

**Département Informatique**

**LP-R2SI**

# **Technologies de Web :**

## **Chapitre 5 : JavaScript**

---

**Aimad QAZDAR**

**a.qazdar@uca.ma**

**8 novembre 2019**

# Plan

---

## Introduction

---

---

<b>Partie 1</b>	<b>HTML et JavaScript</b>
	<b>I/O de données avec JavaScript</b>
	<b>Structure d'un script en JavaScript</b>
	<b>Les variables en JavaScript</b>
	<b>Les instructions répétitives</b>
	<b>Les instructions conditionnelles</b>
	<b>Les fonctions</b>

---

<b>Partie 2</b>	<b>Les objets</b>
	<b>Les événements</b>
	<b>Les cookies</b>

---

# Introduction

- Javascript permet de rendre un site internet développé en HTML **dynamique**.
- Javascript permet de développer de véritables applications fonctionnant exclusivement dans le cadre d'Internet.
- Le Javascript est un langage de script simplifié basé sur les objets dont la syntaxe est basée sur celle du Java.
- Javascript a été initialement élaboré par Netscape en association avec Sun Microsystem.
  - Plus tard, Microsoft développera son propre langage Javascript officiellement connu sous le nom de JScript.

# Introduction

- Contrairement à un applet Java qui est un programme compilé, les **scripts écrits en Javascript sont interprétés**
  - Le **Java**, représenté par un ou plusieurs fichiers autonomes dont l'extension sera **\*.class** ou **\*.jar**, est invoqué par une balise HTML spécifique
  - Le **JavaScript** est écrit directement au sein du document HTML sous forme d'un **script** encadré par des balises HTML spéciales.
- Javascript est standardisé par un comité spécialisé, l'ECMA (European Computer Manufactures Association).

# HTML et JavaScript

- La page HTML devra TOUJOURS contenir les deux balises spécifiques et indispensables

```
<script language="JavaScript">  
    code de votre script  
</script>
```

- Le code JavaScript s'intègre de deux manières avec le code HTML :
  - **Insertion directe dans le code HTML**
    - Le code JavaScript s'insère le plus souvent dans la page HTML elle même.
    - C'est la méthode la plus simple et la plus fréquemment utilisée par les développeurs de sites Internet.
  - **Insertion comme un module externe**

# HTML et JavaScript

- Insertion dans une page HTML pour exécution directe
  - Le code s'exécute automatiquement lors du chargement de la page HTML dans le navigateur en même temps que le contenu de la page HTML s'affiche à l'écran.

```
<html>
  <head>
    <title>..... </title>
  </head>
  <body>
    <script language="JavaScript">
      alert('bonjour') ;
    </script>
  </body>
</html>
```

# HTML et JavaScript

- **Insertion par appel de module externe**

- On peut insérer du code JavaScript en faisant appel à un module externe se trouvant à n'importe quelle adresse (URL).

```
<script src="URL du module externe">  
    .....  
</script>
```

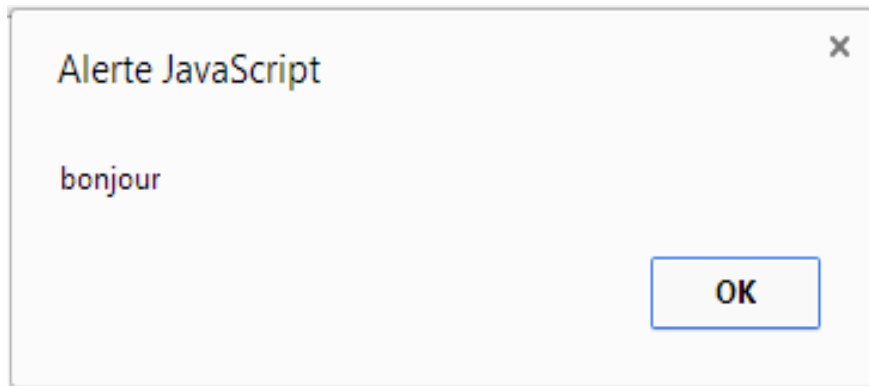
- Les deux balises de Javascript doivent être placés entre :
  - Les Tags <body> ... </body> dans le cas d'une exécution directe
  - Les Tags <head> ... </head> de la page HTML pour une exécution différée.

