



(Cours n° 05)

2<sup>ème</sup> année Licence Informatique  
Université Abou-Bekr Belkaïd - Tlemcen  
2018 / 2019

Dr. Mahfoud Houari  
[mahfoud.houari@gmail.com](mailto:mahfoud.houari@gmail.com)

## Plan du chapitre

- 1 Introduction
- 2 Généralités
- 3 Fonctions
- 4 Tableaux
- 5 Manipulation du DOM HTML

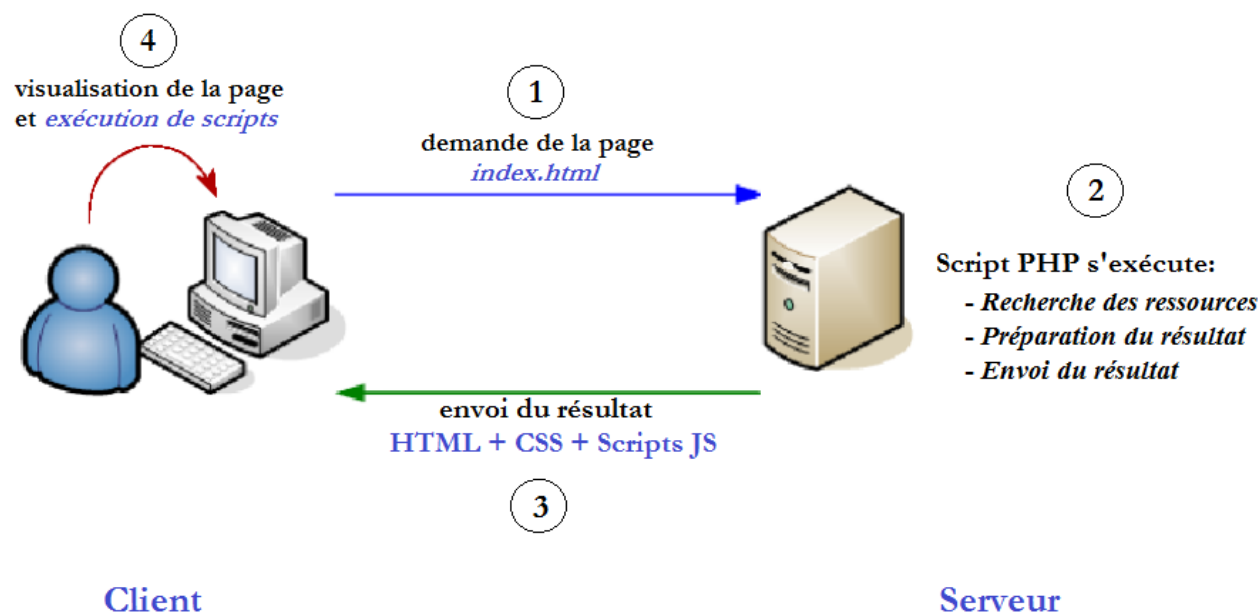
# JavaScript



# Introduction

## Qu'est-ce que c'est JavaScript ?

- Un langage de scripts interprété et orienté objet.
- Intégré au sein des pages Web et s'exécute sur le client Web. (langage de scripts côté client).
- Les scripts sont exécutés par le navigateur via un interpréteur de scripts.
- Les scripts permettent une interaction entre la page et l'utilisateur.
- Les scripts côté client peuvent être générés par un script côté serveur.



Quoi faire avec JavaScript ?

- Changer les attributs d'un élément HTML.
- Changer le style CSS d'un élément HTML.
- Cacher (faire apparaître) des éléments HTML.
- Gérer les composants d'un formulaire.
- Afficher des boîtes de dialogues (avertissement, confirmation,...).
- Gérer les événements liés à la souris et au clavier.
- Communication en temps réel ( *Web Sockets* ).
- Stockage des données locales ( *Web Storage* ).
- Des petits jeux ( *Tetris, Sudoku, Labyrinthe*,... ).
- Autres: *Manipulation des objets, Géolocalisation, Dessin*,...

# Introduction

## Où définir un script JS ?

- 1) Dans l'élément `<script>` (Cas de script interne) :

Exemple:

