

## 1. le langage JavaScript

JavaScript est le langage de programmation le plus populaire au monde, il définit le comportement de la page web et toutes interactions avec l'utilisateur.

JavaScript peut changer:

- Le contenu HTML,
- Les valeurs d'attribut HTML,
- Les styles CSS,
- Cacher ou afficher des éléments HTML,

### 1.1.Insertion du code Javascript

Dans une page HTML, Le code JavaScript est inséré entre les balises `<script>` et `</script>` dans n'importe quelle section du document.

Exemple

```
<script>
```

```
document.getElementById("label1").innerHTML = "mon nouveau html";
```

```
</script>
```

### 1.2.Opérateurs [1]

Opérateur	Explication	Symbole(s)	Exemple
Addition	Utilisé pour ajouter deux nombres ou concaténer deux chaînes.	+	6 + 9; "Bonjour " + "monde !";
Soustraction, multiplication, division	Les opérations mathématiques de base.	-, *, /	9 - 3; 8 * 2; 9 / 3;
Assignation	On a déjà vu cet opérateur : il affecte une valeur à une variable.	=	myVariable = '123';
Égalité	Teste si deux valeurs sont égales	==	If(i==j)...

### 1.3.Quelques fonctions de base

3.1. **getElementById** : Permet la sélection d'un élément HTML en utilisant son attribut « id »

3.2. **alert** : permet d'afficher un message à l'écran.

Exemple : `alert("Bonjour")` ;

3.3. **parseInt** : permet de convertir une variable contenant une chaîne de caractère à une variable de type entier.

Exemple : `i = parseInt("123")` ;

3.4. accéder au style d'un élément `document.getElementById("element_id").style`

Exemple : `document.getElementById("div1").style.display="none"`;

### 1.4.Les fonctions

La fonction dans le langage JavaScript est définie avec le mot-clé **function**, suivi d'un nom, suivi de parenthèses (). Les noms de fonction peuvent contenir des lettres, des

chiffres, des traits de soulignement et des signes dollar (mêmes règles que pour les variables). Les parenthèses peuvent inclure des noms de paramètres séparés par des virgules: (paramètre1, paramètre2, ...) Le code à exécuter, par la fonction, est placé entre des accolades: { }

Exemple :

```
Function nomDeFonction (paramètre1, paramètre2, paramètre3)
{
    // code à exécuter
}
```

Dans la définition de fonction, les paramètres sont listés entre parenthèses (). Les arguments de fonction sont les valeurs reçues par la fonction lorsqu'elle est appelée. À l'intérieur de la fonction, les paramètres se comportent comme des variables locales.

### 1.5. Invocation de fonction

Les fonctions sont appelées :

- Lorsqu'un événement se produit comme le clique d'un bouton ou l'utilisation du clavier.
- Lorsqu'elle est appelée directement à partir d'un autre code Javascript.
- Ou automatiquement comme dans le cas de chargement de page.

Les fonctions calculent souvent une valeur de retour qui est renvoyée appelant.

Exemple : Calculez le produit de deux nombres et renvoyez le résultat

```
var x = produit (4, 3); // La fonction 'produit' est appelée, et la valeur de retour sera affectée à x
```

```
function produit (a, b)
{
    return a * b; // La fonction renvoie le produit de a et b
}
```

### 1.6. Définition d'objet

En JavaScript, on déclare un objet en listant ses propriétés entre des accolades comme montré par l'exemple suivant :

```
var personne = {nom: "Larbi", prenom: "Mohamed", age: 23};
```

l'accès au propriété des objets se fait de deux manières différentes en utilisant la syntaxe : objet.nomChamp, ou objet['nomChamp'].

Exemple

```
X=personne["prenom"] ; //x recevra la valeur du prénom de l'objet personne.
```

### 1.7. Méthodes d'objets

Les fonctions peuvent être déclarées à l'intérieur des définitions des objets.

Exemple :

```
var personne =
{
  nom:"Larbi",
  prenom:"Mohamed",
  age:23,
  nomComplet : function ()
  {
    return this.nom + " " + this.prenom; // le mot clé 'this' fait référence
    au "propriétaire" de la fonction
  }
};
```

On accède à la fonction d'un objet en utilisant la syntaxe `Objet.methode()`. Par exemple, pour l'objet déclaré en dessus, on peut récupérer le nom complet d'un objet en utilisant l'instruction :

```
x= personne.nomCopmlet();
```

### 1.8.Événements JavaScript

Un événement HTML peut être une action du navigateur ou une action d'un utilisateur. Ci-dessous quelques exemples d'événements HTML:

- Une page Web HTML a fini de se charger
- Un champ de saisie HTML a été modifié
- Un bouton HTML a été cliqué
- ...

Event	Description
onchange	Un élément HTML a été modifié
onclick	L'utilisateur clique sur un élément HTML
onmouseover	L'utilisateur déplace la souris sur un élément HTML
onmouseout	L'utilisateur éloigne la souris d'un élément HTML
onkeydown	L'utilisateur appuie sur une touche du clavier
onload	Le navigateur a fini de charger la page

Exemple d'utilisation : la date sera affichée lors d'un clic sur le bouton html :

```
<button onclick="displayDate();">l'heure est?</button>
```

### 1.9.Instructions de flux de contrôle

#### 1.9.1. L'instruction conditionnelle 'if'

La syntaxe générale des instructions est:

```
if (cond) {traitement;}
```

```
else {traitement;}
```

### **1.9.2. L'instruction switch**

Une instruction switch ressemble à ceci:

```
switch («expression»)
```

```
{«corp de switch »}
```

Exemple :

```
switch(val)
{
    case 1: return 'Dimanche';
    Case 2: return 'Lundi';
    Case 3: return 'Mardi';
}
```

### **1.9.3. La boucle While**

La boucle 'while' a la syntaxe suivante:

```
while («condition»)
```

```
{«Traitement ;»}
```

### **1.9.4. La boucle do-while**

La boucle 'do-while' fonctionne à peu près de la même manière, mais elle vérifie son état après chaque boucle, pas avant.

Syntaxe :

```
do { Traitement ;}
```

```
while (condition)
```

### **1.9.5. la boucle 'for'**

La boucle 'for' a la syntaxe suivante:

```
for («initialisation»; «condition»; «post_iteration»)
```

```
{
«traitement»
}
```