%i!=0 && i<=n/2);
if (i>n/2) document.writeln(n);}
</script>
```

c) La structure while

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

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

Activité 9

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.

```
<script language="javascript">
var a,b;
do
{
a=Number(prompt("donner un entier",""));
b=Number(prompt("donner un autre entier","")); }
while (a<=1||b<=1);
while (a!=b)
if (a>b)
a-=b;
else
b-=a;
alert("le pgcd est "+a);
</script>
```


VII. Les Fonctions en JavaScript :

Activité 10

Saisir le code JavaScript permettant de calculer le PGCD de deux entiers donnés m et n, tels que ($m > 1$ et $n > 1$)

```
<script language="JavaScript">
function pgcd(a,b)
{
  while(a!=b)
  {
    if(a>b)
      a-=b;
    else
      b-=a;
  }
  return a;
}
var n,m;
do
{
  n=Number(prompt("donner un entier",""));
  m=Number(prompt("donner un autre entier",""));
}while(n<=1||m<=1);
var k=pgcd(n,m);
document.write("le pgcd est: "+k);
</script>
```

Syntaxe

```
Function nom_fonction(arguments)
{
  Traitements ;
Return resultat;
}
```

Constatation

- La mention des arguments est facultative mais les parenthèses doivent rester.
- Dans une fonction une variable déclarée avec le mot Var sera considérée comme une variable locale, dans le cas contraire (sans le mot Var) sera considérée comme une variable globale.
- Il est possible de définir une fonction sans l'utilisation de la clause « return » : on retrouve ainsi l'équivalent d'une procédure

VIII. La gestion des événements en JavaScript

JavaScript offre la possibilité d'utiliser plusieurs événements, contrairement au HTML qui se contente de l'unique clic. La programmation des événements par l'association à chaque événement une action.

La syntaxe est la suivante :

```
<nom_balise OnEvenement = "fonction()">
```

Exemple: ici

Activité 11

Créer un fichier nommé « evenement1.html » composé de deux zones saisie 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.

```
<html>
<head>
<title>onclick</title>
<script language="JavaScript">
function somme()
{
var a=window.document.f1.e1.value;
var b=window.document.f1.e2.value;
a=Number(a);
b=Number(b);
s=a+b;
alert("la somme est: "+s);
}
</script>
</head>
<body>
<form name="f1">
Donner le premier entier:<input type="text" name="e1" size="10"><br>
Donner le deuxieme entier:<input type="text" name="e2" size="10"><br>
<input type="button" value="calculer la somme" name="b1"
OnClick="somme()"><br>
<a href="#" OnClick="somme()">calculer la somme</a>
</form> </body> </html>
```

Activité 12

Créer un fichier nommé « evenement2.html » composé de deux zones de saisie, et offrant les fonctionnalités suivantes :

Lorsque le curseur se trouve dans la première zone le message « Saisissez votre nom et prénom » est affiché dans la barre d'état ;

Lorsque le curseur quitte la première zone : le contenu de la barre d'état se transforme en *** et le texte saisi est mis en majuscule ;

Au fur et à mesure que le texte est saisi dans la première zone, la deuxième zone indique le nombre de caractères tapés.