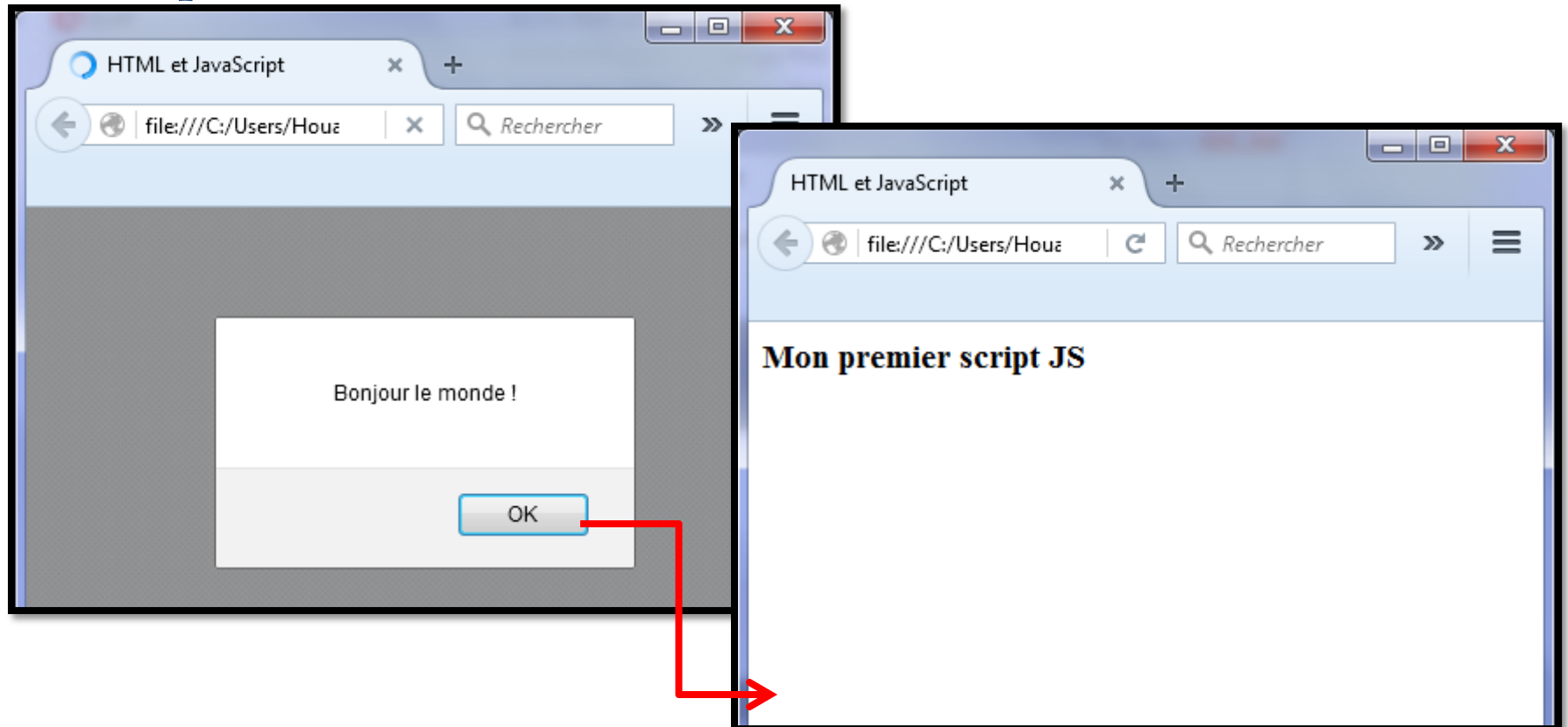
```
<html>
  <head>
    <title>HTML et JavaScript</title>
  </head>
  <script>
    alert("Bonjour le monde !");
  </script>
  <body>
    <h3>Mon premier script JS</h3>
  </body>
</html>
```

# Introduction

## Où définir un script JS ?

- 1) Dans l'élément **<script>** (Cas de script interne) :

Exemple:



# Introduction

## Où définir un script JS ?

- 1) Dans l'élément `<script>` (Cas de script interne) :

Exemple:

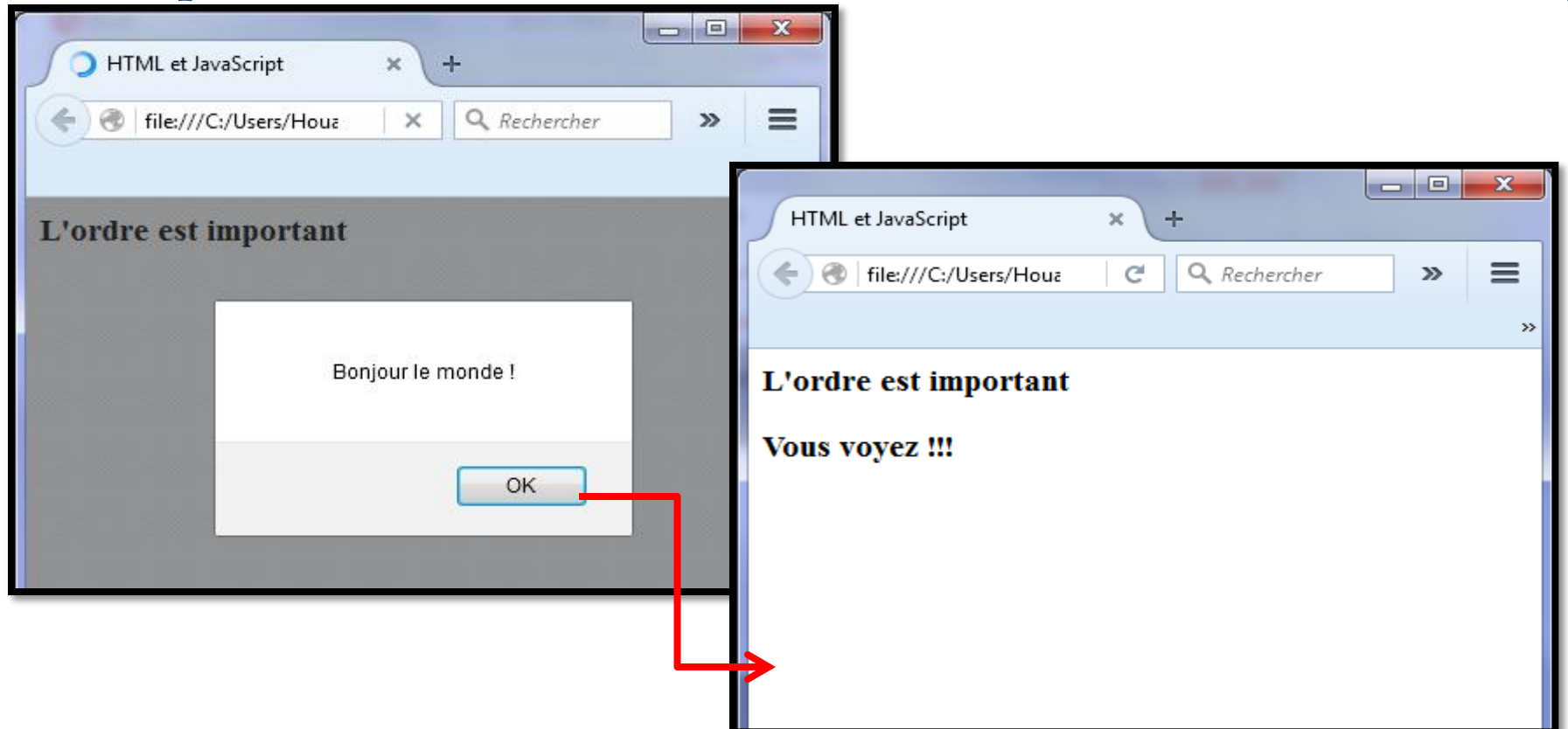
```
<html>
  <head>
    <title>HTML et JavaScript</title>
  </head>
  <body>
    <h3>L'ordre est important</h3>
    <script>
      alert("Bonjour le monde !");
    </script>
    <h3>Vous voyez !!!</h3>
  </body>
</html>
```

