



مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle
et de la Promotion du Travail

Institut Spécialisé de Technologie Appliquée de Gestion Bab Tizimi - Meknès

Le langage JavaScript

Année de Formation : 2020/2021

Table des matières

Définition:.....	4
Déclaration d'un script :	4
Les notions de base d'un script JavaScript.....	5
Les fonctions de Base de JavaScript :	5
Les fonctions de conversion :	10
Les fonctions mathématiques:	13
Les fonctions de test :.....	17
Les tableaux en JavaScript.....	19
Création d'un tableau en JS :.....	19
Remplir un tableau en JS	19
Afficher le contenu d'un tableau en JS.....	20
Tableaux associatifs :	23
Tableaux d'objets :	23
Les fonctions en JavaScript.....	25
Définition :	25
Appel d'une fonction :	25
Les fonctions de date en JavaScript	26
Introduction :	26
Les fonctions de date :	26
Les fonctions d'heure:	27
Remarque :	27
Les fonctions de modification de la date:	27
Les fonctions de modification de l'heure:.....	28
Remarque :	28
Calcul de la différence de date :	28
Les évènements en JavaScript.....	30
Introduction :	30
La structure DOM d'un document HTML (Document Object Model)	33
Accès aux éléments HTML :	34
Accéder aux éléments HTML en utilisant les sélecteurs :	38
Accès aux éléments Radios et checkbox en JavaScript :	39
Les événements en JavaScript :	47
Le JavaScript et le CSS3	48

Introduction :.....	48
Passage des données entre les pages web	51
Utilisation de la propriété localStorage :.....	51
Utilisation de la propriété sessionStorage :	52
Utilisation des cookies:.....	52
Création de cookies :	53
Récupération de cookies :	53
Passage des données d'une page à une autre par cookie :.....	53
Autres méthodes :	54
La méthode setTimeout() :	54
Les propriétés de la fenêtre Window.....	58
Les méthodes d'un Document	59
.....	60

Le langage JavaScript

Définition:

C'est un Langage de programmation côté client qui permet de dynamiser un document HTML d'améliorer la présentation des éléments HTML.

Exemple:

- récupérer les données d'un serveur mysql.
- gérer les éléments HTML (ajouter- supprimer.....)
- mettre en forme (application des styles) des éléments HTML
- animer les éléments HTML

Déclaration d'un script :

- Au niveau de la partie « head » :

Dans la balise <head></head>, comme on a vu pour les styles , on peut ajouter un script JavaScript, en utilisant la balise :

```
<head>
  <title>Javascript</title>
  <script>
    /* Ecrire Ici votre script Javascript */
  </script>
</head>
```

- Au niveau de la partie « body »

On peut ajouter le script au début ou à la fin de la balise <body> main il est recommandé de l'utiliser à la fin de cette balise pour assurer le chargement de tous les éléments HTML du document HTML:

```
<body>
  <script>
    /* Ecrire Ici votre script Javascript */
  </script>
```

```
</body>
```

- Au niveau d'un fichier externe :

On peut utiliser notre script dans un fichier externe nommé : **Javascript.js**,
sans oublier l'ajout du lien de ce fichier dans notre document HTML,

```
<script src="Monsite/js/javascript.js"></script>
```

Les notions de base d'un script JavaScript.

Les fonctions de Base de JavaScript :

- **Alert()** : c'est une fonction qui permet d'afficher un message dans une boîte de dialogue (MessageBox).

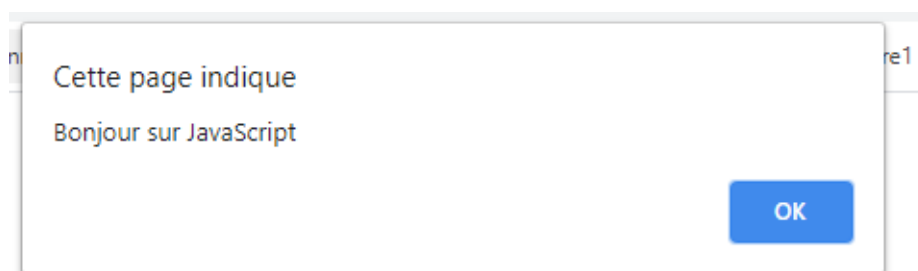
```
<script>

/* Ecrire Ici votre script Javascript */

alert("Bonjour sur JavaScript");

</script>
```

Le résultat est :

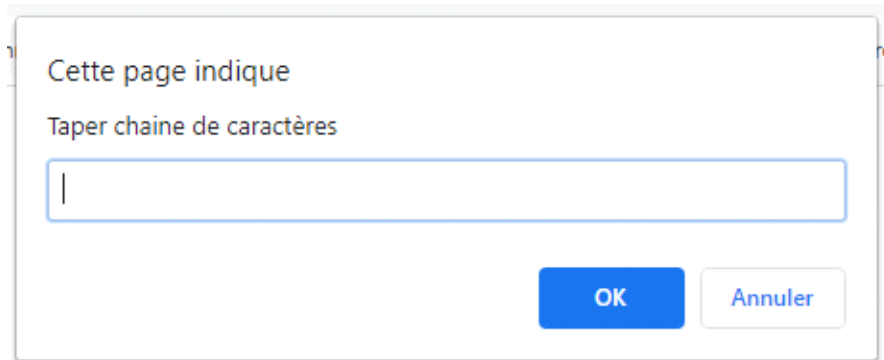


- **prompt ()** : c'est une fonction qui permet d'afficher une boîte de dialogue de saisie (input Box).

Exemple :

```
<script>
  /* Ecrire Ici votre script Javascript */
  var x= prompt ("Taper chaine de caractères");
  alert ("La chaine saisie est :"+x)
</script>
```

Le résultat est :

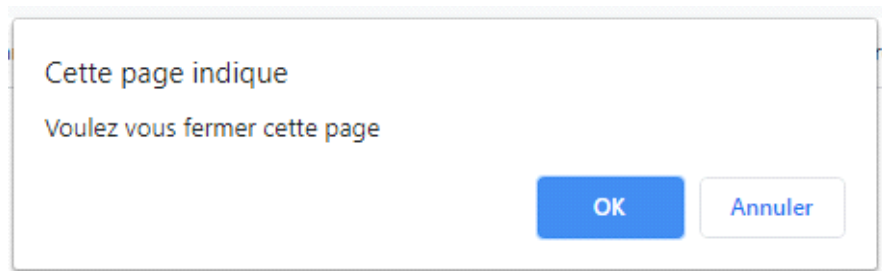


- **Confirm()** : c'est une fonction qui permet d'afficher une boîte de dialogue de confirmation.

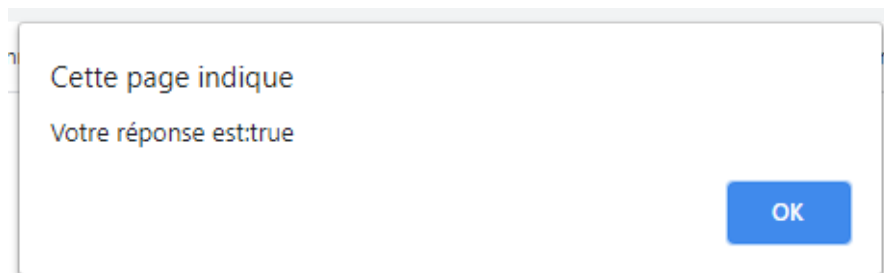
Exemple :

```
<script>
  /* Ecrire Ici votre script Javascript */
  var x= confirm("Vlez vous fermer cette page");
  alert("Votre réponse est:"+x)
</script>
```

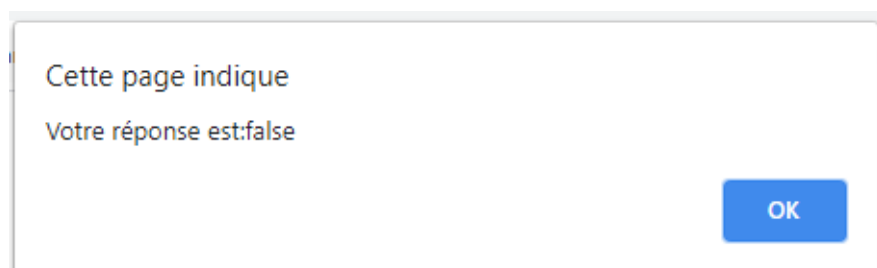
Le résultat est que x aura la valeur true ou false.



En cliquant sur OK la boîte sera :



En cliquant sur Annuler la boîte sera :



- **Les variables JavaScript :**

En JavaScript, on peut déclarer une variable en utilisant **var**.

