

Lecon 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 :*

Dans un formulaire ,on a besoin de contrôler les informations saisie par l'internaute donc on besoin d'utiliser un langage de programmation qui peut être implémenter dans le code html et qui doit être exécuter par les ordinateurs des clients.

❖ *Constatation :*

Le Langage JavaScript c'est un langage exécute au coté client qui permet de contrôler le contenu d'un formulaire en utilisant des structures algorithmiques.

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 2 page 79

Constatation :

- ✓ Le code javascript est délimiter par `<script language="javascript">...</script>`
- ✓ Le commentaire d'explication est délimité par `//` ou `/* ... */`
- ✓ Le langage javascript est sensible à la casse .

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 :*

Le langage javascript permet de manipuler les objets qui possède des noms (image,les objets de formulaire)

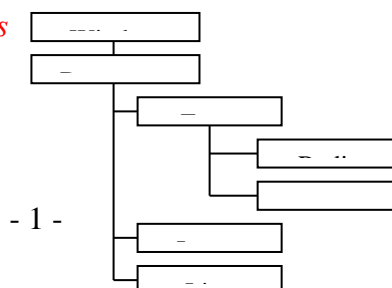
- *Constatation :*

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

- Les objets d'interface : permettent de gérer les aspects visuels des différents contrôles graphiques : L'objet Window, document, button, radio, checkbox, etc.
- Les objets des propriétés et des fonctions prédéfinies : fournissent les différentes ressources requises pour la programmation : l'objet String, math, date, navigator, array et object.

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 : nom_objet.nom_propriété

Exempl : voir fichier

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* :

- L'opérateur « + » joue le rôle de concaténation lorsqu'il est utilisé avec la méthode « write ».
- Il est possible de générer du code HTML lors de l'utilisation de la méthode « write ».
- Si le code HTML contient (") et pour ne pas confondre avec les guillemets de write il sera mieux de les remplacer par des apostrophes (').

2. Les différents emplacements du code JavaScript

Activité : Définir les différents emplacements du code JavaScript dans un document HTML.

a) 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 fenetre 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 :
 - On peut écrire le code javascript dans la partie <head>..</head> ou dans la partie <body>..</body>

- La déclaration de fonctions JavaScript peut se faire :
 - Entre <head> et </head>
 - Entre <body> et </body>

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 :

Pour afficher le contenu de la variable moyenne il faut ajouter l'instruction moy() ; qui permet l'appelle de la fonction .

❖ Constatations :

- La déclaration des variables :
 - ✓ 1^{er} méthode : déclaration explicite on utilisant le mot VAR
Syntaxe : var nom_variable = valeur ;
Exemple :.....
 - ✓ 1^{eme} méthode : déclaration implicite sans l'utilisant le mot VAR
Syntaxe : nom_variable = valeur ;
- La visibilité des variables :
 - ✓ Les variables déclarent au dehors et avant les fonctions se sont des variables globales qq soit la méthode de déclaration et ils sont visibles par tous les scripts de la page
 - ✓ Les variables déclarent dans une fonctions avec la méthode explicite (avec mot VAR) se sont des variables local (visible que dans cette fonction) par contre les variables declares avec la méthode implicite (sans le mot var) se sont des variables globale toute en ajoutant l'appelle de la fonction avant l'utilisation de cette variable
- Les types utilisé en javascript :

Type	Description
Numérique	Tout nombre entier ou réel
Chaîne de caractère	Caractère ou une suite de caractère entre deux guillemets
booléens	Les valeurs true ou false
Le mot Null	Indique l'absence d'une valeur

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

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

❖ *Réponse* :

❖ *Constatations* :

✓ Les méthodes de lectures et d'affichages des données

- la méthode prompt : permet la lecture des données dans une boîte de dialogue.
- La méthode alert : permet l'affichage des résultats dans une boîte de dialogue.
- La méthode write : permet l'affichage des résultats dans la page web.

✓ Les fonctions de la conversion des types :

✚ La fonction Number(chaine numérique) :

Permet de convertir une chaîne de caractère a une valeur numérique s'il est possible

Exemple : var a = Number("12") → a=12

var x = Number("12a") → x=NaN

✚ La fonction String(valeur numérique) :

Permet de convertir une valeur numérique à une chaîne de caractère .

Exemple : var a = String(12.36) → a="12.36"

✚ La fonction isNaN(valeur) :

C'est une fonction boolean qui retourne FALSE si la valeur est numérique et TRUE dans le cas contraire.

Exemple : var a = isNaN("12") → a=false

var a = isNaN("12a") → a=true

✓ Les méthodes de l'objet Math :

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

NB : voir livre page 168

✓ Les méthodes de l'objet String :

Nom	syntaxe	rôle	Exemple
length	Chaine.length		
charAt	Caractere=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'incrément

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	Incrément (x++ est le même que x=x+1)	Y=x++ ; Z=++x ;	Y=3 et x=4 Y=4 et x=4
x--, --x	Décrément (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) : elle est faisable avec la méthode « prompt » de l'objet « window » ou à l'aide d'objets graphiques du formulaire HTML
- 🔗 La sortie : est possible en utilisant la méthode « write » de l'objet « document », la méthode « alert » de l'objet « window » ou à l'aide d'objets graphique du formulaire HTML
- 🔗 L'affectation : nom_variable = valeur ou expression

Syntaxe :

L'entrée : nom_variable=prompt("texte de la boite d'invite","valeur par défaut") ;

La sortie : document.write("message"+nom_variable) ;

