

# Développement Web - INF04

JavaScript

Mussab Zneika  
CY-Tech

# Introduction(1)

- JavaScript (également appelé JS) est un langage de programmation que vous pouvez utiliser pour ajouter des fonctions intéressantes et importantes à vos pages Web.
- Parfois, il est également appelé langage de script (JavaScript) avec l'implication qu'il est en quelque sorte plus facile de scripter que de programmer.
- JavaScript a été créé par Netscape.
- JavaScript n'est pas Java, les deux sont comparables en syntaxe, mais JavaScript vous permet d'incorporer des fonctionnalités simples directement dans votre code HTML, plutôt que d'écrire du matériel à compiler comme avec JAVA.

# Introduction(2)

- Le Javascript est un langage "de script" simplifié orienté objet
- Javascript permet de rendre dynamique un site internet développé en HTML:
  - Validation de formulaires, calculs, messages,
  - Modification de la page web,
  - Communication avec un serveur directement (AJAX)
- Javascript est standardisé par un comité spécialisé, l'ECMA (European Computer Manufacturers Association).
- Javascript est Supporté par la plupart des navigateurs web.

# HTML et JavaScript

## Intégration de javascript dans la page html:

- Directement dans le fichier HTML:
  - `<script>` utilisé pour charger le code JavaScript
  - Type Propriété avec une valeur "texte / javascript"

```
<script type="text/javascript">
```

```
... code javascript ici
```

```
</script>
```

- Dans un fichier externe:
  - La propriété SRC spécifie le chemin du fichier JS

```
<script src="unFichier.js" type="text/javascript">
```

```
</script>
```

# Insertion dans une page HTML

- Dans le corps de la page HTML

- Entre les balises <body> et </body>.
- Code exécuté lors du chargement de la page.

```
<body>  
  <script type="text/javascript" src="foo.js">  
  </script>  
</body>
```

- Dans l'entête de la page

- Entre les balises <head> et </head>.
- Code exécuté pour l'événement d'un utilisateur
- L'événement se trouve dans le corps du document.

# Types Primitifs

- En JavaScript, il y a 3 types de base:
  - **Boolean**
    - ✓ 2 constantes **true**, **false**
    - ✓ opérateurs : négation **!**, et logique **&&**, ou logique **||**
  - **Number**
    - ✓ Pas de séparation nette entre entiers et flottants
    - ✓ opérateurs : **+**, **-**, **\***, **/** (division flottante), **%** (reste de la division)
  - **String**
    - ✓ les chaînes se notent entre **"** ou **'** : **"exemple"**, **'un autre'**
    - ✓ opérateurs de concaténation : **+**

# Les opérateurs

Les opérateurs	Mode
+, -, *, /, %	Actes numériques
, &&, !	Actes logiques
==, !=, <, >, <=, >=	Comparaison
++, --	Augmentation, diminution
+	concaténation
=, +=, -=, *=, /=, %=	Affectation

# Variables

- Déclaration et affectation

- Déclaration avec le mot clé "var"
- Affectation avec le signe d'égalité (=)
- Une variable doit être déclarée avant d'être utilisée.
- Les variables déclarées sont contraintes dans le contexte d'exécution dans lequel elles sont déclarées.
- Les variables non-déclarées sont toujours globales

```
var i, j, k;  
i = 1;  
var prix = 0;  
var nbr = 10;  
var fl = 3.141;  
var str1 = "L etoile";  
var lien = '<a href="index.htm">Home</a>';  
var a, b=a=9;  
var a=b, b=9; //undefined
```



# Variables

- Déclaration et affectation

- *let* permet de déclarer une variable dont la portée est celle du bloc courant, éventuellement en initialisant sa valeur.
- *let* peut parfois permettre de rendre le code plus lisible lorsqu'on utilise des fonctions internes.

```
let x = 1;
var y=5;
if (x === 1) {
  let x = 2;
  y=2;
  alert(x); // output: 2
  alert(y); // output: 2
}
alert(x); // output: 1
alert(y); // output: 2
```

# Variables

- Contraintes concernant les noms de variables :
  - Les noms de variables ne peuvent contenir que des lettres, chiffres, ou le caractère "\_" (underscore)  
`var Mon_Prenom; // correct`
  - Les caractères spéciaux et accentués sont interdits :  
`var Mon_Prénom; // incorrect`
  - Attention aux majuscules et minuscules :  
`MonPrenom` est différent de `Monprenom`.
  - Un nom de variable ne peut contenir d'espaces.  
`var Mon Prenom; // incorrect`
  - Les mots réservés JavaScript ne peuvent être utilisés comme noms de variable.

# Portée des variables

- Globale :
  - Variable déclarée en début de script.
  - Accessible à n'importe quel endroit du programme.
- Locale :
  - Variable déclarée à l'intérieur d'une fonction.
  - Accessible uniquement dans la fonction.