Exemples : (NB : JavaScript est sensible à la casse)

```
Var x ;      // déclaration d'une variable
```

```
var x , y ;      // déclaration de plusieurs variables
```

```
var x= "20" ;      // déclaration d'une variable de type chaine initialisée par x= "20".
```

var x=20 ; // déclaration d'une variable de type nombre entier.

var x=20.50 ; // déclaration d'une variable de type nombre réel.

var x=new Array() ; // déclaration d'une variable de type tableau.

var x=true ; // déclaration d'une variable de type booléen.

On peut utiliser la clause : **let** pour déclarer une variable en JavaScript, mais celle rend une variable locale tandis que la clause **Var** permet l'utilisation des variables globale. (**Let** n'est pas prise en charge par certains navigateurs).

Remarque : On peut déclarer une variable sans casse de tête comme suit (sans var ou let):

x= "20" ;

x=20 ;

- **Les opérateurs en JavaScript :**

- **Les opérateurs arithmétiques :**

opérateur	Nom	But	Exemple
+	Addition	Ajoute deux nombres.	6 + 9
-	Soustraction	Soustrait le nombre de droite de celui de gauche.	20 - 15
*	Multiplication	Multiplie les deux nombres.	3 * 7
/	Division	Divise le nombre de gauche par celui de droite.	10 / 5
%	Reste (quelquefois nommé modulo)	Renvoie le reste de la division du nombre de gauche par celui de droite.	8 % 3 (renvoie 2, car 3 est contenu 2 fois dans 8, et il reste 2.)

(Pour les nombres de type entier ou réel)

- **Les opérateurs d'incrémentation :**

i++	i=i+1
------------	--------------

i--	i=i-1
X+=y	X=x+y
x/=y	X=x/y

- Les opérateurs logiques:

- **&&** : Et logique, sa table de vérité est :

a	b	a && b
0	0	0
1	0	0
0	1	0
1	1	1

- **||** ou le Ou (or) logique :

a	b	a b
0	0	0
1	0	1
0	1	1
1	1	1

- **!** ou le non logique :

a	!a
0	1
1	0

- Les opérateurs de comparaison en JavaScript:

>	supérieur
<	inférieur
>=	supérieur ou égale
<=	inférieur ou égale
!=	différent en valeur
!==	différent en valeur et en type
==	égalité des valeurs

===

égalité des valeurs et des types

- **L'opérateur de concaténation:**

+

Exemple :

```
<script>
x="ofppt";
y="itag bab tizimi";
z=x+" "+y;
alert("le résultat est :"+z)
</script>
```

Remarque :

```
/* ceci est un commentaire d'un seule ligne */
```

Ou

```
/* ceci est un commentaire
de plusieurs
lignes */
```

Les fonctions de conversion :

- **parseInt()** : Cette fonction permet de convertir la valeur d'une variable en nombre entier.

Exemple1 :

```
<script>
x="25";
y="30";
z=x+y;

/*la concaténation*/

alert("le résultat est :"+z)
```

```

/*la conversion en entier*/

z=parseInt(x)+parseInt(y)

alert("le résultat est :"+z)

</script>

```

Exemple2 :Exécuter le script suivant sous un navigateur.

```

<script>

var x=prompt("Taper le premier nombre entier:");

var y=prompt("Taper le deuxième nombre entier:");

z=parseFloat(x)+parseFloat(y);

alert("la somme de"+x+"et"+y+" est =" +z);

</script>

```

- **parseFloat()** : Cette fonction permet de convertir la valeur d'une variable en nombre réel.

Exemple : Exécuter le script suivant sous un navigateur.

```

<script>

var x=prompt("Taper le premier nombre réel:");

var y=prompt("Taper le deuxième nombre réel:");

z=parseFloat(x)+parseFloat(y);

alert("la somme de"+x+"et"+y+" est =" +z);

</script>

```

La fonction « toFixed » permet de formater un nombre réel.

```

alert("la somme de"+x+"et"+y+" est =" +z.toFixed(2));

```

Le résultat est :



- **Eval()** : Cette fonction permet de rendre une chaîne de caractères sous la forme d'expression .

Exemple :

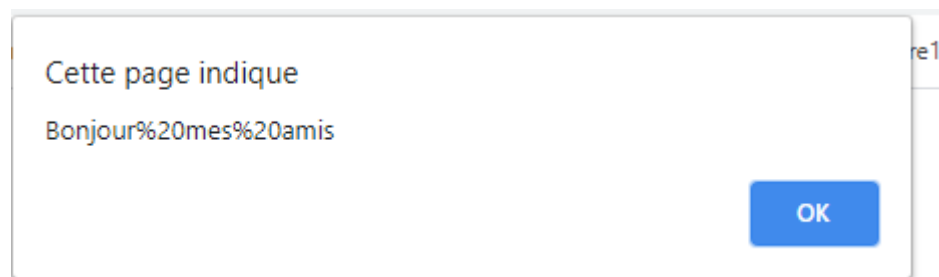
```
z= "6+9+20"
x=eval(z)
alert(x);
```

Le résultat est : **35.**

Escape () : permet de convertir un espace en %20.

Exemple :

```
<script>
alert(escape("Bonjour mes amis"));
</script>
```

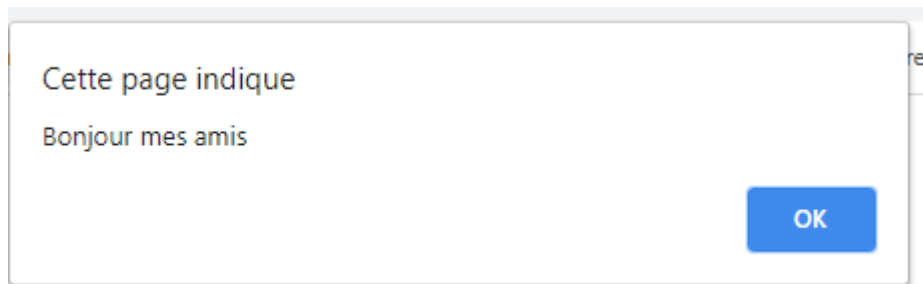


Unescape () : permet de convertir %20 en espace ..

```
<script>
alert(unescape("Bonjour%20mes%20amis"));
```

```
</script>
```

Le résultat est :



Les fonctions mathématiques:

random() : cette fonction permet de retourner en hasard un nombre entre 0 et 1.

Exemple :

```
<script>
var x=Math.random();
alert(x);
</script>
```

La valeur est entre 0 et 1.

- Pour afficher un nombre en deux valeurs (0, x), il suffit de multiplier par la valeur de x.

Exemple :

```
<script>
var x=5*Math.random();
alert(x);
</script>
```

Les valeurs sont toujours entre 0 et 5.

- Pour afficher un nombre en deux valeurs (min, max) ; la procédure est la suivante :

Soit $h = \text{Math.random}()$;

Alors : $0 < h < 1$.

$$0 < h (\text{Max} - \text{min} + 1) < (\text{Max} - \text{min} + 1)$$

$$\text{min} < h (\text{Max} - \text{min} + 1) + \text{min} < (\text{Max} + 1)$$

la valeur entre min et (max+1) est : $h (\text{max} - \text{min} + 1)$.

Exemple :

```
<script>
min=5;
max=10;
h=Math.random();
t=h*(max-min +1)+min;
alert(t);
</script>
```

Le résultat est entre 5 et 11.

- **floor()** :cette fonction permet d'arrondir un nombre réel vers sa partie entière .

```
x=25.26;
alert(Math.floor(x));
```

Le résultat est : 25.

Pour afficher un nombre entier en hasard ; il suffit d'appliquer la fonction `floor()` sur la valeur de (t) ci-dessus :

```
<script>
min=5;
max=10;
h=Math.random();
t=h*(max-min +1)+min;
alert(Math.floor(t));
</script>
```

Le résultat est toujours un nombre entier entre 5 et 10(y compris).

- `ceil ()`: cette fonction permet d'arrondir un nombre réel vers sa partie entière +1 .

Exemple1 :

```
min=15.5;

alert(Math.ceil(min));
```

Le résultat est : 16.

Exemple2 :

```
<script>
min=5;
max=10;
h=Math.random();
t=h*(max-min +1)+min;
alert(Math.ceil(t));
</script>
```

Le résultat est toujours un nombre entier entre 6 et 11(y compris).