Alert("message"+nom_variable) ;

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* :

<pre> <script language="javascript"> var t=prompt("donner une heure"); h=t.substr(0,2); m=t.substr(3,2); s=t.substr(6,2); h=Number(h); m=Number(m); s=Number(s); if ((isNaN(t)==false) (t.length != 8)) alert(" la periode doit etre hh:mm:ss"); else if (isNaN(h)) alert("heur non numerique"); else if (isNaN(m)) alert("munite non numerique"); else if (isNaN(s)) alert("seconde non numerique"); else if ((h<0) (h>24)) alert("heur non valide"); else if ((m<0) (m>60)) alert("minute non valide"); else if ((s<0) (s>60)) alert("seconde non valide"); else alert("periode valide"); </script></head> </pre>
--

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

La syntaxe est la suivante :

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.

```
<html>  
<head>  
<title>nombre de jours</title>  
</head>  
<body>  
<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>  
</body>  
</html>
```

Remarques:

- L'instructions break permet de quitter la structure switch après l'exécution du bloc convenable ;
- Lorsque le résultat de l'expression est différent de toutes les valeurs envisagées ; c'est le bloc n relatif à default qui sera exécuté ;
- La clause default (bloc n) est facultative.

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

```
<html><head><title>boucle pour</title>
</head><body><table border="5">
<script language="javascript">
var n = Number(prompt("donner n"));
if (isNaN(n))
alert("erreuuuuuuuur");
else
for (i=1;i<=(n/2);i++)
{
if (n % i == 0)
document.write("<tr> <td>" + i + "</td> </tr>");
}
</script></table></body></html>
```


b) La structure do...while

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

Activité : Ecrire un code JavaScript permettant de lire un nombre ($10 \leq n \leq 100$).

```
<html><head><title>do while</title>
</head><body>
<script language="javascript">
do
{
n=Number(prompt("donner un entier: ",""));
} while(!((n>=1) && (n<=100)));

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

 Remarques:

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)

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 : 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.

```
<html><head><title>boucle while</title>
</head><body>
<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></body></html>
```

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.

```
<html><head><title>Les fonctions</title></head><body>
<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></body></html>
```

Remarques :

- La mention des arguments est facultative mais les parenthèses doivent rester ;
- Une variable déclarée dans une fonction par le mot var est locale à cette dernière, sans le mot var sa portée est globale ;
- Il est possible de définir une fonction sans l'utilisation de la clause « return » : on retrouve ainsi l'équivalent d'une procédure ;
- Il est possible de déclarer une fonction dans la partie entête de la page web <head> et </head>.

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.

```
<html><head><title>onclick</title>
<script language="JavaScript">
function somme()
{
var a=document.f1.e1.value;
var b=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>
</form></body></html>
```

- OnClick : peut être utilisé avec les boutons et les liens hypertextes. Il est possible de programmer l'évènement OnClick avec les objets de type case à cocher, l'option radio et la zone de liste.

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">
  Chaîne : <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">
  Chaîne 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 utilisé 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 à perdu le focus et son contenu est modifié par l'utilisateur) ;
- OnKeyUp : est déclenché au cours de la saisi d'une zone de texte lorsque l'utilisateur relâche une touche.

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 ;

IX. Les formulaires en Javascript :**IX.1. Le contrôle zone de texte**

Activité : 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.

```
<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>
```

Remarques :

- 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 ;

IX.2. Les boutons radio

Activité : 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.

```
<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 degres</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>

```

Remarques :

- 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.

IX.3. Les boutons case à cocher (checkbox)

Activité : 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==true))
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>

```

Remarques :

- 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

IX.4. Liste de sélection

Activité :

```
<html>
<head>
<title>liste</title>
<script language="JavaScript">
function ajout()
{
np=document.f1.znp.value;
adr=document.f1.zadr.value;
var taille=document.f1.carnet.options.length;
var nouveau=true;
if(taille!=0)
{
for (i=0;i<taille;i++)
{
elt=document.f1.carnet.options[i].text;
if(elt==np)
{
nouveau=false;
alert("contact deja existant");
break;
}
}
}
if(nouveau)
{
nelt=new Option(np,adr);
document.f1.carnet.options[taille]=nelt;
}}
function supp()
{
sel=document.f1.carnet.options.selectedIndex;
if(sel==-1)
alert("veuillez sélectionner un élément SVP");
else
{
document.f1.carnet.options[sel]=null;}
}
function adresse()
{
sel=document.f1.carnet.options.selectedIndex;
if(sel==-1)
alert("veuillez sélectionner un élément SVP");
```

```

else
{
np=document.f1.carnet.options[sel].text;
adr=document.f1.carnet.options[sel].value;
alert("l'adresse mail de: "+np+" est: "+adr);
}
}
</script>
</head>
<body>
<p align="center"><font color="red" size="6">CARNET D'ADRESSES</font></p>
<form name="f1">
Nom et Prénom: <input type="text" name="znp" size="30"><br><br>
Adresse E-mail:<input type="text" name="zadr" size="30"><br><br>
<table>
<tr>
<td><input type="button" name="b1" value="Ajouter" OnClick="ajout()"></td>
<td><input type="button" name="b2" value="Supprimer" OnClick="supp()"></td>
<td><input type="button" name="b3" value="Retrouver Adresse E-mail"
OnClick="adresse()"></td>
</tr>
</table>
<select name="carnet" size="5"></select>
</form></body>
</html>

```

Remarques :

- 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.

X. 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 ».

```
<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.