JavaScript

```
var message = 'Ici la variable globale !';  
function showMsg() {  
  var message = 'Ici la variable locale !';  
  alert(message); }  
showMsg();           // On affiche la variable locale  
alert(message);      // Puis la variable globale
```

# Conversions de types

- Une chaîne des caractères dont les caractères représentent un nombre est convertie en nombre.
- Dans un expression avec l'opérateur + c'est la conversion vers chaîne qui l'emporte
- **NaN** : *Not a Number* valeur de conversion pour toute expression qui ne peut être convertie en un nombre peut se tester avec fonction **isNaN**.

```
"9" *3; // -> 27  
"100" -2; // -> 98  
"99"+5 // -> "995"  
"deux" *3; // -> NaN  
isNaN ("deux" *3) ; // -> true
```

JavaScript

# Conversions de types

**String():** La fonction String () convertit la valeur d'un objet à une chaîne.

- **Number():** La fonction Number () convertit l'argument d'objet en un numéro qui représente la valeur de l'objet.
- **parseInt(string, base)** :analyse une chaîne de caractère fournie en argument et renvoie un entier exprimé dans une base donnée.
- **parseFloat (String):**analyse une chaîne et renvoie un nombre réel.

```
var str1= String(33.33); // str1 est "33.33"  
var num1=6.6; var str2=num1.toString(); //str2 est "6.6"  
var num2= Number(str1); // num2 est 33.33  
var num3= str1 - 0; // num3 now is 33.33  
var num4= parseInt(str1); // num4 now is 33  
var num5= parseFloat(str1); // num5 now is 33.33  
var num6=Number("555 444");// NaN  
var num7= Number("4rt");//NaN
```

JavaScript

# parseInt et parseFloat

- Convertissent une chaîne en nombre (entier ou réel)
- Seul le premier nombre dans la chaîne est retourné, les autres caractères sont ignorés
- Si le premier caractère ne peut être converti en un nombre, le résultat sera NaN
- Les espaces en tête sont ignorés

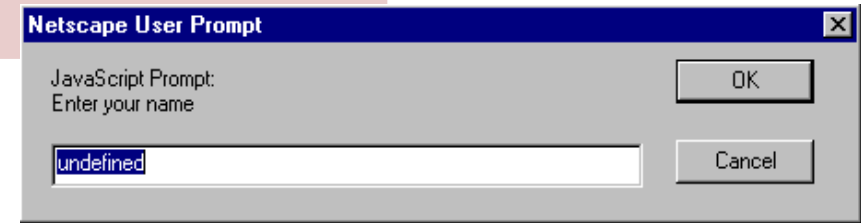
```
parseFloat(" 1.24 "); // -> 1.24
parseInt("42"); // -> 42
parseInt("42 est la reponse "); // -> 42
parseInt(" 42 est la reponse "); // -> 42
parseInt("4rt "); // -> 4
parseInt("555 444"); // -> 555
parseInt(" reponse = 42"); // -> NaN
```

JavaScript

# Entrées/sorties

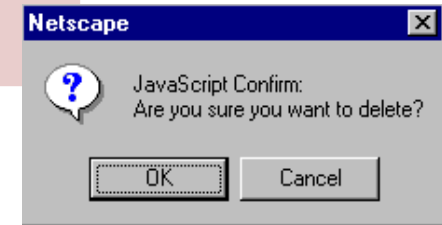
## prompt()

Ouvre une boîte de dialogue avec une zone saisie et 2 boutons OK et Annuler



## confirm()

Ouvre une boîte de dialogue avec 2 boutons OK et Annuler, retourne un booléen



## alert()

Permet d'ecrire un message dans une fenêtre



# Entrées/sorties

---

## Exercice 1

- Ecrire un script qui affiche le message " Bonjour World !" dans une boîte de dialogue.
- Améliorer le script pour qu'il permette de demander le Prénom de l'utilisateur et afficher « Bonjour 'votre prénom' ».



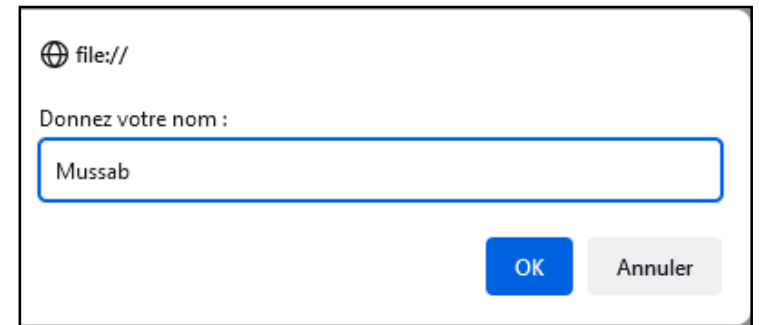
# Entrées/sorties

## Exercice 1: solution

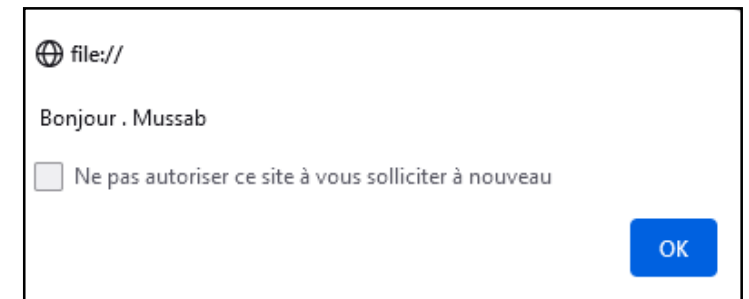
JavaScript

```
<script type="text/javascript">  
var a = prompt("Donnez votre nom : ");  
alert(" Bonjour . "+a);  
//document.write("Bonjour... "+a);  
</script>
```