```
<html>
<head>
<title>evenements</title>
<script language = "JavaScript">
function majuscule()
{
np=window.document.f1.nom.value;
npmaj=np.toUpperCase();
window.document.f1.nom.value=npmaj;
}
function affiche(msg)
{
window.status=msg;
}
function longueur()
{
l=window.document.f1.nom.value;
window.document.f1.nbc.value=l.length;
}
</script>
</head>
<body>
<form name="f1">
NOM ET PRENOM : <input type="text" name="nom" OnFocus="affiche('saisissez
votre nom et prénom')"
OnChange="majuscule()" OnBlur="affiche('***')" OnKeyUp = "longueur()">
Nombre de cacarctères tapés : <input type="text" name="nbc" size="3" value="0">
</form>
</body>
</html>
```

1- Les évènements :

Evénement (gestionnaire d'événement)	Description
Click (onClick)	Se produit lorsque l'utilisateur clique sur l'élément associé à l'événement
Load (onLoad)	Se produit lorsque le navigateur de l'utilisateur charge la page en cours
Unload (onUnload)	Se produit lorsque le navigateur de l'utilisateur quitte la page en cours
MouseOver (onMouseOver)	Se produit lorsque l'utilisateur positionne le curseur de la souris au-dessus d'un élément
MouseOut (onMouseOut)	Se produit lorsque le curseur de la souris quitte un élément
Focus (onFocus)	Se produit lorsque l'utilisateur donne le focus à un élément, c'est-à-dire que cet élément est sélectionné comme étant l'élément actif
Blur (onBlur)	Se produit lorsque l'élément perd le focus, c'est-à-dire que l'utilisateur clique hors de cet élément, celui-ci n'est alors plus sélectionné comme étant l'élément actif
Change (onChange)	Se produit lorsque l'utilisateur modifie le contenu d'un champ de données
Submit (onSubmit)	Se produit lorsque l'utilisateur clique sur le bouton de soumission d'un formulaire (le bouton qui permet d'envoyer le formulaire)

2- Objets auxquels on peut associer des évènements

Objet	Evénements associables
Lien hypertexte	onClick, onMouseOver, onMouseOut
Page du navigateur	onLoad, onUnload
Bouton, Case à cocher, Bouton radio, Bouton Reset	onClick
Liste de sélection d'un formulaire	onBlur, onChange, onFocus
Bouton Submit	onSubmit
Champ de texte et zone de texte	onBlur, onChange, onFocus, onSelect
Lien hypertexte	onClick, onMouseOver, onMouseOut
Page du navigateur	onLoad, onUnload
Bouton, Case à cocher, Bouton radio, Bouton Reset	onClick
Liste de sélection d'un formulaire	onBlur, onChange, onFocus
Bouton Submit	onSubmit
Champ de texte et zone de texte	onBlur, onChange, onFocus, onSelect

3- Les méthodes :

- **Length** : permet de retourner la longueur d'une chaîne ou 0 si elle est vide ;
- **toUpperCase()** : transforme une chaîne de caractères en majuscule ;
- **Status** : Valeur du texte affiché dans la barre d'état de la fenêtre.

Activité 13

Créer un fichier nommé « evenement2.html » composé d'une image (e1.jpg) et un objet texte « Exemple3 » avec les fonctionnalités suivantes :

Au chargement de la page un message de bienvenue est affiché, de même un message d'au revoir est affiché lorsqu'on la quitte ;

Deux messages sont affichés lorsqu'on survole et en quitte le lien hypertexte (exemple3) ;

Lorsque l'image e1.jpg est survolée, elle est remplacée par une autre image e2.jpg.

```

<html>
<head>
<title>evenement3</title>
<script language="javascript">
function bienvenue()
{
alert("soyez la bienvenue");
}
function aurevoir()
{
alert("merci au revoir");
}
function activer()
{
document.images["img1"].src="e2.jpg";
}
function desactiver()
{
document.images["img1"].src="e1.jpg";
}
</script>
</head>
<body OnLoad="bienvenue()" OnUnload="aurevoir()">
<a href="#" OnMouseOver="activer()" OnMouseOut="desactiver()">
</a><br>
<a href="#" OnMouseOver="alert('bonne chance')" OnMouseOut="alert('bon
travail')">Exemple 3</a>
</body>
</html>

```

IX. Les formulaires en Javascript :

1- Le contrôle zone de texte

Activité 14

Créer un nouveau document html comportant un script qui permet de lire un entier dans une zone de saisie puis d'afficher son carré dans une autre zone.

The image shows a screenshot of a web browser displaying a simple form. On the left, there is a text input field with the label "Donner un entier:" and the number "2" entered. In the center, there is a button with the text "calculez le carré". On the right, there is another text input field with the label "Le carré est:" and the number "4" entered. The entire form is enclosed in a rectangular border.

```
<html><head><title>carre</title>
<script language="JavaScript">
function calcul()
{
var n=document.f1.v1.value;
n=Number(n);
m=n*n;
document.f1.v2.value=m;
}
</script>
</head>
<body>
<form name="f1">
Donner un entier:<input type="text" name="v1" size="10" value="">
<input type="button" name="b1" value="calculez le carré" onClick="calcul()">
Le carré est:<input type="text" name="v2" size="10" value="">
</form></body></html>
```

Constatations

- Pour affecter le contenu d'une zone de saisie à une variable :
Nom_variable=window.document.nom_formulaire.nom_zone_de_saisie.value
- Pour modifier la valeur d'une zone de saisie :
Window.document.nom_formulaire.nom_zone_de_saisie.value=expression

2- Les boutons radio

Activité 15

Créer un fichier html qui comporte un script Javascript permettant de donner en entrée le signe du discriminant d'une équation de second degré pour afficher le nombre de solutions possibles.

Signe du discriminant d'une équation de second degré

Strictement négatif
 Nul
 Strictement positif