# Entrée et sortie de données avec JavaScript

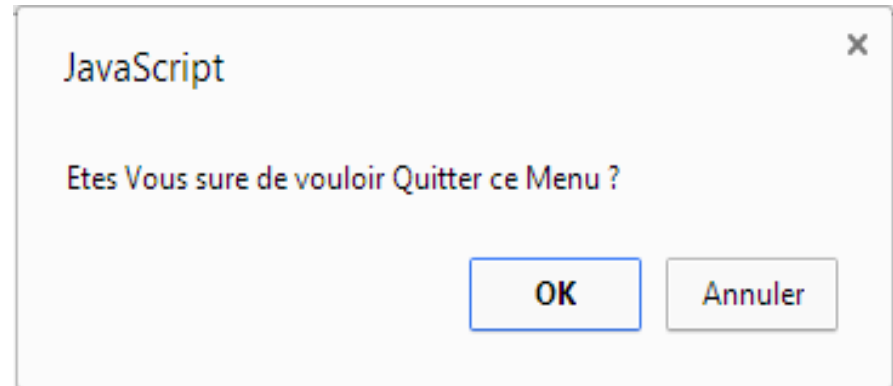
- Il existe 3 types de boîtes de messages peuvent être affichés en utilisant Javascript : **Alerte**, **Confirmation** et **Invite**
- Méthode **alert()**
  - sert à afficher à l'utilisateur des informations simples de type texte. Une fois que ce dernier a lu le message, il doit cliquer sur OK pour faire disparaître la boîte
- Méthode **confirm()**
  - permet à l'utilisateur de choisir entre les boutons OK et Annuler.
- Méthode **prompt()**
  - La méthode `prompt()` permet à l'utilisateur de taper son propre message en réponse à la question posée
- La méthode **document.write** permet d'écrire du code HTML dans la page WEB



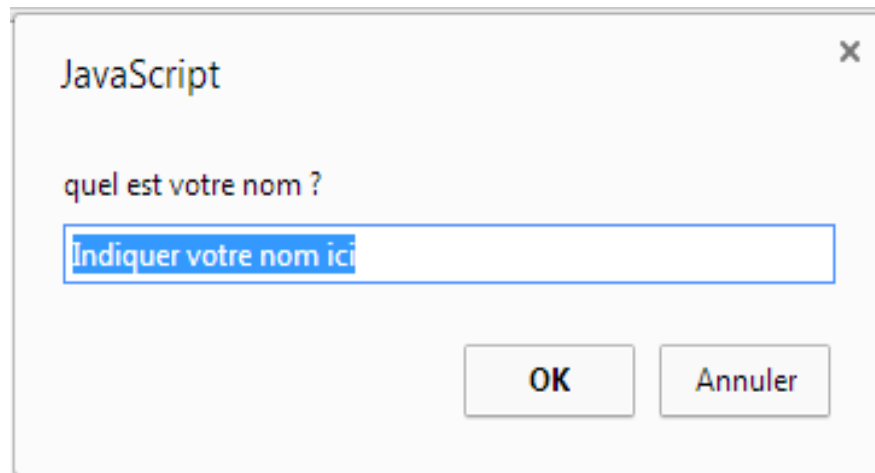
# Entrée et sortie de données avec JavaScript



**alert()**



**confirm()**



**prompt()**

# Entrée et sortie de données avec JavaScript

```
<html>
  <head>
    <title> Une page simple </title>
  </head>
  <body>
    Bonjour
    <script language='javascript'>
      alert('bonjour');
      document.write (
        prompt('Quel est votre nom ?' ,
          'Indiquer votre nom ici')
      );
      confirm('Quel bouton allez-vous choisir ?');
    </script>
  </body>
</html>
```

# La structure d'un script en JavaScript

- La syntaxe du langage Javascript s'appuie sur le modèle de Java et C
- Règles générales :
  - Chaque commande doit être terminée par un **point-virgule** (;).
  - Un nombre à virgule est séparé par un **point** (.) et non par une virgule
  - Le langage Javascript est **sensible à la casse**
  - Il existe deux méthodes permettant d'intégrer des commentaires à vos scripts.
    - Placer un double **slash** (//) devant le texte
    - Encadrer le texte par un **slash suivi d'une étoile** (/\*) et la même séquence inversée (\*/)