## Le Résultat



A screenshot of a JavaScript prompt dialog box. The title bar shows a globe icon and the text "file://". The main text says "Donnez votre nom :". Below it is a text input field containing the name "Mussab". At the bottom right, there are two buttons: "OK" (blue) and "Annuler" (grey).



A screenshot of a JavaScript alert dialog box. The title bar shows a globe icon and the text "file://". The main text says "Bonjour . Mussab". Below it is a checkbox with the label "Ne pas autoriser ce site à vous solliciter à nouveau". At the bottom right, there is a single "OK" button (blue).

# Entrées/sorties

---

## Exercice 2

- Ecrire un script qui demande à l'utilisateur de saisir deux nombres entiers puis affiche leur somme.

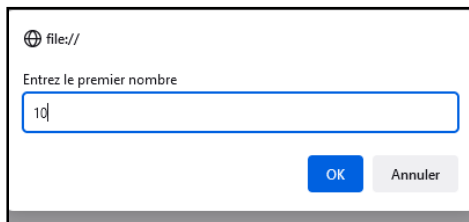
# Entrées/sorties

## Exercice 2: solution

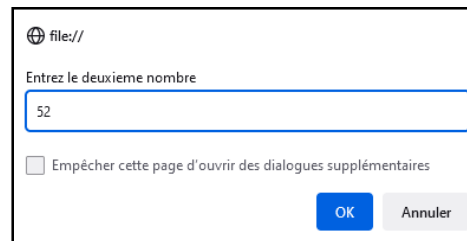
JavaScript

```
var a = prompt("Entrez le premier nombre");  
var b = prompt("Entrez le deuxieme nombre");  
var somme = parseInt(a) + parseInt(b);  
alert("La somme de "+a+" et "+b+" egal à "+ somme);  
//document.write("La somme de "+a+" et "+b+" egal à "+  
somme);
```

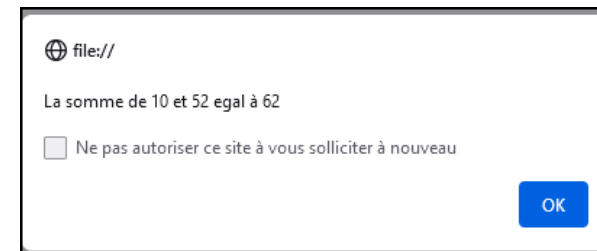
### Le Résultat



A browser window showing a file:// address bar. The main content area displays the text "Entrez le premier nombre" above a text input field containing the number "10". At the bottom right, there are two buttons: "OK" (blue) and "Annuler" (grey).



A browser window showing a file:// address bar. The main content area displays the text "Entrez le deuxieme nombre" above a text input field containing the number "52". Below the input field is a checkbox labeled "Empêcher cette page d'ouvrir des dialogues supplémentaires". At the bottom right, there are two buttons: "OK" (blue) and "Annuler" (grey).



A browser window showing a file:// address bar. The main content area displays the text "La somme de 10 et 52 egal à 62". Below this text is a checkbox labeled "Ne pas autoriser ce site à vous solliciter à nouveau". At the bottom right, there is a single button: "OK" (blue).

# Les conditions

- **if (conditions) {} else if (conditions ) else{}**
  - la partie **else** n'est pas obligatoire
  - **false, 0, "", NaN, null, undefined** valent **false**,
  - tout le reste vaut **true**
  - Égalité : **==, !=**
  - inférieur, supérieur : **=<, >=, >, <**
  - Identique à : **===, !==** (teste valeur et type)  
`('1' == 1) // true`  
`('1' === 1) // false`

# Les conditions

---

## Exercice 3

- Demander le nom de l'utilisateur au chargement de la page. Demander ensuite une confirmation du nom de l'utilisateur. Si le nom est confirmé, afficher ce dernier dans une boîte de dialogue

# Les conditions

## Exercice 3: solution

JavaScript

```
var a = prompt("Donnez votre nom : ");  
var b = prompt("Confirmez votre nom : ");  
if(a == b) {  
    alert("Le nom est : "+a);  
}  
else {  
    alert("Error");  
}
```

# Les conditions

---

## Exercice 4

- Ecrire un script permettant de résoudre une équation du second degré :  $ax^2+bx+c=0$ . Ce script devra demander à l'utilisateur les valeurs de  $a$ ,  $b$  et  $c$  ; puis donner le résultat.