# Introduction

## Où définir un script JS ?

- 1) Dans l'élément **<script>** (Cas de script interne) :

Exemple:





# Introduction

## Où définir un script JS ?

2) Dans l'élément **<script>** (Cas de script externe) :

Exemple:

page.HTML

```
<html>
  <head>
    <title>HTML et JavaScript</title>
  </head>
  <script src="mon_script.JS">
  </script>
  <body>
    <h3>Cas de script externe</h3>
  </body>
</html>
```

mon\_script.JS

```
/* ceci est un commentaire JS */
alert("Ceci est un script externe");
```

# Introduction

## Où définir un script JS ?

### 3) À travers des liens :

#### Exemple:

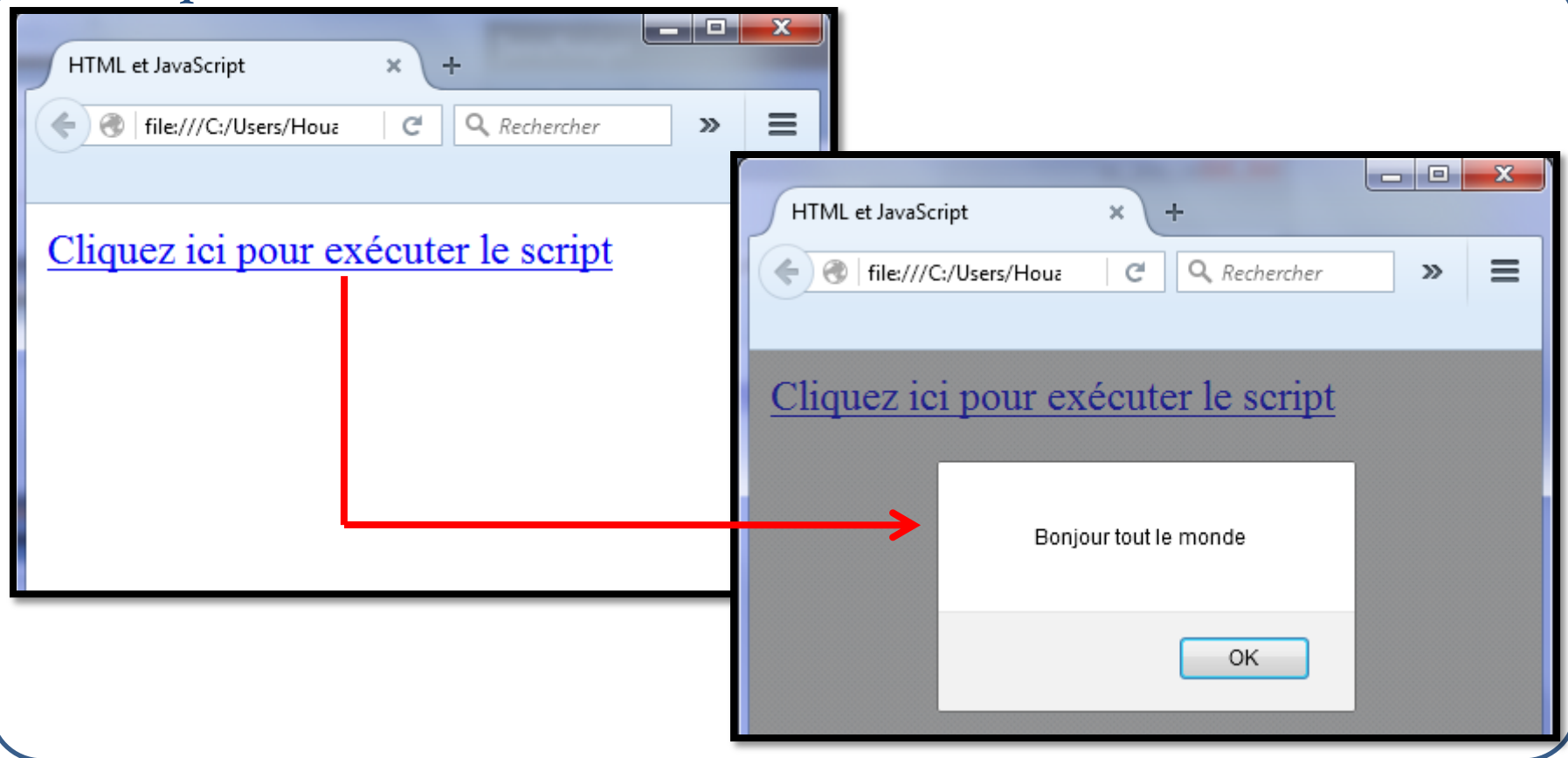
```
<html>
  <head>
    <meta charset = "UTF-8" >
    <title>HTML et JavaScript</title>
  </head>
  <body>
    <a href="javascript: (alert('Bonjour tout le monde'))">
      Cliquez ici pour exécuter le script
    </a>
  </body>
</html>
```