- **min ()**, **max ()** : ces deux fonction permettent d'afficher le minimum ou le maximum d'une suite de nombres réels.

```
<script>  
min=Math.min(0.2,1,1.5,25,36) ;  
max=Math.max(1.2,1,1.5,25,36);  
alert(" le min est: "+min+" et le max est : "+max);  
</script>
```

Le résultat est :



-**round()** : permet d'arrondir un nombre à l'entier le plus proche par min ou par max .

Exemple :

```
<script>  
x=25.4;  
y=25.6;  
alert("la valeur 1 est :"+Math.round(x)+" est la valeur 2  
est :"+Math.round(y));  
</script>
```

Le résultat est :



Autres fonctions mathématiques :

`y=Math.abs(x); // la valeur absolue`

`y=Math.sqrt(x); // la racine carré.`

`y=Math.pow(x,i); //x puissance i.`

`y=Math.cos(x); // cosinus de x`

`y=Math.sin(x); // le sinus de x.`

`y=Math.tan(x); // la tangente de x`

`var p=Math.PI; //3.14`

Les fonctions de test :

`Number.isInteger(x)` : permet de vérifier si le nombre en paramètre est un entier ou non.

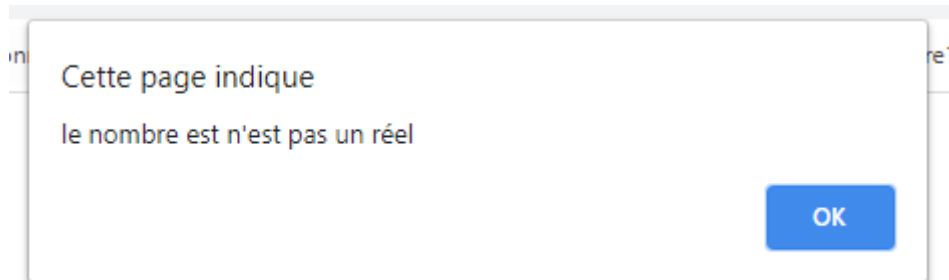
Exemple :

```
<script>
x=25;
if(Number.isInteger(x))
alert(x+" est un entier");
else
alert(x+" n'est pas un entier ");
</script>
```

Pour vérifier si un nombre pris en paramètre est un entier ou réel ; on utilise le test suivant :

```
<script>
x=25;
if(parseInt(x)==parseFloat(x))
    alert("le nombre est n'est pas un réel");
else
    alert("le nombre est réel");
</script>
```

Le résultat :



Exercices d'application TP1.

Les tableaux en JavaScript

Création d'un tableau en JS :

Un tableau est une variable complexe qui contient plusieurs valeurs ; La classe Array a un constructeur par défaut Array() qui permet de créer un tableau.

```
<script>
var t=new Array();
var K=new Array(10); // tableau contenant 10 cases.
</script>
```

Remplir un tableau en JS

Le script pour remplir un tableau est :

Exemple1 :

```
<script>

var t=new Array();

for(var i=0;i<5;i++)

t[i]=prompt("Taper la valeur de t["+i+"]");

</script>
```

Exemple2 :

Le tableau suivant contient les jours d'une semaine.

```
<script>
var t=new Array();
t[0]="Lundi";
t[1]="Mardi";
t[2]="Mercredi";
t[3]="Jeudi";
t[4]="Vendredi";
t[5]="Samedi";
t[6]="Dimanche";
alert(t.length);
</script>
```

Afficher le contenu d'un tableau en JS

En utilisant la fonction **write()** de la classe document, on peut écrire le résultat d'affichage du tableau sur une page web.

Exemple1 :

```
<body>
<p>Les valeurs du tableaux sont :</p>
<script>
var t=new Array();
//remplissage du tableau
for(var i=0;i<5;i++)
t[i]=prompt("Taper la valeur de t["+i+"]");
//affichage du tableau
for(var i=0;i<5;i++)
document.write("La valeur de t["+i+"] est :"+t[i] + "<br>");
</script>
</body>
```

Le résultat est :

Les valeurs du tableaux sont :

La valeur de t[0] est :5
 La valeur de t[1] est :6
 La valeur de t[2] est :7
 La valeur de t[3] est :8
 La valeur de t[4] est :9

Exemple2 :

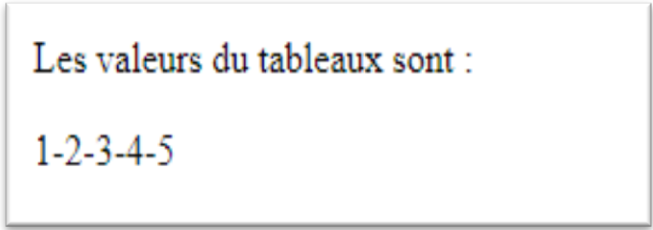
```
<body>
<p>Les valeurs du tableaux sont :</p>
<script>
var t=new Array();
t[0]="Lundi";
t[1]="Mardi";
t[2]="Mercredi";
t[3]="Jeudi";
t[4]="Vendredi";
t[5]="Samedi";
t[6]="Dimanche";
alert(t.length);
```

```
for(var i=0;i<7;i++)
document.write("La valeur de t["+i+"] est :"+t[i] + "<br>");
</script>
</body>
```

De même, on peut utiliser la fonction **join ()** qui permet de retourner le contenu d'un tableau sous la forme d'une chaîne de caractères.

```
<body>
<p>Les valeurs du tableaux sont :</p>
<script>
var t=new Array();
//remplissage du tableau
for(var i=0;i<5;i++)
t[i]=prompt("Taper la valeur de t["+i+"]");
//affichage du tableau
document.write(t.join("-")); // où - est le séparateur des valeurs du
tableaux.
</script>
</body>
```

Le résultat est :



Les valeurs du tableaux sont :
1-2-3-4-5

Au contraire de la fonction **join ()**, la méthode **split()** permet de transformer une chaîne de caractères en tableau.

Exemple :

```
<script>
var t=new Array();
// convertir une chaîne en tableau
x="1+2+3+4+5+6+7+8";
t= x.split("+")
//affichage du tableau
```

```
document.write(t.join("-")); // où - est le séparateur des
valeurs du tableaux.
</script>
```

Le résultat est :

```
Les valeurs du tableaux sont :
1-2-3-4-5-6-7-8
```

Pour trier un tableau en ordre croissant, on utilise la méthode **sort()**.

Exemple :

```
<script>
var t=new Array();
// convertir une chaine en tableau
x="1+12+31+14+11+16+17+8";
t= x.split("+")
//affichage du tableau
t.sort();
document.write(t.join("-")); // où - est le séparateur des valeurs du
tableaux.
</script>
```

Le résultat est :

```
Les valeurs du tableaux sont :
1-11-12-14-16-17-31-8
```

(A remarquer; le triage est alphabétique)

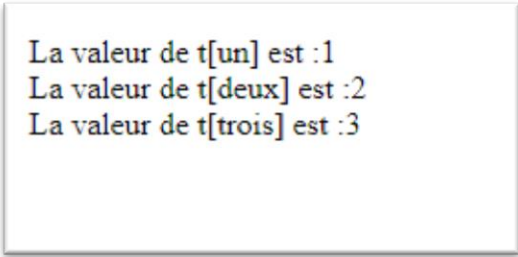
Tableaux associatifs :

Le *tableau associatif* est un tableau qui permet d'associer une valeur à une clé autre qu'un indice numérique.

Exemple :

```
<script>
//création d'un tableau
var t=new Array();
t = { "un" : 1, "deux" : 2, "trois": 3 };
// afficher le contenu du tableau
for(var i in t)
document.write("La valeur de t["+i+"] est :"+t[i]+"<br>");
</script>
```

Le résultat est :



```
La valeur de t[un] est :1
La valeur de t[deux] est :2
La valeur de t[trois] est :3
```

Pour afficher les clés on utilise la boucle for comme suit :

```
for(var i in t)
document.write("la clé est : " +i) ;
```

Tableaux d'objets :

Exemple :

```
<script>
var t=new Array();
t=[
{
  "num":1,
  "nom":"a",
  "prenom":"ap"
},
{
  "num":2,
  "nom":"a2",
  "prenom":"ap2"
```

```
},  
{  
  "num":3,  
  "nom":"a3",  
  "prenom":"ap3"  
}  
]  
for(var i in t)  
document.write("NC: "+t[i].nom+"<br/>");  
</script>
```

Ou bien

```
for(var i =0;i<t.length;i++)  
  
document.write("NC: "+t[i].nom+"<br/>");
```

TP2 Exercice 1 et Exercice 2

Les fonctions en JavaScript

Définition :

En JavaScript, une fonction est un bloc d'instructions qui est précédé par le mot clé **function**.