# Les conditions

## Exercice 4: solution

JavaScript

```
var a, b, c, root_part, denom, root1, root2;
a = prompt("What is the value of 'a'? \n", "1");
b = prompt("What is the value of 'b'? \n", "");
c = prompt("What is the value of 'c'? \n", "");
// Compute the square root and denominator of the
result root_part = Math.sqrt(b * b - 4.0 * a * c);
denom = 2.0 * a;
root1 = (-b + root_part) / denom;
root2 = (-b - root_part) / denom;
if (isNaN(root1)) {
    alert("No real roots !");
}

else{
    alert("The first root is: "+ root1);
    alert("The second root is: "+root2);
}
```



# Les Boucles

- Boucle for :  
    for (i=0; i<5; i++) {...}
- Boucle while :  
    while (test) {...}
- Boucle Do , while  
    do {...} while (test);

# Les Boucles : For

- Syntaxe comme en C, C++, Java...
- Il répète les instructions dans le corps selon une condition
- L'initialisation s'exécute en premier.
- Si la condition est fausse, c'est fini.
- Si la condition est vraie, les instructions du corps s'exécutent. Alors le corps exécute le pas.
- La condition est vérifiée à nouveau, et ainsi de suite.

```
for (initialisation ; conditions; étape )  
{  
    corps  
}
```

# Les Boucles : While

- Syntaxe comme en C, C++, Java...
- Il répète les instructions dans le corps selon une condition.
- La condition est vérifiée en premier
- Si la condition est fausse, c'est fini.
- Si la condition est vraie, les instructions du corps s'exécutent. Alors le corps exécute le pas.
- La condition est vérifiée à nouveau, et ainsi de suite.

```
while (conditions ) {  
    corps  
}
```

# Les Boucles : Do, While

- Syntaxe comme en C, C++, Java...
- Il répète les instructions dans le corps selon une condition.
- **Initialement, le corps est exécuté une fois**
- **La condition est vérifiée**
- Si la condition est fausse, c'est fini.
- Si la condition est vraie, les instructions du corps s'exécutent. Alors le corps exécute le pas.
- La condition est vérifiée à nouveau, et ainsi de suite.

```
do{  
    corps  
}while(conditions);
```

# Break

- Syntaxe comme en C, C++, Java...
- Peut apparaître dans l'un des contrôles suivants :  
**for, while, do... while, switch**
- Il arrête le flux et continue immédiatement après
- Plus de répétitions après **break**

JavaScript

```
var i;  
for(i=0;i<6;i++){  
  if(i==4){ break;}  
  document.write(i+"<br>");  
  
}  
document.write("i after the for "+i+"<br>");
```

Le Résultat

```
0  
1  
2  
3  
i after the for 4
```

# Continue

- Syntaxe comme en C, C++, Java...
- Peut apparaître dans l'un des contrôles suivants :  
**for, while, do... while**
- Il arrête le flux et continue de vérifier la condition.
- D'autres répétitions peuvent se produire après **continue**.

JavaScript

```
var i;  
for(i=0;i<6;i++){  
  if(i==4){ continue;}  
  document.write(i+"<br>");  
  
}  
document.write("i after the for "+i+"<br>");
```

## Le Résultat

```
0  
1  
2  
3  
5  
i after the for 6
```

# Les Boucles

---

## Exercice 5

- Ecrire une page HTML faisant apparaître la table de multiplication pour les entiers compris entre 1 et 9 et présenter le résultat dans un tableau [10x 10].

# Les Boucles

## Exercice 5: solution

JavaScript

```
document.write("<center>");
/* Mettre un titre pour la table */
document.write("<caption> Table de multiplication</caption>");
/* Créer la table*/
document.write("<table border=4>");
/* Créer la première ligne */
document.write("<tr><td>","x","</td>");
for(var i=1;i<=9;i++)
    document.write("<td>",i,"</td>");
for(var i=1;i<10;i++){
    /* Créer la ligne numéro i*/
    document.write("<tr>");
    document.write("<td>",i,"</td>");
    for(var j=1;j<=9;j++){
        /* Créer la colonne numéro j*/
        document.write("<td>",i*j,"</td>");
    }
    document.write("</tr>");
}
document.write("</table>");
document.write("</center>");
```

### Le Résultat

x	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81



# Fonctions

- Le mot-clef **function** permet de définir une donnée de type fonction
- On précise entre parenthèses les **paramètres formels**, séparés par des virgules
- Le corps de la fonction est noté entre accolades

```
function( param1, param2, ...) {  
    ... corps de la fonction  
}
```

# Fonctions

- Le mot-clef **function** permet de définir une donnée de type fonction
- On précise entre parenthèses les **paramètres formels**, séparés par des virgules
- Le corps de la fonction est noté entre accolades

**function**( *param1*, *param2*, ...) {

*... corps de la fonction*

```
<script type="text/javascript">
function fac(n) {
  if (n < 2) {
    return 1; }
  else{
    return n * fac(n-1); }
  }
var x=fac(5);
document.write(x);
</script>
```

JavaScript

# Fonctions

- Pas de type dans la signature de la fonction
- La déclaration d'arguments est optionnelle
- On peut accéder aux arguments via la variable "arguments"

```
function test(argument1) {  
  document.write( " "+ argument1);  
  document.write( " "+arguments.length);  
  for (var i=0 ; i<arguments.length; i++) {  
    document.write(" " +arguments[i]);  
  }  
  document.write("<br>");  
}  
test(); // undefined-0  
test("5"); // 5 1 5  
test("4","5"); // 4 2 4 5
```

JavaScript

# Fonctions

---

## Exercice 6

- Créez une fonction qui fait la somme d'un nombre indéterminé d'entiers.
- Écrivez un script qui va tester cette fonction.