# Introduction

## Où définir un script JS ?

3) À travers des liens :

Exemple:



# Introduction

## Où définir un script JS ?

4) À travers des attributs d'évènement :

**Exemple:** *Script JS exécuté après le chargement du body*

```
<html>
  <head>
    <meta charset = "UTF-8" >
    <title>HTML et JavaScript</title>
  </head>
  <body onload="alert(' Page rechargée ')">
    <h3>Contenu de la page :</h3>
  </body>
</html>
```

# Introduction

## Où définir un script JS ?

4) À travers des attributs d'évènement :

Exemple: *Script JS exécuté après le chargement du body*



## Introduction

### Où définir un script JS ?

4) À travers des attributs d'évènement :

Exemple: *Scripts JS exécutés via des boutons*

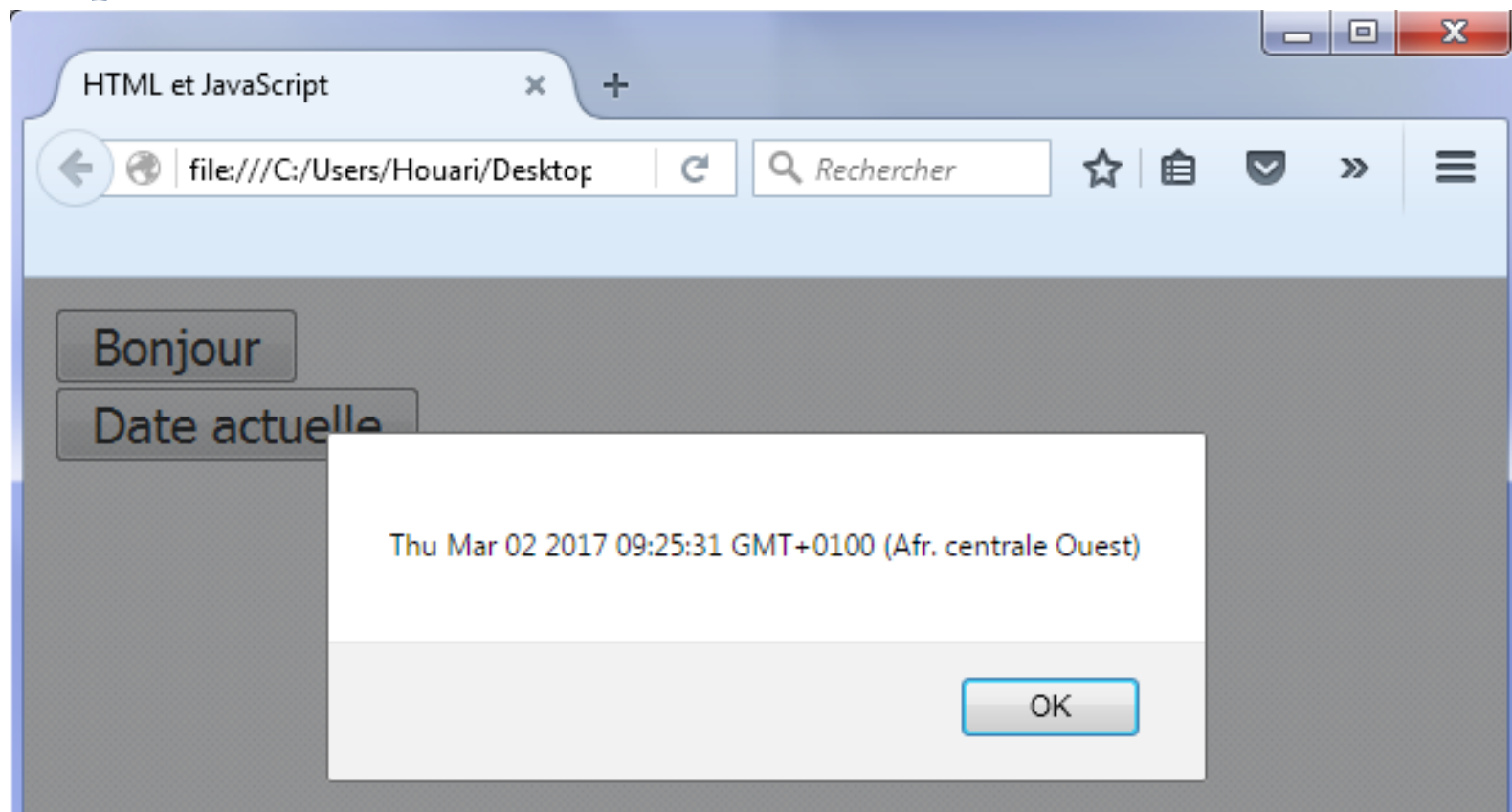
```
<html>
  <head>
    <meta charset = "UTF-8" >
    <title>HTML et JavaScript</title>
  </head>
  <body>
    <input type="button" value="Bonjour" onclick="alert('Bonjour...')" >
    <br>
    <input type="button" value="Date actuelle" onclick="alert(Date())" >
  </body>
</html>
```

# Introduction

## Où définir un script JS ?

4) À travers des attributs d'évènement :

Exemple: *Scripts JS exécutés via des boutons*



## Introduction

### Quelques attributs d'évènements:

- **onload**: Après le chargement de la page.
- **onclick**: Après avoir cliqué sur l'élément (un bouton, champ input,...).
- **ondblclick**: Après un double clic sur l'élément.
- **oninput**: Après avoir saisi une valeur dans l'élément.
- **onchange**: Après avoir changé la valeur de l'élément.
- **onselect**: Après avoir sélectionné une partie du texte de l'élément.
- **oncopy**: Après avoir copié une partie du texte de l'élément.
- **oncut**: Après avoir coupé une partie du texte de l'élément.
- **onpaste**: Après avoir collé un texte dans l'élément.
- **onreset**: Après avoir cliqué sur le bouton reset d'un formulaire.
- **onsubmit**: Après avoir cliqué sur le bouton submit d'un formulaire.
- **onkeypress**: Après avoir tapé un caractère sur le clavier.
- **onkeydown**: Après avoir cliqué sur n'importe quel bouton du clavier.
- Autres: **onbeforeprint**, **onafterprint**, **onerror**, ...



