

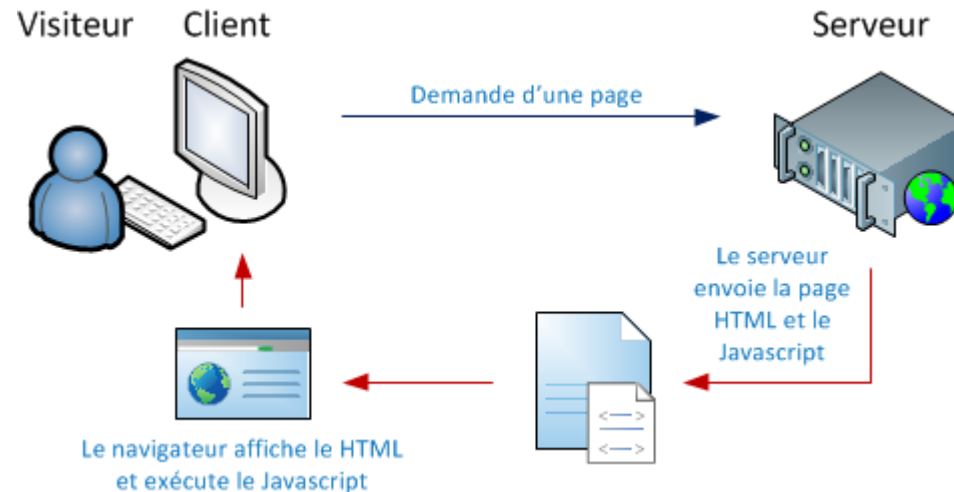
# Le langage Javascript pour documents HTML

## JavaScript : Principe

- Le JavaScript est un langage de programmation de scripts orienté objet.
- Le JavaScript est utilisé sur Internet, conjointement avec les pages Web HTML.
- Le JavaScript s'inclut directement dans la page Web (ou dans un fichier externe) et permet de *dynamiser* une page HTML :
  - Afficher/masquer du texte ;
  - Faire défiler des images ;
  - Créer un diaporama avec un aperçu « en grand » des images ;
  - Etc.

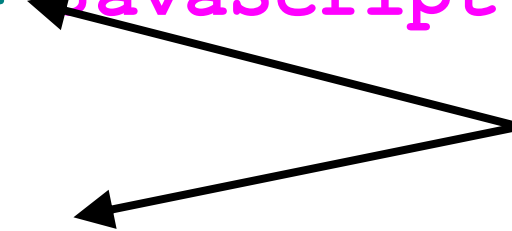
## JavaScript : Principe

- Le JavaScript est un langage dit **client-side**, c'est-à-dire que les scripts sont exécutés par le navigateur chez l'internaute (le **client**). Cela diffère des langages de scripts dits **server-side** qui sont exécutés par le serveur Web. C'est le cas des langages comme le [PHP](#).



## JavaScript : Balise script

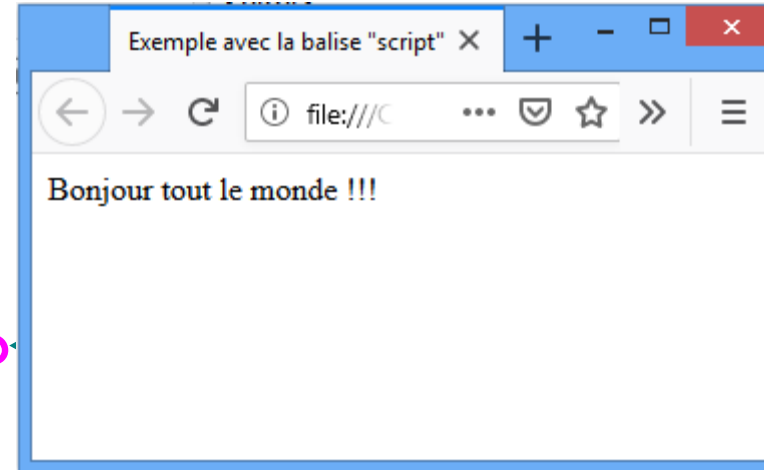
```
<script type="text/javascript"  
  language="JavaScript">  
<!--  
  script  
  // -->  
</script>
```



Masquer le script aux navigateurs non compatibles avec JavaScript

# JavaScript : Exemple

```
<html>
  <head>
    <title>Ma première page Web
  </head>
  <body>
    <script type="text/javascript" language="JavaScript">
      <!--
        document.writeln("Bonjour tout le monde !") ;
      // -->
    </script>
  </body>
</html>
```



## JavaScript : un fichier externe (.js)

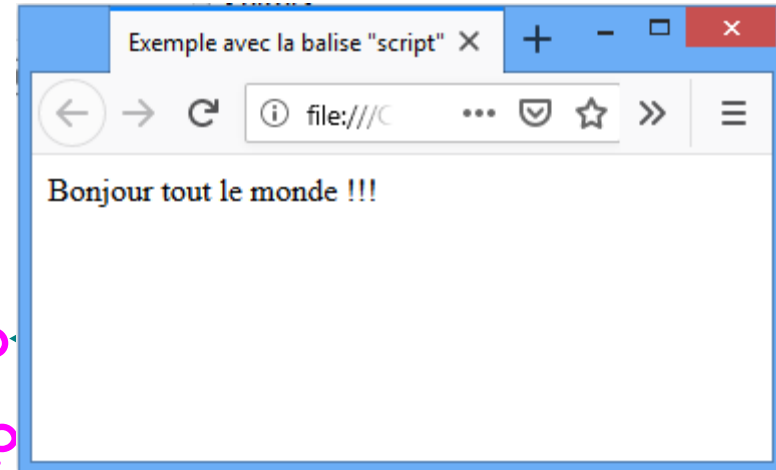
```
<script type="text/javascript"  
  language="JavaScript" src="URI">  
</script>
```