# Fonctions

## Exercice 6: solution

```
function sumAll() {  
  var index; var sum = 0;  
  for(index = 0; index < arguments.length; ++index) {  
    sum += arguments[index];  
  }  
  return sum;  
}  
document.writeln(sumAll(a, b, c) + "<br>") //17  
document.writeln(sumAll(b, d) + "<br>") //30  
document.writeln(sumAll() + "<br>") //0
```

JavaScript

# Tableaux

- Un tableau permet de stocker plusieurs valeurs de même type, comme par exemple plusieurs entiers dans une seule variable.
- En javascript : la notion de **tableaux** n'est pas conventionnelle par rapport `a la plupart des langages.

## L'objet Array:

il permet de représenter des tableaux mais qui sont aussi des collections (listes, piles, files)

Met à disposition des méthodes **join()**, **reverse()**, **sort()**, **push()**, **pop()**, etc.

# Tableaux

## Création

- Constructeur `Array()`:  
`var tab = new Array (10) ; // tableau de 10 éléments`
- Expression littérale:  
`var tabVide = [];`  
`var couleur = [" trefle ", " carreau ", " coeur ", " pique "];`
- Ce qui est moins classique
  - ✓ les éléments d'un tableaux ne sont pas typés  
`var tab = [12 , " timoleon ", true , 0, " autre "];`
  - ✓ les tableaux sont dynamiques

## Lecture/écriture des éléments

- Premier élément : indice 0  
`couleur [0]; // -> " trefle "`  
`tab [0]; // -> undefined`  
`tab [1] = 3;`
- Propriété `length`  
`tab.length ;`

# Tableaux

## Parcourir un tableau

- Boucle **for**: Une boucle simple qui incrémente un indice et accède aux éléments successifs par cet indice.

```
tableau_entiers = new Array(56,12,43,4,9);  
tableau_entiers = [56,12,43,4,9];  
for (i=0;i<tableau_entiers.length;i=i+1)  
    document.writeln(tableau_entiers[i]);
```

- Boucle **for in** : la boucle for in est l'équivalent de la boucle foreach du PHP

```
tableau_entiers = new Array(56,12,43,4,9);  
tableau_entiers = [56,12,43,4,9];  
for (var i in tableau_entiers)  
    document.writeln(tableau_entiers[i]);
```

- Itération avec for in ne parcourt que les éléments « définis »



# Tableaux

## Tableaux multi-dimensionnels

- il s'agit de tableaux de tableaux
- chacun des tableaux éléments peut avoir sa propre « taille »

### Example1:

```
var tab= [[5 , 7], [1, 4], [12 ,6]];
tab. length ; // -> 3
tab [0][0]; // -> 5
tab[1]. length ; // -> 2
tab[0]; // -> [5 ,7]
var ttt = [[ true ], ["a", "b", "c"], [10 , -4]];
```

### Example2:

```
a = new Array(4);
for (i=0;i<4;i=i+1) {
  a[i] = new Array(4);
  for (j=0;j<4;j=j+1)
    a[i][j] = i+j;
}
```

# Tableaux

## Les tableaux associatifs

### Création 1:

```
var tableau = new Array();  
tableau["un"] = "La premiere chaîne ";  
tableau["deux"] = "La deuxième chaîne ";  
tableau["tnt"] = "Plein d'autres chaînes";  
tableau["un"]  
tableau.length
```

### Example 1:

```
var famille = {pere : 'robisson', mère : 'gilda', soeur :  
'Laurence', frère : 'Ludovic', cousin1: 'Guillaume', cousin2:  
'Pauline', cousin3: 'Clarisse' } ;  
for (var id in famille) {  
    document.write(id + ' : ' + famille[id]+"<br>");  
}
```

# Tableaux

## Opérations sur les tableaux

<code>concat()</code>	Regroupe (concaténer) plusieurs tableaux et rend un nouveau tableau
<code>join(delimiter)</code>	Rend une chaîne contenant les éléments du tableau séparés par le délimiteur spécifié
<code>pop()</code>	Retourne le dernier élément du tableau et le supprime du tableau
<code>push(e1,e2, ... )</code>	Ajoute les éléments e1, e2, ... à la fin du tableau et retourne la nouvelle longueur.
<code>reverse()</code>	Renverse le tableau.
<code>shift()</code>	Retourne le premier élément du tableau et le supprime du tableau.
<code>slice(begin[,end])</code>	Retourner un nouveau tableau constitue d'une sous-partie d'un tableau existant.
<code>sort([function])</code>	Trie le tableau selon l'ordre lexicographique (tri ascendant).
<code>splice(index,n,[,el1,el2 ])</code>	Ajouter/Enlever des éléments d'un tableau.
<code>unshift(e1,e2,...)</code>	Ajoute les éléments e1,e2, ... en tête du tableau et retourne la longueur du nouveau tableau.
<code>valueOf()</code>	Retourne une chaîne décrivant les valeurs initiales du tableau.