## Plan du chapitre

- 1 Introduction
- 2 Généralités
- 3 Fonctions
- 4 Tableaux
- 5 Manipulation du DOM HTML

# JavaScript



# Généralités

## Commentaires:

### Commentaire mono-ligne :

```
//Ceci est un commentaire en une seule ligne
```

### Commentaire multi-ligne :

```
/* Ceci est un commentaire  
sur plusieurs lignes */
```

## Instructions:

### Délimitées par des " ; " :

```
alert("Bonjour...") ; alert("Comment allez-vous ?") ;
```

### Délimitées par des retours à la ligne :

```
alert("Bonjour...")  
alert("Comment allez-vous ?")
```

Identificateurs:Utilisés pour :

- Nommer une fonction
- Nommer une variable

Règles de définition : Un identificateur

1. Commence pas une lettre , \_ , ou le \$.
2. Peut contenir des lettres, des chiffres ou les caractères \$ et \_.
3. Ne contient pas des caractères spéciaux.
4. Ne contient pas des espaces.
5. N'est pas un mot-clé comme **for**, **if**, **while**, **this**, **return**,...
6. Sensibles à la casse (**a** et **A** sont considérés comme deux identificateurs différents).

# Généralités

## Variables:

### Déclaration :

**var** x ;      *// Variable sans valeur*

### Affectation :

x = 12 ;      *//affectation de la valeur entière 12*

**var** x = 12 ;      *//Déclaration et affectation au même temps*

### Déclarations et affectations multiples :

**var** variable1 = "Coucou", variable2 = variable3 = 10, variable4 ;

# Généralités

Variables globales: Déclarées en dehors des blocs de fonctions, des if, des boucles.

Exemple: *Variable globale*

```
<html>
  <head>
    <meta charset = "UTF-8" >
    <title>HTML et JavaScript</title>
  </head>
  <body>
    <script>
      var A = 10 ; ← Variable globale A
    </script>
    <script>
      alert(A); ← Affiche: 10
    </script>
  </body>
</html>
```

# Généralités

Typage dynamique des variables: Le type d'une variable est déterminé automatiquement selon la valeur affectée à cette variable.

Exemple: *Typage dynamique*

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

```
var A ;
```

```
alert ( typeof (A) ); ← Affiche: undefined
```

```
A = -10.56 ;
```

```
alert ( typeof (A) ); ← Affiche: number
```

```
A = "Bonjour";
```

```
alert ( typeof (A) ); ← Affiche: string
```

```
A = false;
```

```
alert ( typeof (A) ); ← Affiche: boolean
```

```
A = [12, 34];
```

```
alert ( typeof (A) ); ← Affiche: object
```

```
</script>
```

# Généralités

## Opérateurs arithmétiques:

Opérateur	Signification	Simplification
$a = a + 12;$	<i>Addition</i>	$a += 12$
$a = a - 12;$	<i>Soustraction</i>	$a -= 12$
$a = a * 12;$	<i>Multiplication</i>	$a *= 12$
$a = a / 12;$	<i>Division réelle</i>	$a /= 12$
$a = a \% 12;$	<i>Reste de la division (modulo)</i>	$a \% = 12$
$a++;$	<i>Incrémentation</i>	
$a--;$	<i>Décrémentation</i>	

## Opérateurs booléens:

ET ( **&&** ), OU ( **||** ), NON ( **!** )

## Généralités

### Opérateurs de comparaison:

- **Égalité ( == )**: Les deux opérandes ont la même valeur.

`alert ("12" == 12.00 )`      Affiche **true**

- **Inégalité ( != )**: Les deux opérandes n'ont pas la même valeur.

`alert("12" != 12 )`      Affiche **false**

- **< , > , <= , >= .**

### Autres opérateurs:

- Le **typeof** : Retourne le type d'un opérande

`typeof 12.67`      Retourne **number**

- Test ternaire : **test ? valeur 1 : valeur 2**

```
var B = 12 ;  
var A = ( B < 0 ? (-1)*B : B ) ;
```



## Généralités

Conversion entre *Number* et *String* :

```
var A = 12 ;  
alert (typeof A );
```

*//Affiche: number*

```
A = A + "" ;  
alert (typeof A );
```

*//Affiche: string*

```
A = parseInt(A) ;  
alert (typeof A );
```

*//Affiche: number*

```
A = "1" + 2 + 3 ;  
alert (A );  
alert (typeof A );
```