```

<html><head>
<title>radio</title>
<script language="JavaScript">
function choisir()
{
if(document.f1.choix[0].checked==true)
alert("il ya"+document.f1.choix[0].value+" solution");
if(document.f1.choix[1].checked==true)
alert("il ya"+document.f1.choix[1].value+" solution");
if(document.f1.choix[2].checked==true)
alert("il ya"+document.f1.choix[2].value+" solutions");
}
</script></head>
<body>
<center><h3>Signe du discriminant d'une equation de second
degre</h3></center>
<form name="f1">
<input type="radio" name="choix" value="0">Strictement negatif<br>
<input type="radio" name="choix" value="1">Nul<br>
<input type="radio" name="choix" value="2">Strictement positif<br>
<input type="button" name="b1" value="nombre de solutions"
OnClick="choisir()">
</form></body>
</html>

```

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 :

Nom_variable=document.nom_formulaire.nom_case[indice].checked

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

Nom_variable=document.nom_formulaire.nom_case[indice].value

- ❑ On peut récupérer le nombre d'option dans un groupe de cases d'options à l'aide de la propriété « length » :

Nom_variable=document.nom_formulaire.nom_case.length

- ❑ Les boutons radio sont utilisés pour sélectionner un seul choix, parmi un ensemble de propositions ;
- ❑ L'indice des boutons radio commence à partir de la valeur 0.

3- Les boutons case à cocher (checkbox)

Activité 16

Créer un fichier html qui comporte un script JavaScript permettant à un utilisateur de sélectionner les nombres premiers parmi la liste des nombres suivantes : 11 ; 24 ; 37 ; 49.

```
<html><head>
<title>Case a cocher</title>
<script language="JavaScript">
function reponse()
{
if((document.f1.ch1.checked==true)&&(document.f1.ch2.checked==false)
&&(document.f1.ch3.checked==true)&&(document.f1.ch4.checked==false))
alert("c'est la bonne réponse");
else
alert("vérifiez votre choix");
}
</script> </head>
<body>
<center><h3>Quels sont les nombres premiers parmi ces cinq
entiers?</h3></center>
<form name="f1">
<input type="checkbox" name="ch1" value="1">11<br>
<input type="checkbox" name="ch2" value="2">24<br>
<input type="checkbox" name="ch3" value="3">37<br>
<input type="checkbox" name="ch4" value="4">49<br>
<input type="button" name="b1" value="Reponse" OnClick="reponse()">
</form> </body> </html>
```


Constatations

- Les cases à cocher sont utilisées pour sélectionner un ou plusieurs choix ;
- Pour vérifier si une case est cochée ou non on utilise la syntaxe suivante :
Nom_variable=document.nom_formulaire.nom_case.checked
- Pour récupérer la valeur d'une case à cocher on utilise la syntaxe suivante :
Nom_variable=document.nom_formulaire.nom_case.value

4- Liste de sélection

- La gestion des listes déroulantes se fait à travers l'objet «options»
document.nom_formulaire.nom_liste.options
- Les propriétés de l'objet options sont :
 - ✓ **Length** : retourne le nombre d'éléments d'une liste déroulante ;
 - ✓ **selectedIndex** : retourne l'indice de l'élément sélectionné dans une liste déroulante à sélection unique. (L'indice d'une liste déroulante commence par 0)
- Pour une liste déroulante à sélection multiple, selectedIndex retourne l'indice du premier élément sélectionné pour cela on utilise les propriétés des éléments de l'objet «options» :
 - ✓ **selected** : renvoie true si l'option est sélectionnée, false sinon ;
 - ✓ **text** : renvoie le texte de l'élément sélectionné ;
 - ✓ **value** : renvoie la valeur de l'élément sélectionné.
- L'ajout d'une nouvelle option :
nom_variable=new Option(texte, valeur) ;
document.nom_formulaire.nom_liste.options[taille]=nom_variable ;
 - ❖ *La taille correspond à la position de l'élément à ajouter dans la liste*
- La suppression d'une option :
document.nom_formulaire.nom_liste.options[i]=null ;
 - ❖ *i correspond à la position de l'élément à supprimer.*

Application

Créer un fichier HTML qui permet de remplir un formulaire par le nom, le prénom, l'adresse Email, l'âge 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'âge 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 ».

Remplissage du formulaire

Nom:

Prénom:

Email:

Age:

```

<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) || (document.f1.mail.value==""))
{ alert("vérifier votre adresse électronique"); 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>

```