Exemple 1:

```
function somme(x,y)
{
  alert(parseInt(x)+ parseInt(y));
}
```

Remarque :

En JavaScript une fonction peut retourner une valeur, comme peut ne pas retourner une valeur (méthode).

Exemple 2:

```
function somme(x,y)
{
  return (parseInt(x)+ parseInt(y));
}
```

Appel d'une fonction :

L'appel de la fonction se fait par son nom.

Exemple 1 :

```
var x=prompt("Taper le premier nombre");
var y=prompt("Taper le premier nombre");
somme(x,y);
function somme(x,y)
{
  alert(parseInt(x)+ parseInt(y));
}
```

Exemple 2 :

```
var x=prompt("Taper le premier nombre");
var y=prompt("Taper le premier nombre");
var z=somme(x,y);
alert("la somme est :"+z);
```

```
function somme(x,y)
{
    return parseInt(x)+ parseInt(y);
}
```

Exercices d'application : TP2 Exercice 1 et Exercice 2

Les fonctions de date en JavaScript

Introduction :

La classe Date contient le constructeur qui permet de créer une variable de type Date.

Exemple 1:

```
Var d=new Date() ; //d a valeur de d est la date système.
Document.write(d) ;
```

Le résultat est :

```
Wed Dec 23 2020 20:52:13 GMT+0100 (UTC+01:00)
```

Exemple 2:

```
Var d=new Date() ; //d a valeur de d est la date système.
alert(d) ;
```

Les fonctions de date :

- La fonction **getDate()** : Renvoie le jour de la date.
- La fonction **getMonth()** : Renvoie le mois -1 de la date.
- La fonction **getFullYear()** : Renvoie le jour de la date.

Exemple :

```
<script>
var d=new Date();
date=d.getDate()+"/"+(d.getMonth()+1)+"/"+d.getFullYear();
alert(" la date du système est :"+date);
</script>
```

Les fonctions d'heure:

- La fonction **getHours()** : Renvoie l'heure du système.
- La fonction **getMinutes()** : Renvoie l'heure du système.
- La fonction **getSeconds()** : Renvoie l'heure du système.
- La fonction **getMiliSeconds()** : Renvoie l'heure du système.

Exemple :

```
<script>
var d=new Date();
t=d.getHours()+":"+d.getMinutes()+":"+d.getSeconds();
alert(" l'heure système est :"+t);
</script>
```

Remarque :

La méthode **getTime()** de l'objet **Date** retourne le nombre de millisecondes écoulées depuis le 1er janvier 1970.

```
<script>
var t=new Date();
d=t.getTime()
alert(d);
</script>
```

Les fonctions de modification de la date:

Pour modifier une date il suffit de définir sa valeur une deuxième fois :

```
var d=new Date("2020,09,09");
```

Les fonctions qui permettent de modifier le jour, le moi, l'année sont :

```
<script>
var d=new Date("2020,09,09");
//modifier le jour
d.setDate("25");
// la modification se fait par ajout de +1 au Mois.
d.setMonth("11");
//Modifier l'année
d.setFullYear("2021");
date=d.getDate()+"/"+(d.getMonth()+1)+"/"+d.getFullYear();
alert(" la date du système est :"+date);
```

```
</script>
```

Les fonctions de modification de l'heure:

Les fonctions qui permettent de modifier l'heure, les minutes, et les secondes sont :

```
<script>
var t=new Date();
t.setHours("12");
t.setMinutes("50");
t.setSeconds("25");
//en millisecondes
</script>
```

Remarque :

Pour modifier l'heure, on peut utiliser la fonction **setTime()** qui est en millisecondes.

```
<script>
var d=new Date();
d.setTime("225252552"); //en millisecondes
t=d.getHours()+":"+d.getMinutes()+":"+d.getSeconds()
alert(t);
</script>
```

Calcul de la différence de date :

Ma date de naissance =

29 année(s), 11 mois, et 27 jour(s)

Vous êtes sur Terre depuis :

En mois :	359 mois
En jours :	10955 jours
En heures :	262920 heures
En minutes :	15775200 minutes
En secondes :	946512000 secondes

El code en Javascript est :

```
<script>
var d1=new Date();
var d2=new Date("1990,12,31");
```

```
var diff=d1-d2 ;//en millicecondes  
diff=diff/1000;//en secondes  
diff=diff/3600;//en heures  
diff=diff/24; //en jours  
diff=diff/366;//en années  
alert(diff);  
</script>
```

Exercice :

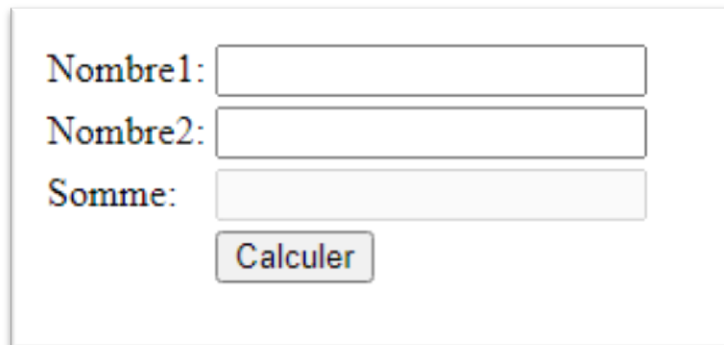
Ecrire le code de la boîte ci-dessus.

Les évènements en JavaScript

Introduction :

En JavaScript, un évènement est une action qui se produit sur un élément HTML ; cet événement est relié à un code directement ou à une fonction.

Exemple 1: Utilisation du Formulaire.



Nombre1:	<input type="text"/>
Nombre2:	<input type="text"/>
Somme:	<input disabled="" type="text"/>
<input type="button" value="Calculer"/>	

Le code HTML de cette Boite est :

```
<form>
  <table>
    <tr>
      <td>Nombre1:</td> <td><input type="nombre" id="n1" name="n1" /></td>
    </tr>
    <tr>
      <td>Nombre2:</td> <td><input type="nombre" id="n2" name="n2"/></td>
    </tr>
    <tr>
      <td>Somme:</td> <td><input type="nombre" id="n3" name="n3" disabled /></td>
    </tr>
    <tr>
      <td></td> <td><input type="button" value="Calculer" onclick="Afficher()"/></td>
    </tr>
  </table>
</form>
```

Le code JavaScript de la fonction Afficher() est :

```
<script>
function Afficher()
{
  n3.value=parseInt(n1.value)+parseInt(n2.value);
}
</script>
```

Remarque :

- on peut utiliser ce code dans la balise <button> comme suit :

```
onclick="n3.value=parseInt(n1.value)+parseInt(n2.value);"/></td>
```

Exemple 2: Utilisation du Formulaire.

- on peut utiliser ce code dans la balise <Form> comme suit :

```
<form oninput="n3.value=parseInt(n1.value)+parseInt(n2.value);">
  <table>
    <tr>
      <td>Nombre1:</td> <td><input type="nombre" id="n1" name="n1" /></td>
    </tr>
    <tr>
      <td>Nombre2:</td> <td><input type="nombre" id="n2" name="n2"/></td>
    </tr>
    <tr>
      <td>Somme:</td> <td><input type="nombre" id="n3" name="n3" disabled /></td>
    </tr>
    <tr>
      <td></td> <td><input type="button" value="Calculer"/></td>
    </tr>
  </table>
</form>
```

Ou

```
<form oninput="Afficher()">
```

Et le code JavaScript :

```
function Afficher()
{
  n3.value=parseInt(n1.value)+parseInt(n2.value);
}
```

NB : Les événements JavaScript sont :

1. **click** (onClick)

Cet événement est capturé sur un objet quand on clique dessus. Idéal pour les boutons, images, hyperliens, vidéos...