# Qu'est ce qu'une variable?

- Une variable est un emplacement de la mémoire de l'ordinateur qui va servir à stocker des informations.
- Ces informations vont pouvoir être utilisées plus tard.
- Une variable est définie par :
  - Le **nom**
  - Le **type**
  - La **valeur**



# Les variables : Déclaration et affectation

- Le mot-clé **var** permet de déclarer une ou plusieurs variables.
- Après la déclaration de la variable, il est possible de lui affecter une valeur par l'intermédiaire du signe d'égalité (=).
- Si une valeur est affectée à une variable sans que cette dernière ne soit déclarée, alors Javascript la déclare **automatiquement**
- La lecture d'une variable non déclarée provoque une erreur
- Une variable correctement déclarée mais dont aucune valeur n'est affectée, est indéfinie (**undefined**).

# Les variables : Déclaration et affectation

- Exemple :

```
//Déclaration de i, de j et de k.
```

```
var i, j, k;
```

```
//Affectation de 1 à i.
```

```
i = 1;
```

```
//Déclaration et affectation de prix.
```

```
var prix = 0;
```

```
//Déclaration et affectation de caractère
```

```
var caractere = ["a", "b", "c"];
```

# Les variables : Nom

- **Contraintes concernant les noms de variables :**
  - Les noms de variables ne peuvent contenir que des **lettres, chiffres, ou le caractère "\_"** (underscore)
    - ➔ **Mon\_Prenom** est un nom valide
  - Les **caractères spéciaux et accentués** sont **interdits** (é, à, ç, ï, etc..)
    - ➔ **Mon\_Prénom** n'est pas un nom valide. Il y a un caractère accentué.
  - Les **majuscules** et les **minuscules** ont leur importance.
    - ➔ **MonPrenom** est différent de **Monprenom**.
  - Un nom de variable ne peut contenir d'**espaces**.
    - ➔ **Mon Prenom** n'est pas un nom de variable correct. **Il y a un espace.**
  - Les mots **réservés JavaScript** ne peuvent être utilisés comme noms de variable.

# Les variables : Types

- **Le type d'une variable** dépend de la valeur stockée dans cette variable.
- **Pas de déclaration de type :**  
var maVariable = ' mohamed ' ;  
maVariable =10;
- Trois principaux types de valeurs
  - **String**
  - **Number** : 3 valeurs spéciales
    - Positive Infinity ou +Infinity (valeur infini positive)
    - Negative Infinity ou –Infinity (valeur infinie négative)
    - Nan (Not a Number) habituellement générée comme résultat d'une opération mathématique incohérente
  - **Boolean**
    - Deux valeurs littérales : true (vrai) et false (faux).



# Les variables : Portée

- Les variables peuvent être **globales** ou **locales**.
- Une variable **globale** est déclarée en **début** de script et est **accessible à n'importe quel endroit du programme**.
- Une variable **locale** est déclarée à **l'intérieur** d'une fonction et **n'est utilisable que dans la fonction elle-même**.

# Les opérateurs

- Deux type d'opérateurs : **Arithmétiques** et **Logiques**

- **Opérateurs Arithmétiques:**

```
Var a=2;  
var b=5;  
var addition, soustraction,  
multiplication, division;  
addition= a+b;  
soustraction = a-b;  
Multiplication = a*b;  
divison = a/b;
```

- **Opérateurs Logiques :**

```
Inférieur : <  
Supérieur : >  
Egal : =  
Supérieur ou égal : >=  
Inférieur ou égal : <=  
Différent : !=
```

# Les opérateurs sur type « String »

- **La concaténation**

```
Var chaine = « bonjour » + « FI3/FCD1 »;
```

- **Déterminer la longueur d'une chaîne**

```
Var ch1 = « bonjour »;
```

```
Var longueur = ch1.length;
```

- **Identifier le nième caractère d'une chaîne**

```
Var ch1 = « Rebonjour ! »;
```

```
Var carac = ch1.charAt(2);
```

- **Extraction d'une partie de la chaîne**

```
Var dateDuJour = « 04/04/03 »
```