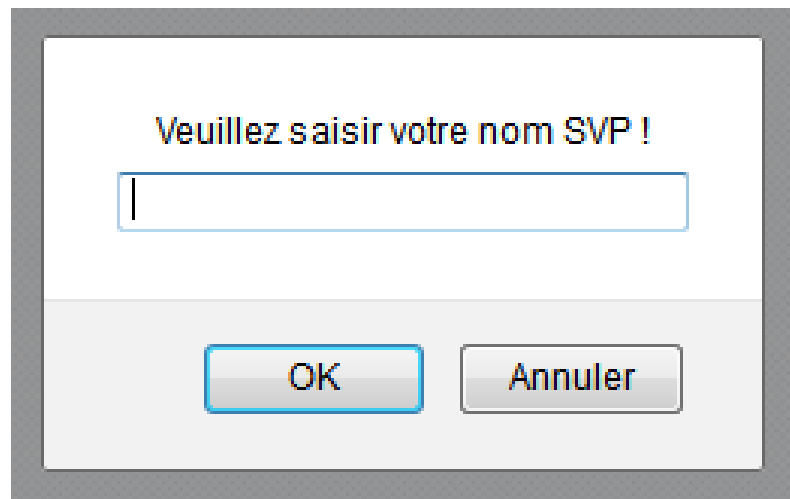
*//Affiche: 123*

*//Affiche: string*

# Généralités

Boite de saisie de valeur: La fonction **prompt()**

```
var nom = prompt(" Veuillez saisir votre nom SVP ! ");
```

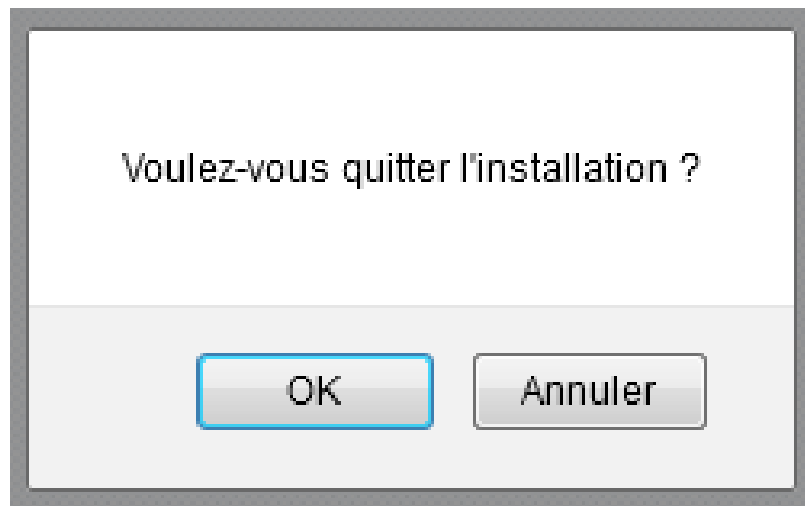


Si la saisie est annulée, alors la valeur retournée est ***null***.

## Généralités

Boite de confirmation: La fonction **confirm()**

```
var confirmation = confirm(" Voulez-vous quitter l'installation ? ");
```



Valeur retournée : *true* ou *false*.

# Généralités

## Branchement conditionnel:

### Le " if " :

```
if (exp_logique)  
    alert ("condition vraie");
```

*//Exécution d'une seule ligne  
d'instructions*

```
if (exp_logique) {  
    alert ("condition vraie");  
    alert ("...je le confirme");  
}
```

*//Exécution d'un bloc  
d'instructions*

Branchement conditionnel:Le " if else " :

```
if (exp_logique)  
    alert ("condition vraie");  
else  
    alert ("condition fausse");
```

*//Exécution d'une seule ligne  
d'instructions*

```
if (exp_logique) {  
    alert ("condition vraie");  
    alert ("...je le confirme");  
} else {  
    alert ("condition fausse");  
    alert ("...je le confirme");  
}
```

*//Exécution d'un bloc  
d'instructions*

Branchement conditionnel:Le " switch " :

```
switch (variable){  
    case 0: alert("valeur nulle");  
           break;  
    case 1:  
    case 3: alert("valeur impaire");  
           break;  
    case 2:  
    case 4: alert("valeur paire");  
           break;  
    default: alert("variable > 4 , ou < 0");  
}
```

Boucles:Le " for " :

```
for (var i = -4 ; i <= 4 ; i+=2) {  
    alert(i);  
}
```

*//Affiche: -4, -2, 0, 2, 4.*

```
for (var i = -2 ; i <= 2 ; i++) {  
    if (i==0) break;  
    alert(i);  
}
```

*//Affiche: -2, -1.*

```
for (var i = -2 ; i <= 2 ; i++) {  
    if (i==0) continue;  
    alert(i);  
}
```

*//Affiche: -2, -1, 1, 2 mais pas le zéro.*

# Généralités

## Boucles:

### Le " while " :

```
while (condition) {  
    actions;  
}
```

### Le " do while " :

```
do {  
    actions;  
} while (condition);
```



## Généralités

### Modifier les paramètres d'un élément:

- Méthode `document.getElementById("id_element").propriété=valeur`

## Modifier les paramètres d'un élément:

- Méthode **document.getElementById("id\_element").propriété=valeur**
- Quelques propriétés :
  - **value** : La valeur de l'attribut **value** des éléments **input**, contenu du **textarea**
  - **textContent** : La valeur d'un **td**, **h1**, **label**,...
  - **style.color** : La couleur du texte.
  - **style.width** : La largeur de l'élément.
  - **style.height** : La hauteur de l'élément.
  - **style.borderColor** : La couleur de la bordure.
  - **style.borderStyle** : Le style de la bordure.
  - **style.backgroundColor** : La couleur du fond.
  - **style.fontFamily** : La police du texte.
  - **style.textAlign** : L'alignement du texte.