2. **dblClick** (onDbClick)

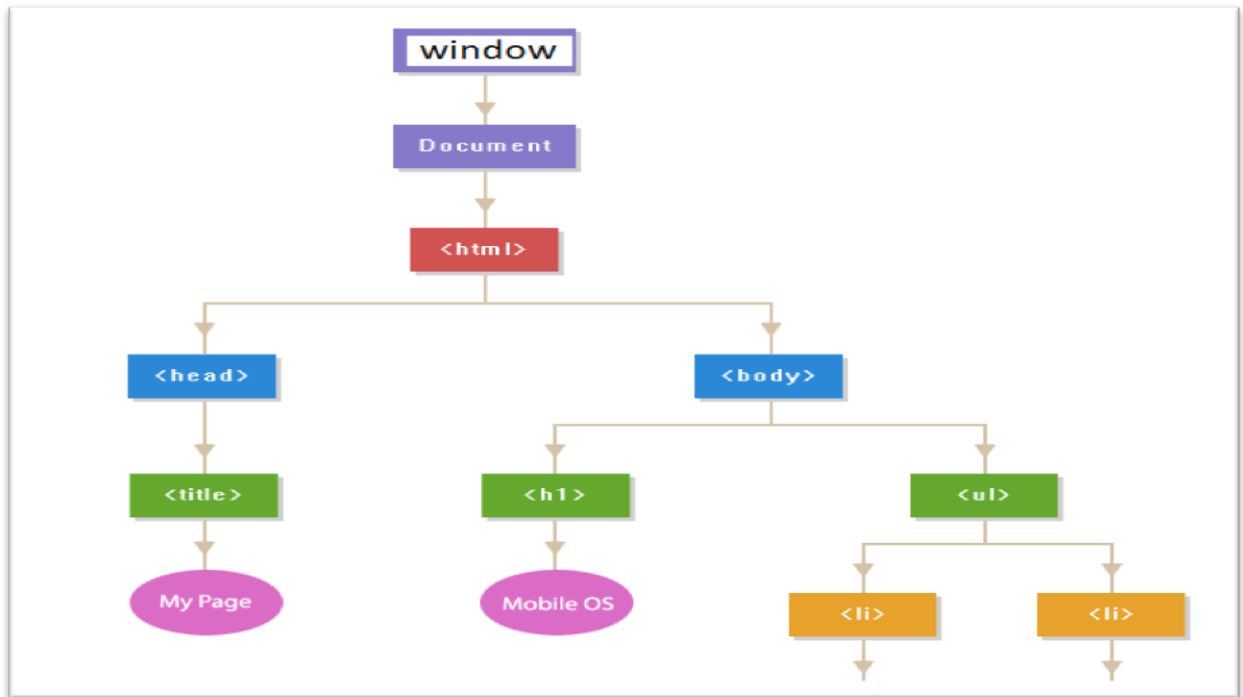
Quand on clique sur un objet deux fois de suite (double clic).

3. **mouseover** (onMouseOver)
Quand on survole un objet avec le curseur de la souris.
4. **mouseout** (onMouseOut)
Quand on quitte l'objet avec la souris après l'avoir survolé.
5. **focus** (onFocus)
Quand on active un élément (quand on place le curseur dans un champ de formulaire, par click ou par tabulation, pour commencer la saisie par exemple).
6. **blur** (onBlur)
Quand un élément perd le focus (quitter un champ de formulaire après être activé par exemple).
7. **keydown** (onKeyDown)
Quand une touche du clavier est enfoncée.
8. **keyup** (onKeyUp)
Quand une touche du clavier est relâchée.
9. **keypress** (onKeyPress)
Quand une touche du clavier est maintenue enfoncée.
10. **load** (onLoad)
Quand un élément est chargé par le navigateur. Elle peut être appliquée à la page entière (balise **<body>**), dans ce cas l'événement se produira quand tous les éléments de la page seront chargés.
11. **unload** (onUnload)
Quand la page en cours est fermée ou quittée.
12. **resize** (onResize)
Quand l'internaute redimensionne la taille de la fenêtre du navigateur.
13. **select** (onSelect)
Quand l'internaute essaie de sélectionner le texte contenu dans l'objet accueillant l'événement.
14. **change** (onChange)
Quand l'internaute change le contenu d'un élément (liste de sélection ou zone de texte par exemple).
15. **submit** (onSubmit)
Quand l'internaute clique sur n'importe quel bouton de type **submit** présent dans la page (ou dans le formulaire).
16. **mousedown** (onMouseDown)
Quand l'internaute appuie sur n'importe quel bouton de la souris.

17. **mouseup** (onMouseUP)
Quand le bouton de la souris est relâché.
18. **mousemove** (onMouseMove)
Quand l'internaute fait bouger le curseur de la souris dans la zone accueillant l'événement.
19. **dragStart** (onDragStart)
Se produit quand l'internaute commence le déplacement d'un élément par "glisser-déposer" (Drag & Drop).
20. **dragEnter** (onDragEnter)
Se produit quand l'internaute entre dans la zone où sera déposé l'élément glissé.
21. **dragOver** (onDragOver)
Se produit quand l'internaute survole la zone où sera déposé l'élément glissé (même si le concept est proche de celui de **onDragEnter**, ils sont des événements légèrement différents).
22. **drop** (onDrop)
Se produit quand l'internaute dépose l'élément glissé dans la zone prévue à cet effet.

La structure DOM d'un document HTML (Document Object Model)

Le DOM est un modèle représentatif qui permet de décrire la structure d'un document HTML ; il assure l'accès aux éléments HTML pour les modifier, supprimer, Ajouter ou les mettre à jours.



Accès aux éléments HTML :

Parmi les fonctions qui permettent d'accéder aux éléments HTML on a :

- **getElementById()** : c'est une méthode de la classe document qui permet d'accéder à un élément HTML par son Id ; si l'élément n'existe pas cette méthode retourne **null**.

Exemple : Soit l'interface suivante dont le code HTML est au dessous.

Nombre1:	<input type="text"/>
Nombre2:	<input type="text"/>
Somme:	<input type="text"/>
<input type="button" value="Calculer"/>	

Son code HTML est :

```

<div>
  <table>
    <tr>
      <td>Nombre1:</td> <td><input type="nombre" id="n1" /></td>
    
```

```

</tr>
<tr>
<td>Nombre2:</td> <td><input type="nombre" id="n2" /></td>
</tr>
<tr>
<td>Somme:</td> <td><input type="nombre" id="n3" disabled /></td>
</tr>
<tr>
<td></td> <td><input type="button" value="Calculer"/></td>
</tr>
</table>
</div>

```

Le code JavaScript qui permet de calculer la somme de deux nombres entiers est :

1^{ère} étape : On associer un événement au bouton :

```
<input type="button" value="Calculer" onclick="Afficher()"/>
```

2^{ème} étape :

On créer une fonction JavaScript qui permet de calculer la somme.

```

<script>
function Afficher()
{
// accéder 'élément HTML
var n1=document.getElementById("n1");
var n2=document.getElementById("n2");
var n3=document.getElementById("n3");
//récupérer leurs valeurs
var x1=n1.value;
var x2=n2.value;
var s=parseInt(x1)+parseInt(x2);
n3.value=s;
}
</script>

```

Remplaçant le zone n3 par une label.

```

<tr>
<td>Somme:</td> <td><label id="n3">000</label></td>
</tr>

```

Le code de n3 devient :

```
n3.innerHTML=s;
```

Attention : faire la différence entre value et innerHTML en JavaScript.

- **getElementByTagName()** : c'est une méthode de la classe document qui permet d'accéder aux éléments HTML par leur tagName (nom de la balise);

Exemple :

Soit le code HTML suivant :

```
<div>
  <p>Mon premier paragraphe</p>
  <p>Mon deuxième paragraphe</p>
  <p>Mon troisième paragraphe</p>
  <p>Mon quatrième paragraphe</p>
  <input type="button" Value="Afficher" onclick="Afficher()" />
</div>
```

Pour accéder aux éléments HTML (p) on peut utiliser le code suivant :

```
<script>
function Afficher()
{
  // ps est un tableau contenant les élément HTML de balise <p>
  var ps=document.getElementsByTagName("p");
  // afficher le nombre de paragraphes
  alert(ps.length) ;
  //afficher la valeur du 1er paragraphe
  alert(ps[0].innerHTML);
}
</script>
```

- **getElementByClassName()** : c'est une méthode de la classe document qui permet d'accéder aux éléments HTML par leur attribut class sous la forme d'un tableau.

Exemple 1 :

Soit le tableau suivant :

<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
000	000
<input type="button" value="Calculer"/>	

Son code HTML est :

```
<table>
  <tr>
    <td><input type="number" class="t" /></td>
```

```

        <td><input type="number" class="t2" /></td>
    </tr>
    <tr>
        <td><input type="number" class="t" /></td>
        <td><input type="number" class="t2" /></td>
    </tr>
    <tr>
        <td><input type="number" class="t" /></td>
        <td><input type="number" class="t2" /></td>
    </tr>
    <tr>
        <td><input type="number" class="t" /></td>
        <td><input type="number" class="t2" /></td>
    </tr>
    <tr>
        <td><input type="number" class="t" /></td>
        <td><input type="number" class="t2" /></td>
    </tr>
    <tr>
        <td><label id="res1">000</label></td> <td><label id="res2">000</label></td>
    </tr>
    <tr>
        <td></td> <td><input type="button" value="Calculer"
onclick="Afficher()" /></td>
    </tr>
</table>

```

Le code JavaScript pour calculer la somme de chaque colonne est :

```

<script>
function Afficher()
{
    var t=document.getElementsByClassName("t");
    var t2=document.getElementsByClassName("t2");
    var res1=document.getElementById("res1");
    var res2=document.getElementById("res2");
    s=0;
    s1=0;
    for(i=0;i<t.length;i++)
        s=s+parseInt(t[i].value);
    for(i=0;i<t2.length;i++)
        s1=s1+parseInt(t2[i].value);
    res1.innerHTML=s;
    res2.innerHTML=s1;
}
</script>

```

Le résultat est comme exemple ;

2	2
2	2
2	2
2	2
8	8
<input type="button" value="Calculer"/>	

Accéder aux éléments HTML en utilisant les sélecteurs :

En JavaScript, on peut accéder au contenu d'un élément HTML par un de ses sélecteurs.