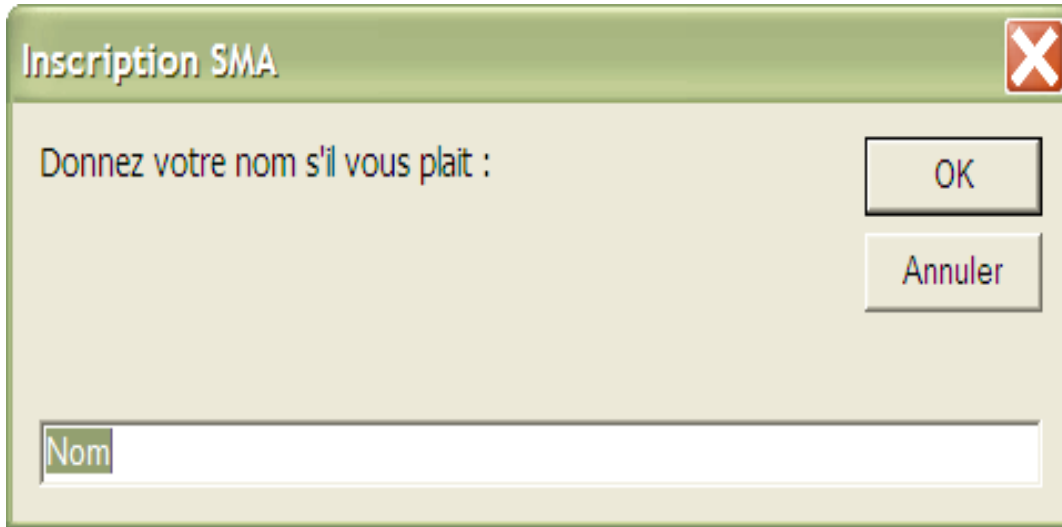
```
Var mois = dateDuJour.substring(3, 5);
```

Indice du premier caractère de la sous-chaîne à extraire

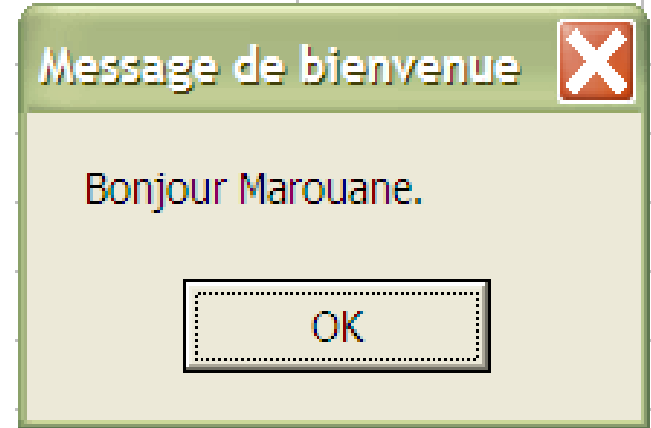
Indice du dernier caractère à prendre en considération; ce caractère ne fera pas partie de la sous-chaîne à extraire

# Exercices :

- **Exercice 1:** Réaliser les prompts suivants :



A Java dialog box titled "Inscription SMA" with a close button (X) in the top right corner. The main text reads "Donnez votre nom s'il vous plaît :". Below the text is a text input field with the placeholder text "Nom". To the right of the input field are two buttons: "OK" and "Annuler".



A Java dialog box titled "Message de bienvenue" with a close button (X) in the top right corner. The main text reads "Bonjour Marouane.". Below the text is a single button labeled "OK".

- **Exercice 2:** Réaliser les prompts suivants :  
Faire deux appels avec la fonction `prompt()` pour entrer deux variables et calculer leurs : somme, soustraction, multiplication et Division et les afficher dans un `alert()`.

# Exercices :

- **Exercice :**

1. Faire un appel de 4 prompts et stocker le résultat de chaque appel dans une variable: Le premier étant votre nom, le deuxième votre prénom, le troisième votre âge et le quatrième votre emploi.
2. Afficher dans un alert un résultat identique à celui là:

**Je m'appelle ZINEDINE Zidane**

**J'ai 43 ans**

**Je suis Entraîneur de foot**

# Les tableaux

- Un tableau est une collection **homogène** de données, **ordonnée** et de **taille statique**. Chaque élément est repéré par un indice (son rang dans le tableau).
- Exemple:

```
var tab1 = new Array();
```

```
var tab2 = new Array('Jean','Michel');
```

```
var tab3 = new Array(2);
```

```
tab3[0] = 'Jean'
```