# Tableaux

## Opérations sur les tableaux

- **Exemple 1 :**

```
var tab= ['A', 'B'];  
tab.push('C'); // Ajoute «C» à la fin du tableau  
tab.push('D', 'Guillaume'); // Ajoute «D» et « Guillaume » à la fin du tableau  
tab.shift(); // Retire « A»  
tab.pop(); // Retire « Guillaume »  
tab.unshift('T','H'); // Ajoute «T» et « H » à en tête du tableau
```

- **Exemple 2 :**

```
var array1 = ['a', 'b', 'c'];  
var array2 = ['d', 'e', 'f'];  
var array3= array1 .concat(array2));  
array3.slice(2, 4);  
array3.sort();  
var array4= [1, 30, 4, 21];  
array4.sort();
```

# Tableaux

## Exemples

Ajouter des éléments dans un tableau (push,unshift):

JavaScript

```
var prenoms=['Mussab',  
'Marc','julian','Naya'];  
var p = ' ';  
//On ajoute les valeurs Sarra et Francia en  
fin de tableau  
prenoms.push('Sarra', 'Francia');  
//On ajoute deux éléments en début de  
tableau  
var taille = prenoms.unshift('Flora',  
'claire');  
//On affiche les valeurs de notre tableau  
for(var i = 0; i < prenoms.length; i++)  
{ p += 'Prénom n' + (i+1) + ' : ' +  
prenoms[i] + '\n'; }  
//On affiche la taille du tableau et les  
prénoms  
alert('Le tableau contient ' +taille+ '  
éléments : \n\n' +p);
```

### Le Résultat

Le tableau contient 8 éléments :

Prénom n1 : Flora  
Prénom n2 : claire  
Prénom n3 : Mussab  
Prénom n4 : Marc  
Prénom n5 : julian  
Prénom n6 : Naya  
Prénom n7 : Sarra  
Prénom n8 : Francia

# Tableaux

## Exemples

Supprimer des éléments dans un tableau (pop,shift):

JavaScript

```
var prenom=['Mussab', 'Marc', 'julian',  
'Naya'], p = '';  
//On supprime le premier prénom du tableau  
var supprp = prenom.shift();  
// On supprime le dernière prénom du  
tableau  
var supprd = prenom.pop();  
//On récupère les valeurs de notre tableau  
for(var i = 0; i < prenom.length; i++){  
p += 'Prénom n°' + (i+1) + ' : ' +  
prenom[i] + '\n'; }  
//On affiche les prénoms du tableau et le  
prénom supprimé  
alert('Nouveau tableau :\n' +p+ '\n\nPrénom  
supprimé : \n' +supprp+'\n'+supprd);
```

### Le Résultat

Nouveau tableau :  
Prénom n°1 : Marc  
Prénom n°2 : julian

Prénom supprimé :  
Mussab  
Naya

# Tableaux

## Exemples

Ajouter des éléments choisis dans un tableau (splice())

JavaScript

```
var prenom=[ 'Mussab', 'Marc', 'julian',  
            'Naya'], p = '';  
//On ajoute les valeurs Claire et Anne à  
partir de l'index 2  
prenom.splice(2, 0, 'Claire',  
              'Anne' );  
for(var i = 0; i < prenom.length; i++){  
p += 'Pré  
nom n' + (i+1) + ' : ' + prenom[i] + '\n'; }  
//On affiche la taille du tableau et les  
prénoms  
alert('Le tableau contient ' + ' éléments :  
\n\n' +p);
```

## Le Résultat

Le tableau contient éléments :

Prénom n1 : Mussab  
Prénom n2 : Marc  
Prénom n3 : Claire  
Prénom n4 : Anne  
Prénom n5 : julian  
Prénom n6 : Naya

# Tableaux

## Exemples

Supprimer des éléments choisis dans un tableau (Splice())

JavaScript

```
var prenom=['Mussab', 'Marc', 'julian',  
'Naya'], p = '';  
//On supprime deux valeurs e à partir de  
l'index 2  
prenom.splice(2, 2);  
  
for(var i = 0; i < prenom.length; i++){  
p += 'Prénom n' + (i+1) + ' : ' +  
prenom[i] + '\n'; }  
//On affiche la taille du tableau et les  
prénoms  
alert('Le tableau contient ' + ' éléments :  
\n\n' +p);
```

### Le Résultat

Le tableau contient éléments :

Prénom n1 : Mussab  
Prénom n2 : Marc



# Tableaux

## Exemples

Trier les éléments d'un tableau (sort)