- **QuerySelector()** : Permet d'accéder à un élément par un sélecteur qui doit être unique.

Exemple 1:

Soit le code HTML suivant :

```

<input type="button" value="Afficher" onclick="Afficher()"/>
```

Le code JavaScript :

```
<script>
function Afficher()
{
    var img=document.querySelector("img");
    alert("La source est :"+img.src);
}
</script>
```

Exemple 2:

Soit le code HTML suivant :

```
<a href="http://www.ofppt.ma">Ofppt</a>
<input type="button" value="Afficher" onclick="Afficher()"/>
```

Le code JavaScript :

```
<script>
function Afficher()
{
    var a=document.querySelector("a[href='http://www.ofppt.ma']");
    alert("Le lien est :"+a.innerHTML);
}
</script>
```

- **querySelectorAll()** : Permet d'accéder à plusieurs éléments HTML par un sélecteur ou une combinaison de sélecteurs ; le résultat est sous la forme d'un tableau.

Exemple 1 :

Soit le code HTML suivant :

```
<div id="div1">
  <p>paragraphe 1</p>
  <p>paragraphe 2</p>
  <p>paragraphe 3</p>
</div>
<p>paragraphe 4</p>
<p>paragraphe 5</p>
<p>paragraphe 6</p>

<input type="button" value="Afficher" onclick="Afficher()"/>
```

Le code JavaScript :

```
<script>
function Afficher()
{
  //afficher le nombre de paragraphes du document
  var ps=document.querySelectorAll("p");
  alert(ps.length);
  //afficher le nombre de paragraphes du #div1
  ps=document.querySelectorAll("#div1 p");
  alert(ps.length);
}
</script>
```

Dans le cas où on a un formulaire ; on peut utiliser la méthode `getElementsByName()` ; elle permet d'accéder à un ou plusieurs éléments HTML par l'attribut Name (comme l'attribut class) mais on a dit et insister qu'en HTML5 l'attribut Name n'est utiliser que dans un formulaire.

Exercices d'application : TP3.

Accès aux éléments Radios et checkbox en JavaScript :

L'accès à ces éléments HTML peut être par l'un de leurs attributs :

Exemple 1 :

The image shows a web form with a title 'Genre'. Inside a box, there are two radio buttons. The first is labeled 'Masculin' and the second is labeled 'Féminin'. Below these, there is a button labeled 'Afficher'.

Le code HTML et CSS 3 est :

```
<!DOCTYPE html>
<html>
<head>
  <title>Hello!</title>
  <style>
    fieldset
    {
      width:20%;
    }
    input[type='button']
    {
      margin-left: 150px;
    }
  </style>
</head>
<body>
<form>
<fieldset>
<legend> Genre</legend>
  <label><input type="radio" name="rb" />&nbsp;&nbsp;&nbsp;&nbsp; 
  Masculin</label>
  <br></br>
  <label><input type="radio" name="rb"/>&nbsp;&nbsp;&nbsp;&nbsp; 
  Féminin</label>
  <br></br>
  <input type="button" value="Afficher"/>
</fieldset>
</form>
</body>
</html>
```

- Accès par l'attribut name :


```
script>
    function Afficher()
    {
        var t=document.getElementsByName("rb");
        alert("Le nombre de radio dans le groupe "+t[0].name+" est
:"+t.length);
    }

</script>
```

Le résultat est :

Le nombre de radio dans le groupe rb est :2

En modifiant le code HTML comme suit :

```
<label><input type="radio" value=" Masculin " name="rb" />&nbsp;&nbsp;&nbsp; Masculin</label>  
</br></br>  
<label><input type="radio" value=" Feminin " name="rb"/>&nbsp;&nbsp;&nbsp; Féminin</label>
```

- **Accès par l'attribut checked :**

```
<script>
    function Afficher()
    {
        var t=document.getElementsByName("rb");
        for( var i in t)
            if(t[i].checked)
                alert("La case coché est : "+t[i].value);
    }
</script>
```

- **Accès par l'attribut value :**

```
<script>
function Afficher()
{
    var t=document.getElementsByName("rb");
    genre="Féminin";
    for( var i in t)
        if(t[i].checked && t[i].value=="Masculin")
            genre="Masculin" ;
    alert("Le genre est:"+genre);
}
```

</script>

- **Accès par l'attribut type :**

```
<script>
    function Afficher()
    {
        var t=document.getElementsByTagName("input");
        //afficher le nombre de input : qui est 3.
        alert(t.length);
        rd=0;
        for( var i in t)
            if(t[i].type=="radio")
                rd++;
        alert("Le nombre de radio est:"+rd+"/"+t.length);
    }
</script>
```

Remarque : les événements associés à radios sont : onclick ; onchange

Example :

Le code HTML :

[illegible]

Le code JavaScript :

```
<script>
function Afficher()
{
    var val="";
    var t=document.getElementsByName("rb");
    for(var i in t)
    if(t[i].checked)
        val=t[i].value;
}
```

```

    alert("la case coché est :"+val);
}
</script>

```

Vous pouvez utiliser le code HTML suivant avec l'événement **Onchange** :

```

<fieldset>
<legend> Genre</legend>
<label><input type="radio" name="rb" value="Masculin"
onchange="Afficher()"/>&nbsp;&nbsp;&nbsp; Masculin</label>
</br></br>
<label><input type="radio" name="rb" value="Féminin"
onchange="Afficher()"/>&nbsp;&nbsp;&nbsp; Féminin</label>
</br></br>
</fieldset>

```

Exemple2 : Accès aux cases à cocher.

Montant des Jeux

☐ Natation(200DH)

☐ Foot(300DH)

☐ Tennis(250DH)

Montant : 000

Afficher

Le code HTML :

```

<form>
<fieldset>
<legend>Montant des Jeux </legend>
<label><input type="checkbox" value="200" />&nbsp;&nbsp;&nbsp;
Natation(200DH)</label>
</br></br>
<label><input type="checkbox" value="300" />&nbsp;&nbsp;&nbsp;
Foot(300DH)</label>
</br></br>
<label><input type="checkbox" value="250" />&nbsp;&nbsp;&nbsp;
Tennis(250DH)</label>
</br></br>
<label>Montant :</label><label id="mt">000</label>
</br></br>

```

```
<input type="button" value="Afficher" onclick="Afficher()" />
</fieldset>
</form>
```

Le code JavaScript :

```
<script>
function Afficher()
{
    s=0;
    var t= document.querySelectorAll("input[type='checkbox']");
    var mt=document.getElementById("mt");
    for(var i in t)
        if(t[i].checked)
            s=s+parseInt(t[i].value);
            mt.innerHTML=s+ " DH";
    }
</script>
```

Ou bien :

```
mt.textContent=s+ " DH";
```

- Accès aux éléments HTML de type select :

Soit le code HTML suivant :

```
<select id="sel">
</select>
```

Remplir la liste par des valeurs :

Exemple 1:

```
<script>
//remplir la liste
var sel=document.getElementById("sel");
s="";
s=s+"<option>HTML</option>";
s=s+"<option>CSS</option>";
s=s+"<option>JS</option>";
sel.innerHTML=s;
</script>
```

Exemple 2:

Remplir une liste par l'ordre des mois d'une année :

```
<script>
//remplir la liste par l'ordre des mois
var sel=document.getElementById("sel");
s="";
for( var i=1;i<=12;i++)
s=s+"<option>"+i+"</option>";
```

```
sel.innerHTML=s;
</script>
```

Exercice :

Ecrire le code HTML, CSS et JavaScript de la page Facebook.

S'inscrire

X

C'est rapide et facile.

Prénom

!

Nom de famille

Numéro de mobile ou e-mail

Nouveau mot de passe

Date de naissance ?