Le chemin à emprunter pour accéder  
au fichier

# JavaScript : Exemple

```
<html>
  <head>
    <title>Ma première page Web
  <script type="text/javascript"
    src="monscript.js">
  </script>
</head>
<body>
</body>
</html>
```



```
document.writeln("Salut!") ;
```

# Variables

- Déclaration de variables facultative
- Variables non typées à la déclaration

`var nom_variable ;`

- Typage dynamique à l'affectation
- Types gérés:
  - Nombres (`10`, `3.14`)
  - Booléens (`true`, `false`)
  - Chaînes (`"Salut !"`, `'Salut !'`)
  - `null`
  - `undefined`



# Les opérateurs arithmétiques

- Les opérateurs de base sont

Opérateur	Signe
addition	+
soustraction	-
multiplication	*
division	/
modulo	%

- Concernant le dernier opérateur, le modulo est tout simplement le reste d'une division.
- Exemple : 5 modulo 2 vaut 1.

# La concaténation

- Une concaténation consiste à ajouter une chaîne de caractères à la fin d'une autre, comme dans cet exemple :

```
<html>
  <head>
    <title>Ma première page Web</title>
  </head>
  <body>
    <script type="text/javascript" language="JavaScript">
      var ch1="bonjour",ch2="Ali";
      message=ch1+ch2;
      Alert(message);
    </script>
  </body>
</html>
```

# Interagir avec l'utilisateur

- La fonction `prompt()` permet l'interaction avec l'utilisateur :
- Exemple :

```
<html>
  <head>
    <title>Ma première page Web</title>
  </head>
  <body>
    <script type="text/javascript" language="JavaScript">
      var Nom = prompt('Entrez votre prénom :');
      alert(Nom); // Affiche le prénom entré par
                  l'utilisateur
    </script>
  </body>
</html>
```

## Exercice1

- Ecrire un code qui calcule la somme de deux entiers entrés par l'utilisateur.

```
<html>
  <head>
    <title>Ma première page Web</title>
  </head>
  <body>
    <script type="text/javascript" language="JavaScript">
      var n1 = parseInt(prompt('Entrez un entier:'));
      var n2 = parseInt(prompt('Entrez un entier:'));
      var s=n1+n2;
      alert("s="+s); // Affiche le prénom entré par
      l'utilisateur
    </script>
  </body>
</html>
```

# Instructions conditionnelles

- Il s'agit des instructions à utiliser pour tester les conditions.
- Les opérateurs de comparaison :

Opérateur	Signification
==	égal à
!=	différent de
===	contenu et type égal à
!==	contenu ou type différent de
>	supérieur à
>=	supérieur ou égal à
<	inférieur à
<=	inférieur ou égal à

# Instructions conditionnelles

- Les opérateurs logiques :

Opérateur	Type de logique	Utilisation
&&	ET	valeur1 && valeur2
	OU	valeur1    valeur2
!	NON	!valeur

# Instructions conditionnelles

- L'instruction if :

```
if (condition)
```

```
{
```

```
    instructions ;
```

```
}
```

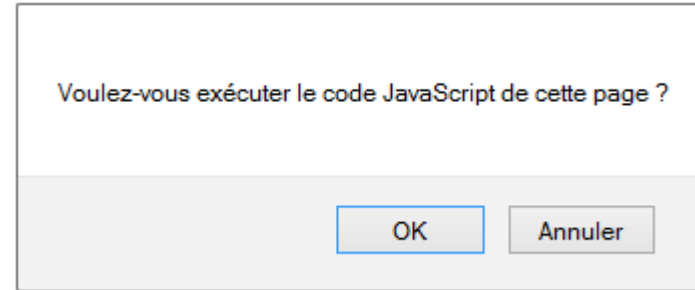
- Exemple :

```
    if confirm("voulez-vous exécuter le code  
                JavaScript de cette page?")
```

```
    {
```

```
        Alert("Le code a bien été exécuté");
```

```
    }
```



# Instructions conditionnelles

- Si on souhaite exécuter un code si la condition est fausse :

```
if (condition)
{
    instructions 1 ;
}
[ else
{
    instructions 2;
} ]
```



# Instructions conditionnelles

- Exemple :

```
if confirm("pour accéder à ce site vous  
devez avoir 18 ans ou plus cliquez sur ok  
si c'est le cas'.")){  
    Alert("Vous allez être redirigé vers le  
site.");  
}  
else{  
    Alert("Désolé, vous n'avez pas accès à ce  
site.« ) ;  
}
```

# Instructions conditionnelles

- Si vous avez plusieurs cas à traiter, on peut utiliser l'instruction `else if` pour traiter à chaque fois le sinon d'une condition :

```
if (condition 1){  
    instructions 1 ;  
}  
  
else if (condition 2) {  
    instructions 2;  
}  
  
else {  
    instructions 3;  
}
```

## Instructions conditionnelles

- C'est une instruction qui permet de traiter plusieurs cas (plus pratique que else si) :

```
switch (expression) {  
    case étiquette :  
        instructions ;  
        break ;  
    case étiquette :  
        instructions ;  
        break ;  
    default :  
        instructions ;  
}
```

## Exercice d'application

- Écrire un code qui permet de fournir un commentaire sur quatre tranches d'âge selon le tableau ci-dessous :

Tranche d'âge	Exemple de commentaire
1 à 17 ans	« Vous n'êtes pas encore majeur. »
18 à 49 ans	« Vous êtes majeur mais pas encore senior. »
50 à 59 ans	« Vous êtes senior mais pas encore retraité. »
60 à 120 ans	« Vous êtes retraité, profitez de votre temps libre ! »