## Généralités

Modifier les paramètres d'un élément:

Voir *Exemple 1.html*

## Plan du chapitre

- 1 Introduction
- 2 Généralités
- 3 Fonctions
- 4 Tableaux
- 5 Manipulation du DOM HTML

# JavaScript

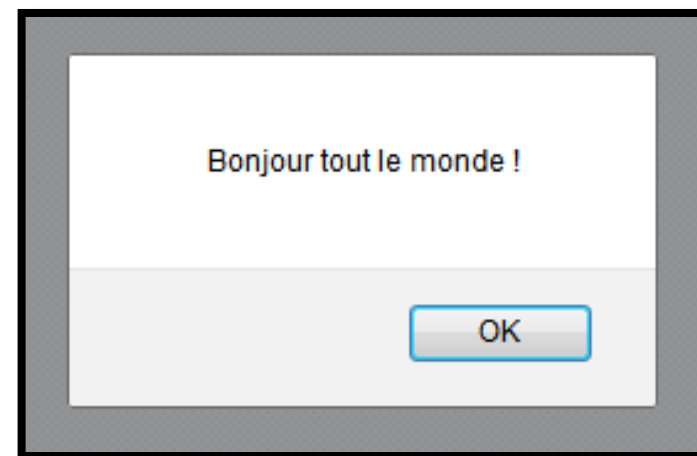


# Fonctions

## Fonctions sans paramètres:

### Exemple:

```
<html>
  <head>
    <title>Déclaration des fonctions</title>
    <script>
      function salutation(){
        alert("Bonjour le monde !");
      }
      salutation();
    </script>
  </head>
  <body>
  </body>
</html>
```

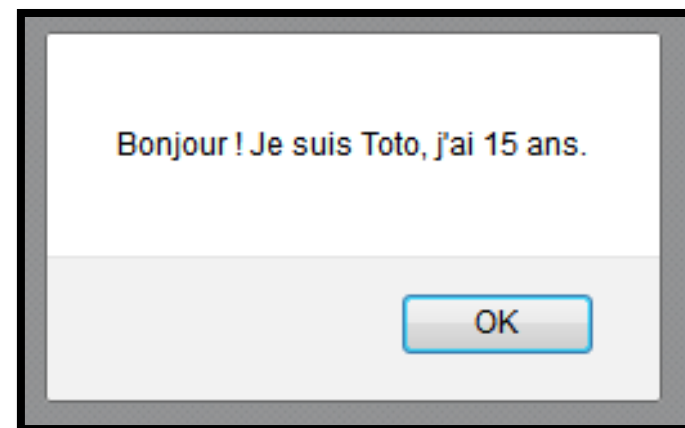


# Fonctions

## Fonctions avec paramètres:

### Exemple:

```
<html>
  <head>
    <title>Déclaration de fonctions</title>
    <script>
      function salutation(nom, age){
        alert("Bonjour ! Je suis "+nom+", j'ai "+age+" ans.");
      }
      salutation("Toto", 15);
    </script>
  </head>
  <body>
  </body>
</html>
```

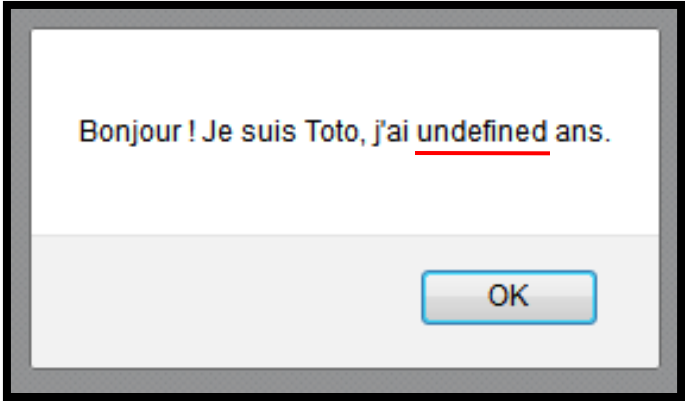


# Fonctions

## Fonctions avec paramètres facultatifs:

Exemple:

```
<html>
  <head>
    <title>Déclaration de fonctions</title>
    <script>
      function salutation(nom, age){
        alert("Bonjour ! Je suis "+nom+", j'ai "+age+" ans.");
      }
      salutation("Toto");
    </script>
  </head>
  <body>
  </body>
</html>
```



Bonjour ! Je suis Toto, j'ai undefined ans.

OK

# Fonctions

## Fonctions avec paramètres facultatifs:

Exemple:

```
<script>
function salutation(nom, age){
    if(typeof(age) == "undefined"){
        alert("Bonjour ! Je suis "+nom);
    } else {
        alert("Bonjour ! Je suis "+nom+", j'ai "+age+" ans.");
    }
}
salutation("Toto" );
salutation("Toto" , 15 );
</script>
```

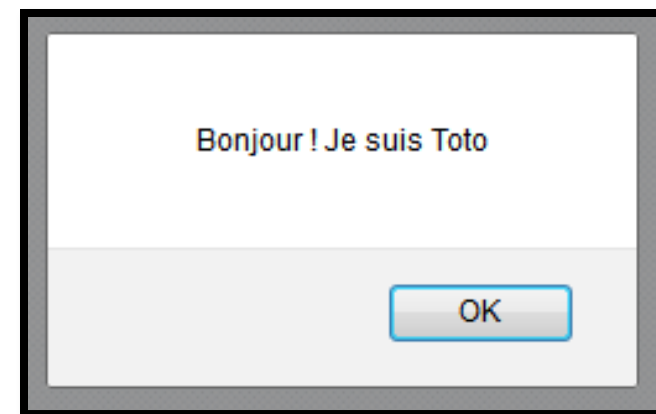


# Fonctions

## Fonctions avec paramètres facultatifs:

Exemple:

```
<script>
function salutation(nom, age){
    if(typeof(age) == "undefined"){
        alert("Bonjour ! Je suis "+nom);
    } else {
        alert("Bonjour ! Je suis "+nom+", j'ai "+age+" ans.");
    }
}
salutation("Toto");
salutation("Toto", 15);
</script>
```