30

▼

déc

▼

1995

▼

Genre ?

Femme

☐

Homme

☐

Personnalisé

☐

En appuyant sur S'inscrire, vous acceptez nos [Conditions générales](#), notre [Politique d'utilisation des données](#) et notre [Politique d'utilisation des cookies](#).
Vous recevrez peut-être des notifications par texto de notre part et vous pouvez à tout moment vous désabonner.

S'inscrire

L'accès au contenu de la liste peut être par :

Soit le code HTML suivant :

```
<label>Sélectionner un Mois : </label><select id="sel"></select>  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
<input type="button" value="Afficher" onclick="Afficher()"/>~  
<label>La Valeur sélectionnée est :</label>~  
<label id= "res"></label>
```

Qui donne :

Sélectionner un Mois : La Valeur sélectionnée est :

Soit le code JavaScript qui permet de remplir l'ordre des mois :

```
<script>
//remplir la liste par l'ordre des mois
var sel=document.getElementById("sel");
```

```
s="";
for( var i=1;i<=12;i++)
s=s+"<option value="+i+">"+i+"</option>";
sel.innerHTML=s;
</script>
```

- L'attribut Value : Afficher la valeur sélectionnée :

```
<script>
//remplir la liste par l'ordre des mois
var sel=document.getElementById("sel");
s="";
for( var i=1;i<=12;i++)
s=s+"<option value="+i+">"+i+"</option>";
sel.innerHTML=s;
function Afficher()
{
    //afficher la valeur de la liste ; utiliser value
    var res=document.getElementById("res");
    res.innerHTML=sel.value;
}
</script>
```

Remarque :

Si les options n'ont pas d'attribut value ; la valeur de l'option est la valeur de l'attribut value par défaut.

Exemple :

```
<script>
//remplir la liste par l'ordre des mois
var sel=document.getElementById("sel");

s="";
for( var i=1;i<=12;i++)
s=s+"<option>"+i+"</option>";
sel.innerHTML=s;
function Afficher()
{
    //afficher la valeur de la liste ; utiliser value
    var res=document.getElementById("res");
    res.innerHTML=sel.value;
}
</script>
```

En exécutant le code la valeur affichée est celle de la valeur de l'option sélectionnée.

Sélectionner un Mois : 3 ▼ Afficher La Valeur sélectionnée est : 3

L'événement associé à la balise `<select>` est `onchange`.

[illegible]

Exercices d'application TP4.

Les événements en JavaScript :

Comme on a vue ci-dessus ; on peut utiliser les attributs d'événements comme Onclick , onchange....

Ou on peut utiliser **onclick** :

```
<script>

    b1=document.getElementById("b1");
    b1.onclick = function()
    {
        document.body.style.backgroundColor = "red";
    }

</script>
```

Ou utiliser : **addEventListener** :

```
<script>
    b1=document.getElementById("b1");
    b1.addEventListener("click",function(){
        document.body.style.backgroundColor = "red";
    });
</script>
```

Le JavaScript et le CSS3

Introduction :

Avec le JavaScript, on peut modifier les styles Css3 d'un élément HTML ; pour se faire ; il suffit de cibler un élément HTML par le code :

```
Elément.style.propriété=valeur_propriété.
```

Exemple 1 :

```
<p id="p">Je suis un paragraphe </p>
<input type="button" value="Modifier" onclick="Modifier()"/>
```

Le code JavaScript pour changer la couleur de <p> est :

```
<script>
function Modifier()
{
    var p=document.getElementById("p");
    p.style.color="red";
}
</script>
```

A vous de jouer :

- Changer le remplissage en couleur : **yellow**.
- Appliquer la bordure **solid 1px red** pour le paragraphe
- Masquer le paragraphe.
- Rendre invisible ce paragraphe.

Exemple 2 :

Ecrire le code HTML qui permet d'afficher l'interface suivante :

Le code HTML est :

```
<fieldset>
<legend>Genre</legend>
<form>

    <label><input type="radio" name="rd" value="M" onclick="Afficher()"/>
    Marié</label><br/><br/>
```



```

<div id="nne">
<label> Choisir le nombre d'enfant </label> <input type="number"
id="ne"/>
</div>
<label><input type="radio" name="rd" value="C" onclick="Masquer()"/>
Célibataire</label>

<div> <br/>
<input type="button" value="Calculer" onclick="Calculer()"/>
<br/><br/>
<label>La prime est :</label> <label id="prime"></label>
</div>

</form>
</fieldset>

```

Ecrire le code style de nee.

- Ecrire le code qui permet d'afficher le nombre d'enfant si on clique sur l'option **Marié** ; et le masquer dans le cas contraire.

Le code CSS est :

```

<style>
div
{
margin-left:80px;
}
fieldset
{
width: 40%;
}
#nne
{
display:none;
}
</style>

```

Le code JavaScript :

```

<script>
var nne=document.getElementById("nne");
function Afficher()
{
nne.style.display="block";
}
function Masquer()

```

```
{  
  nne.style.display="none";  
}  
</script>
```

Refaire le code en utilisant le code HTML suivant :

```
<label><input type="radio" name="rd" value="M" onclick="changer()"/>  
Marié</label><br/><br/>  
<div id="nne">  
<label> Choisir le nombre d'enfant </label> <input type="number"  
id="ne"/>  
</div>
```

- Utiliser la méthode `changer ()` après un clique pour afficher ou masquer le `nne`.

Exercices d'application.

Passage des données entre les pages web

Utilisation de la propriété localStorage :

localStorage est une propriété de Window qui permet d'enregistrer des données dans le navigateur d'une façon permanente ; ces données peuvent être accédées à partir des autres pages Web.

Exemple :



The image shows a simple login form. It has a label 'Utilisateur :' followed by a text input field. Below that is a label 'Mot de passe :' followed by a password input field. At the bottom is a button labeled 'Se Connecter'.

C'est une boîte de connexion vers la page page1.html.

Le Code HTML : **Page Index.html**

```
<label>Utilisateur :<input type="text" id="login"/></label> <br/> <br/>
<label>Mot de passe :<input type="password" id="pwd"/></label> <br/>
<br/>
<input type="button" Value="Se Connecter" onClick="Afficher()"/>
```

Le Code HTML : **Page page1.html**

```
<h3>Votre compte est :</h3>
<label>Utilisateur :<span id="login"></span></label> <br/><br/>
<label>Mot de passe: :<span id="pwd"></span></label>
```

Le code JavaScript de index.html :

```
function Afficher()
{
    var login=document.getElementById("login");
    var pwd=document.getElementById("pwd");
    localStorage.setItem("login",login.value);
    localStorage.setItem("pwd",pwd.value);
    open("page1.html");
}
```

```
}
```

Le code JavaScript de page1.html :

```
<script>

var login1=localStorage.getItem("login");

var pwd1=localStorage.getItem("pwd");

var login=document.getElementById("login");

var pwd=document.getElementById("pwd");

login.innerHTML=login1;

pwd.innerHTML=pwd1;

</script>
```

Remarque :

- Si la localStorage n'a pas de valeur alors sa valeur est **null**.
- Pour supprimer une variable de localStorage on utilise :

```
localStorage.removeItem("login") ;
```

- Pour modifier une variable de localStorage on utilise :

```
pwd1=localStorage.getItem("pwd");//refaire l'opération
```

Test : Fermer le navigateur et le rouvrir puis ouvrir la page page1.html.

Utilisation de la propriété sessionStorage :

Cette propriété permet de stocker les données pour une session ; les données seront perdues à la fermeture du navigateur.

Les méthodes utilisées sont :

```
sessionStorage.setItem("login",login.value);
```

```
var login1=sessionStorage.getItem("login");
```

Refaire le même exemple ci-dessus en utilisant **sessionStorage**.

Utilisation des cookies:

Le cookie est un fichier qui permet de conserver des informations dans le navigateur de l'utilisateur et qui peuvent être servies d'une manière simple ; C'est une chaîne de caractère bien structurée.

Création de cookies :

```
function Afficher()
{
    var login=document.getElementById("login");
    var pwd=document.getElementById("pwd");
    //création de cookie
    document.Cookie="login="+login.value+";"+"pwd="+pwd.value;
}
```

Récupération de cookies :

```
function Recuperer()
{
    var chaine=document.Cookie;
    var t=chaine.split(";");
    alert(t[0]);
    var login=t[0].substring(6);
    alert(login);
    var pwd=t[1].substring(4);
    alert(pwd);
}
```

Passage des données d'une page à une autre par cookie :

```
document.Cookie="login="+login.value+";"+"pwd="+pwd.value+";"+"path=
/";
```

Date d'expiration du cookie :

```
temps=new Date()+9500000;
document.Cookie="login="+login.value+";"+"pwd="+pwd.value+";"+"expires="+temp+";"+"path="/;
open("page1.html");
```

Autres méthodes :

La méthode `setTimeout()` :

C'est une méthode qui appelle une fonction après un nombre de millisecondes.