# Instructions conditionnelles

- Condition : « Si ... alors ... »

```
if (condition) {  
    /* instructions si condition validée */  
}  
else {  
    /* instructions si condition invalidée */  
}
```

## Exemple:

```
x = prompt ("votre age?", "age");  
if ( x < 40) {  
    alert ('vous êtes jeune') ;  
}  
else {  
    alert ('vous êtes vieux') ;  
}
```

# Instructions répétitives

- Boucle « pour ... faire » :

```
for (initialisation ; condition ;  
    incrémentation ou décrémentation) {  
    /* instructions */  
}
```

- Boucle « tant que ... faire... » :

```
while (condition) {  
    /* instructions */  
}
```

- Boucle « répéter ... tant que » :

```
do {  
    /* instructions */  
} while (condition);
```



# Les fonctions

- On appelle **fonction** une partie de code recevant une ou plusieurs informations à partir desquelles elle retourne une ou plusieurs informations.
- Les informations fournies à la fonction sont appelées **arguments** ou **paramètres** de la fonction.
- Les informations renvoyées par la fonction sont appelées **résultat**.
- En JavaScript, il existe deux types de fonctions :
  - Les **fonctions propres à JavaScript**, appelées méthodes. Elles sont associées à un objet en particulier.
  - Les fonctions que **vous définissez vous-même**.

# Les fonction: déclaration et appel

- Déclaration d'une fonction :

```
Function nom_de_la_fonction(liste_des_arguments)
{
    Liste des instructions;

    return (valeur_de_retour); //Facultatif
}
```



Il est plus prudent de **placer les déclarations de fonctions dans l'en-tête <head>...</head>** pour qu'elles soient prises en compte par l'interpréteur avant leur exécution dans le corps de la page <body>...</body>

- Appel d'une fonction :

```
nom_de_la_fonction(liste_des_arguments);
```

# Les fonctions prédéfinies (1)

- **eval**

- Cette fonction permet de calculer une formule mathématique à partir d'une chaîne de caractères

```
<SCRIPT LANGUAGE="JavaScript">  
    function evaluation() {  
        document.formulaire.calcul.value=eval(document.formulaire.saisie.value);  
    }  
</SCRIPT>
```

```
<FORM NAME="formulaire">  
    Saisissez une expression mathématique :  
    <INPUT TYPE="text" NAME="saisie" MAXLENGTH=40 SIZE=40>  
    <INPUT TYPE="button" VALUE="évaluation" onClick="evaluation()">  
    <INPUT TYPE="text" NAME="calcul" MAXLENGTH=40 SIZE=40>  
</FORM>
```

# Les fonctions prédéfinies (2)

## ● isFinite

- Détermine si le paramètre est un nombre fini.
- Renvoie false si ce n'est pas un nombre ou l'infini positif ou infini négatif.

```
isFinite(240) //retourne true  
isFinite("Un nombre") //retourne false
```

## ● isNaN

- Détermine si le parametre n'est pas un nombre (NaN : Not a Number).

```
isNaN("un nombre") //retourne true  
isNaN(20) //retourne false
```

# Les fonctions prédéfinies (3)

- **parseFloat**

- analyse une chaîne de caractères et retourne un nombre décimal.
- Si l'argument évalué n'est pas un nombre, renvoie NaN (Not a Number).

```
var numero="125";  
var nombre= parseFloat(numero); //retourne le  
nombre 125
```

- **parseInt**

- analyse une chaîne de caractères et retourne un nombre entier de la base spécifiée.
- La base peut prendre les valeurs 16 (hexadécimal) 10 (décimal), 8 (octal), 2 (binaire).

```
var prix=30.75;  
var arrondi = parseInt(prix, 10); //retourne 30
```

# Les fonctions prédéfinies (4)

- **Number**

- convertit l'objet spécifié en valeur numérique

```
var jour = new Date("December 17, 1995 03:24:00");  
//convertit la date en ms  
alert (Number(jour));
```

- **String**

- convertit l'objet spécifié en chaîne de caractères

```
jour = new Date(430054663215); //Convertit le nombre  
en date Mois jour, Année etc.  
alert (String(jour));
```

- **Escape**

- retourne la valeur hexadécimale à partir d'une chaîne de caractère codée en ISO-Latin-1.

```
escape("!&") //retourne %21%26%
```