JavaScript

```
var prenom=['Mussab', 'mussab','Marc',  
'julian', 'Naya'], p = '';  
//On classe dans  
l'ordre alphabétique  
prenom.sort() ;  
//On récupère les valeurs de notre tableau  
for(var i = 0; i < prenom.length; i++){  
  p += 'Prénom n°' + (i+1) + ' : ' +  
  prenom[i] + '\n'; }  
//On affiche les prénoms du tableau  
alert('Nouveau tableau :\n\n' +p);  
//console.log(p);
```

### Le Résultat

Nouveau tableau :

Prénom n°1 : Marc  
Prénom n°2 : Mussab  
Prénom n°3 : Naya  
Prénom n°4 : julian  
Prénom n°5 : mussab

# Tableaux

## Exemples

Trier les éléments d'un tableau (sort)

JavaScript

```
var nums=[10,20,3, 30,4, 1,35,98];  
nums.sort();  
document.write(nums);
```

Le Résultat

1,10,20,3,30,35,4,98

JavaScript

```
var nums=[10,20,3, 30,4, 1,35,98];  
nums.sort( function (a,b){return a-b;});  
document.write(nums);
```

Le Résultat

1,3,4,10,20,30,35,98

# Tableaux

## Exemples

### Trier les éléments d'un tableau (sort)

JavaScript

```
var prenom=['Mussab', 'mussab','Marc',  
'julian', 'Naya'], p = '';  
//On classe dans  
l'ordre alphabétique  
prenom.sort() ;  
prenom.reverse() ;  
//On récupère les valeurs de notre tableau  
for(var i = 0; i < prenom.length; i++){  
  p += 'Prénom n°' + (i+1) + ' : ' +  
  prenom[i] + '\n'; }  
//On affiche les prénoms du tableau  
alert('Nouveau tableau :\n\n' +p);  
//console.log(p);
```

### Le Résultat

Nouveau tableau :

Prénom n°1 : mussab

Prénom n°2 : julian

Prénom n°3 : Naya

Prénom n°4 : Mussab

Prénom n°5 : Marc

# Tableaux

---

## Exercice7

Ecrire un programme JavaScript pour trouver l'élément le plus fréquent d'un tableau.

Exemple `tab= [3, 'B', 'B', 'B', 2, 3, 4, 3, 'a', 2, 4, 9, 3,3,3];`  
sortie: 3 (6 fois)

# Tableaux

## Exercice 7 : Solution

JavaScript

```
var tab=[3, 'a', 'a', 'a', 2, 3, 'a', 3, 'a', 2, 4, 9, 3,3,3,3,3];
var maxf = 0, f = 0;
var item;
for (var i=0; i<tab.length; i++) {
  for (var j=i; j<tab.length; j++) {
    if (tab[i] == tab[j]) f++;
    if (maxf<f) {
      maxf=f; item = tab[i];
    }
  }
  f=0;
}
document.write(item+" ( " +maxf + "
times ) " ) ;
```

Le Résultat

3 ( 8 times )

# Tableaux

---

## Exercice 8

- Ecrire un programme qui calcule la moyenne d'un ensemble de notes. le programme demande à l'utilisateur de saisir le numéro des notes et, puis ouvrir une fenêtre pour entrer chacune des notes. Quand toutes les notes sont saisies, une fenêtre affiche la moyenne.

# Tableaux

## Exercice 8 : Solution

JavaScript

```
var note = new Array();
var moyenne = 0;
var nb;
nb = prompt("Entrer le nombre de notes");
nb = parseInt(nb, 10);
if(isNaN(nb)) alert("error nombre de notes
doit etre entier");
else{
    for (var i=0;i<nb;i++){
        note[i] = prompt("Entrer la note "
+ (i+1));
        note[i] = parseInt(note[i], 10);
        moyenne += note[i];
    }
    moyenne = moyenne / nb;
    alert("La moyenne est " + moyenne);
}
```

### Le Résultat

Entrer le nombre de notes

Entrer la note 1

☐ Empêcher cette page d'ouvrir des dialogues supplémentaires

Entrer la note 2

☐ Empêcher cette page d'ouvrir des dialogues supplémentaires

La moyenne est 12.5

☐ Ne pas autoriser ce site à vous solliciter à nouveau

# Tableaux

---

## Exercice 9

Ecrire une fonction JavaScript pour obtenir le premier n élément d'un tableau.

$f([1,2,7,8],2) = 1,2$

$f([1,2,7,8],3) = 1,2,7$



# Tableaux

## Exercice 9 : Solution

```
function f(array, n)
{ if (array == null)
  return void 0;
if (n == null) return array[0];
if (n < 0) return [];

return array.slice(0, n);
};
```

JavaScript

```
document.writeln("f([1,2,7,8],2)=  
"+f([1,2,7,8],2)+"<br>");  
document.writeln("f([1,2,7,8],3)=  
"+f([1,2,7,8],3));
```

### Le Résultat

```
f([1,2,7,8],2)= 1,2  
f([1,2,7,8],3)= 1,2,7
```

# Objects

- Utilise pour stocker des données. Création d'une variable (de type Object) avec new Object() ou de manière littérale (énumération entre accolades).