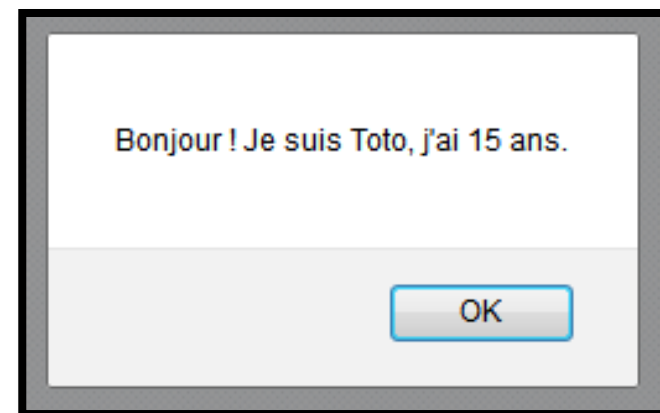


# Fonctions

## Fonctions avec paramètres facultatifs:

Exemple:

```
<script>  
function salutation(nom, age){  
    if(typeof(age) == "undefined"){  
        alert("Bonjour ! Je suis "+nom);  
    } else {  
        alert("Bonjour ! Je suis "+nom+", j'ai "+age+" ans.");  
    }  
}  
salutation("Toto");  
salutation("Toto", 15);  
</script>
```



# Fonctions

## Fonctions avec valeur de retour:

Exemple:

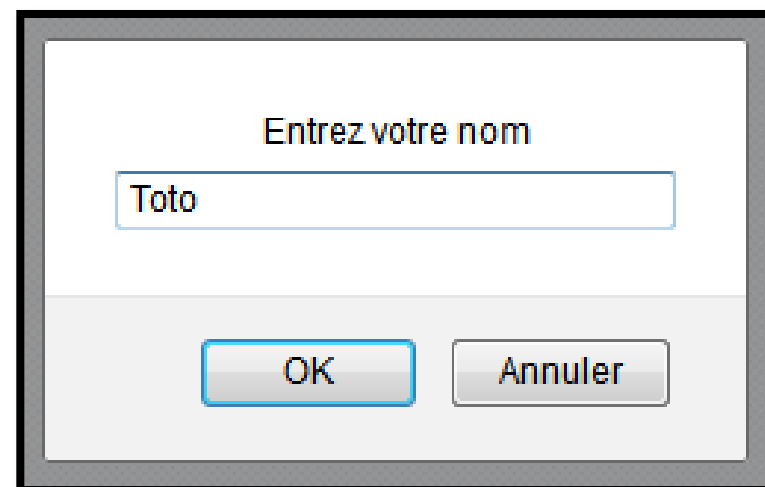
```
<script>  
  function nom() {  
    return prompt("Entrez votre nom");  
  }  
  function age() {  
    return prompt("Entrez votre age");  
  }  
  alert("Je suis "+nom()+" , j'ai "+age());  
</script>
```

# Fonctions

## Fonctions avec valeur de retour:

Exemple:

```
<script>  
  function nom() {  
    return prompt("Entrez votre nom");  
  }  
  function age() {  
    return prompt("Entrez votre age");  
  }  
  alert("Je suis "+nom()+"", j'ai "+age());  
</script>
```

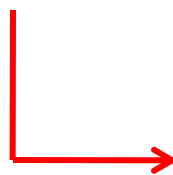


# Fonctions

## Fonctions avec valeur de retour:

Exemple:

```
<script>  
  function nom() {  
    return prompt("Entrez votre nom");  
  }  
  function age() {  
    return prompt("Entrez votre age");  
  }  
  alert("Je suis "+nom()+" , j'ai "+age());  
</script>
```

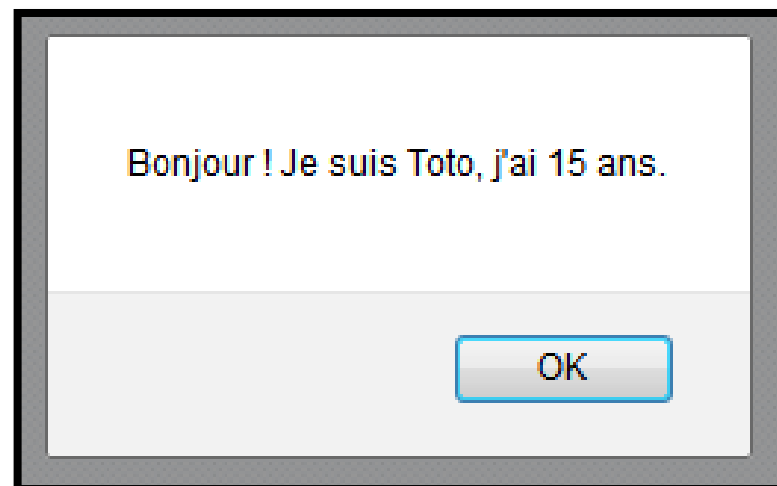
A browser dialog box titled "Entrez votre age". It features a text input field containing the number "15". Below the input field is a checkbox labeled "Empêcher cette page d'ouvrir des dialogues supplémentaires". At the bottom of the dialog are two buttons: "OK" and "Annuler".

# Fonctions

## Fonctions avec valeur de retour:

Exemple:

```
<script>  
  function nom() {  
    return prompt("Entrez votre nom");  
  }  
  function age() {  
    return prompt("Entrez votre age");  
  }  
  alert("Je suis "+nom()+" , j'ai "+age());  
</script>
```



Gestion des évènements liés au clavier:

Voir *Exemple 2.html*

Autres:

```
window.open("www.univ-tlemcen.dz");  
window.print();
```