Exemple :



Le code HTML :

```
<div>

<label id="I">0</label> <br/> <br/>

<input type="button" value="Démarrer" onclick="demarrer()" />

</div>
```

Le code JavaScript :

```
<script>
function demarrer()
{
    var x= setTimeout(compter,1000);
}
function compter()
{
    I=document.getElementById("I");
    I.innerHTML="Je suis Affiché après 1 s" ;
}
</script>
```

```
</script>
```

Remarque :

Pour interrompre l'exécution de la méthode `setTimeout()` on utilise la méthode `clearTimeout()`.

```
clearTimeout(x);
```

La méthode `setInterval()` :

C'est une méthode qui déclenche une fonction pendant un intervalle de temps régulière (c'est une répétition de `setTimeout()`).



Le code JavaScript est :

```
<script>
  var x=0;
  function demarrer()
  {
    var x= setInterval(compter,1000);
  }
  function compter()
  {  x++;
    l=document.getElementById("l");
    l.innerHTML=x ;
  }
</script>
```

Exemple :

Le code HTML :

```

<div>
<label >Heure</label>: <label >Minute</label>:<label
>Seconde</label><br/> <br/>
<label id="l1">0</label>: <label id="l2">0</label>:<label
id="l3">0</label><br/> <br/>
</div>

```

Le code JavaScript est :

```

<script>
l1=document.getElementById("l1");
l2=document.getElementById("l2");
l3=document.getElementById("l3");
var t=new Date();
s=t.getSeconds();
l3.innerHTML=s;
var x= setInterval(seconde,1000);
function seconde()
{
s++;
l3.innerHTML=s;

}
</script>

```

A vous de jouer et compléter le code.

Remarque :

On peut stopper l'action de setInterval() par clearInterval :

```
clearInterval (x)
```

Exemple 2 :

Créer un chronomètre en inverse et afficher la page page1.html lorsque le chrono revient à 0.

Le code HTML :

```
<p>0</p>
```

Le code CSS :

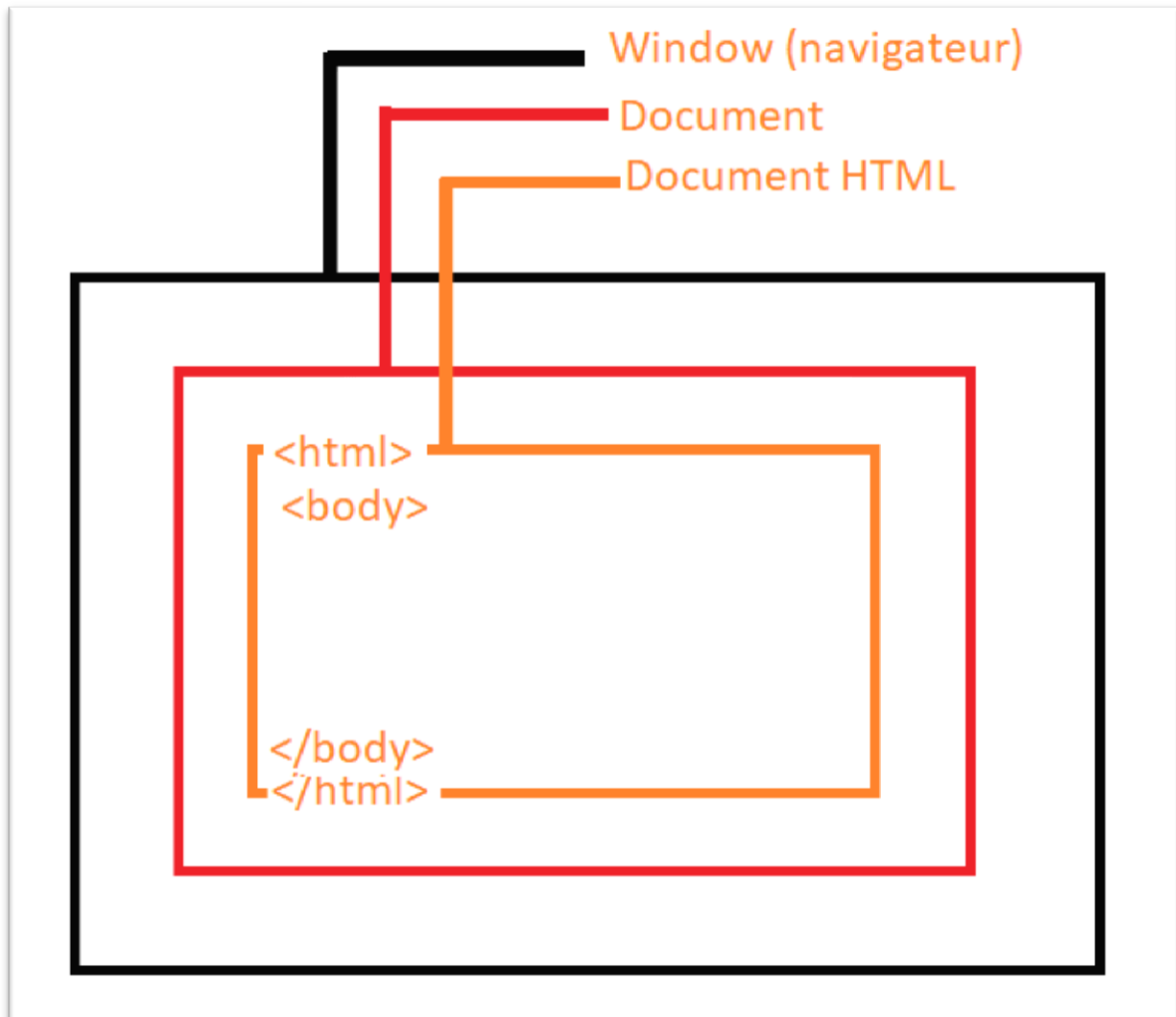

```
p
{
  background: rgb(170, 204, 30) ;
  color:red;
  font-size:30px;
  height:80px;
  width:80px;
  border-radius: 40px;
  text-align:center;
  display:table-cell;
  vertical-align: middle;
}
```

Le code JavaScript :

```
<script>
p=document.getElementById("p");
var x=p.innerHTML;
var z=setInterval(chrono,1000);
function chrono()
{
  if(parseInt(x)>0)
  {
    x=parseInt(x)-1;
    p.innerHTML=x;
  }
  if(x=="0")
  {
    open("page1.html");
    clearInterval(z) ;
  }
}
</script>
```

Les propriétés de la fenêtre Window.

La structure d'une fenêtre Window.



Exemple :

Le code HTML :

```
<h2>Les dimension de la fenêtre</h2>

  <label>La hauteur de la fenêtre:<span id="h">0</span></label>
<br/><br/>

  <label>La largeur de la fenêtre:<span id="l">0</span></label>
<br/><br/>

<h2>Les dimension du navigateur</h2>
```

```
<label>La hauteur du navigateur:<span
id="h1">0</span></label> <br/><br/>
```

```
<label>La largeur du navigateur:<span id="l1">0</span></label>
<br/><br/>
```

Une fenêtre est caractérisée par les propriétés suivantes :

Le code JavaScript :

```
<script>
    var h=document.getElementById("h");
    var l=document.getElementById("l");
    var h1=document.getElementById("h1");
    var l1=document.getElementById("l1");
    //la hauteur de window
    h.innerHTML=window.innerHeight+" px";
    //la largeur de window
    l.innerHTML=window.innerWidth+" px";
    h1.innerHTML=window.outerHeight+" px";
    //la largeur de navigateur
    l1.innerHTML=window.outerWidth+" px";

    window.onresize=function()
    { //la hauteur de window
        h.innerHTML=window.innerHeight+" px";
        //la largeur de window
        l.innerHTML=window.innerWidth+" px";
        // la hauteur du navigateur (y compris les barres)
        //la hauteur de navigateur
        h1.innerHTML=window.outerHeight+" px";
        //la largeur de navigateur
        l1.innerHTML=window.outerWidth+" px";
    }
</script>
```

Les méthodes d'un Document

- La méthode write() : Document.write(" Je suis un texte ")
- La méthode open() :open("page1.html", "_blank ", "width=500 height=800")
- La méthode close() :this.close().

•