## Création d'un objet

- Similaire aux tableaux associatifs
- Séparation par des virgules ","
- On crée un seul objet

Création directe	Création par clonage
<pre>var obj = {   monAttribut: "valeur",   maMethode: function(p)   { alert("parametre : " + p);} }; alert(obj.monAttribut); obj.maMethode("test");</pre>	<pre>var obj = {   monAttribut: "valeur",   maMethode: function(p) {     alert("parametre : " + p);} }; <b>var obj2=Object.create(obj);</b> alert(obj2.monAttribut);</pre>

# Objects

## Création d'un objet

### Example 1 :

```
var person = new Object();  
person.name = "Julien";  
person.age = 23;
```

### Example 2 :

```
function personne(nom,prenom) {  
  this.nom = nom;  
  this.prenom = prenom;  
}  
p = new personne("Dupont","Jean")
```

### Example 3 :

```
function bureau(numero,proprietaire) {  
  this.numero = numero  
  this.proprietaire = proprietaire  
}  
p = new personne("Dupont","Jean");  
b = new bureau("117",p);  
document.writeln("<P>" + toutvoir(p,"p"));  
document.writeln("<P>" + toutvoir(b,"b"));
```

### Example 4 :

```
p = new personne("Dupont","Jean");  
b = new bureau("117",p);  
document.writeln("<P>" + toutvoir(p,"p")  
);  
document.writeln("<P>" + toutvoir(b,"b")  
);  
p2 = new personne("Durand","Jeanne");  
p2.sexe = "F";
```

# Objects

## Example :

```
<script type="text/JavaScript">
function Livre (titre, auteur, publie){
this.titre= titre;
this.auteur= auteur;
this.publié= publie;
this.display = function () {
document.writeln("titre:"+this.titre+" &nbsp; &nbsp; &nbsp;
auteur:"+this.auteur+" &nbsp; &nbsp; &nbsp; publié:"+this.publie);
};
}
var livre1 = new Livre ("Le Chien des Baskerville", "Sir Arthur Conan
Doyle ", 1901);
livre1.display ();
document.writeln(typeof liver1);
</script>
```

# Objects

## Exercise 10

Supposons que vous ayez un objet défini comme suit:

```
var person = {  
  name: "James ",  
  birth: 1989,  
  ship: {  
    name: "USS Enterprise",  
    number: "NCC-1701",  
    commissioned: 2245  
  },  
  getInfo: function() {  
    return this.name + " commands the " + this.ship.name;  
  }  
};
```

Ecrire un programme javascript pour afficher toutes les valeurs de ses propriétés et méthodes.

# Objects

## Exercise 10 : Solution

JavaScript

```
for (var prop in person)
{
    if(typeof person[prop]=="object")
    {var obj=person[prop];
    for (var prop1 in obj){

        document.writeln(obj[prop1]+"<br>");
    }
}

else{
    document.writeln(person[prop]+"<br>"
    );}
}
```

### Le Résultat

```
James
1989
USS Enterprise
NCC-1701
2245
function() { return this.name + " commands the " + this.ship.name; }
```

# Objects

## Objets et égalité

JavaScript

```
var livrel= {  
  title: "Javascript",  
  author: "USP",  
  published: 1999, };  
  
var livre2= {  
  title: "Javascript",  
  author: "USP",  
  published: 1999, };  
  
if (livrel == livre2)  
  { document.write("Les deux livres sont  
  les mêmes");  
  }  
else {  
  document.write("Les deux livres sont  
  différents");  
  }
```

### Le Résultat

Les deux livres sont différents

# Objects

## Objets et égalité

JavaScript

```
function Livre (titre, auteur, publié){  
  this.titre= titre;  
  this.auteur= auteur;  
  this.publié= publié; };  
  
var livre1 = new Livre ("Le Chien des  
Baskerville", "Sir Arthur Conan Doyle ", 1901);  
  
var livre2 = new Livre ("Le Chien des  
Baskerville", "Sir Arthur Conan Doyle ", 1901);  
if (livre1 == livre2) {  
  document.write("Les deux livres sont les  
  mêmes");  
}  
else {  
  document.write("Les deux livres sont  
  différents auuuu");  
}
```

Le Résultat

Les deux livres sont différents



# SOURCES D' INFORMATION

- ▶ MDN Web Docs: JavaScript: <https://developer.mozilla.org/fr/docs/Web/JavaScript>
- ▶ W3C JavaScript and HTML DOM Reference: <https://www.w3schools.com/jsref/>
- ▶ Cours de Thierry Lecroq  
*<http://www-igm.univ-mlv.fr/~lecroq/cours/javascript.pdf>*
- ▶ Cours de Jean-Loup Guillaume JAVASCRIPT  
*[http://jlguillaume.free.fr/www/documents/teaching/ntw1213/LI385\\_C4\\_Javascript.pdf](http://jlguillaume.free.fr/www/documents/teaching/ntw1213/LI385_C4_Javascript.pdf)*