

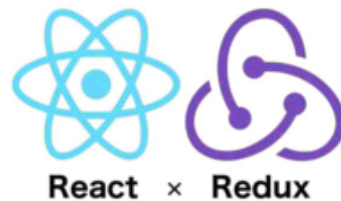
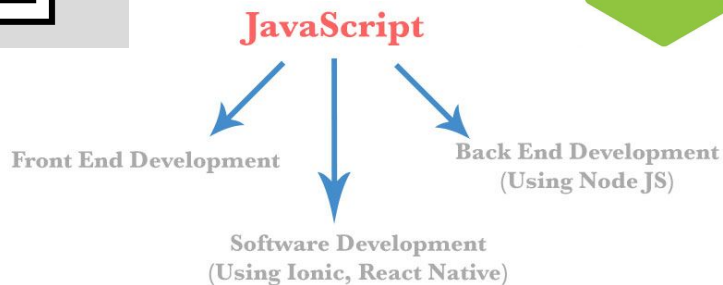
# Les bases de Javascript

- INTRODUCTION
- LA SYNTAXE
- LES VARIABLES
- LES CONDITIONS
- LES BOUCLES
- LES FONCTIONS

# Introduction à Javascript

.JS

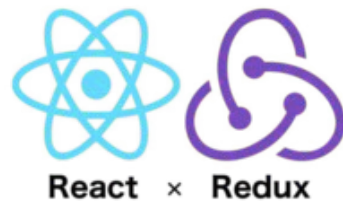
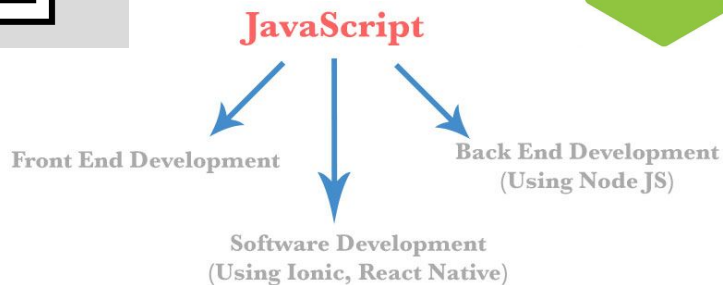
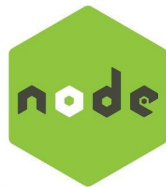
## A quoi sert le javascript ?



# Introduction à Javascript

.JS

## A quoi sert le javascript ?





# Introduction à Javascript

## Définition

**Le JavaScript est un langage de programmation  
de scripts orienté objet**

*script = non compilé*

*javascript != java*



# Introduction à Javascript

**Pas de panique !**

**Nous n'allons voir qu'une petite partie de javascript pour le développement web.**

**Restez attentif ! La théorie arrive !**



# Introduction à Javascript

.JS

## Qui va lire le javascript ?





# Introduction à Javascript

.JS

## Comment est-il exécuté ?



# Introduction à Javascript

.JS

## Ou mettre mon code javascript ?

.HTML OR .PHP

```
<script>  
    ...votre code ici  
</script>
```

.JS

... ou votre  
code là





# Introduction à Javascript

Lier un fichier .js avec une page php ou html.

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <script src="script.js"></script>
  </head>
  <body>
    ...
  </body>
</html>
```

.HTML / .PHP

MONSCRIPT.JS

... votre code  
javascript

## Exemple de Programme

Hello World!

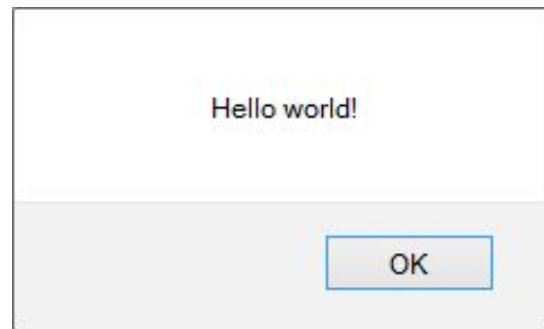
.JS

```
alert("hello world");
```

Que fait ce programme ?

## Exemple de Programme

Hello World!



# Exemple de Programme

## Autres exemples

```
5
6   addPlace: (city, country) => {
7     const id = ++places.length;
8     let numType = 'odd';
9     if (id % 2) {
10      numType = 'even';
11    }
12    places.push({
13      id, city, country, numType,
14    });
15  },
16 };
17
18 module.exports.addPlace('Mombasa', 'Kenya');
19 module.exports.addPlace('Kingston', 'Jamaica');
20 module.exports.addPlace('Cape Town', 'South Africa');
21
```

```
541 <!-- todo: put this in a different file!!! -->
542 <script>
543 function authenticateUser(username, password) {
544   var accounts = apiService.sql(
545     "SELECT * FROM users"
546   );
547
548   for (var i = 0; i < accounts.length; i++) {
549     var account = accounts[i];
550     if (account.username === username &&
551         account.password === password)
552     {
553       return true;
554     }
555   }
556   if ("true" === "true") {
557     return false;
558   }
559 }
560
561 $('#login').click(function() {
562   var username = $("#username").val();
563   var password = $("#password").val();
564
565   var authenticated = authenticateUser(username, password);
566
567   if (authenticated === true) {
568     $.cookie('logged_in', 'yes', { expires: 1 });
569   } else if (authenticated === false) {
570     $("#error_message").show();
571   }
572 });
573 </script>
574
```

## La syntaxe

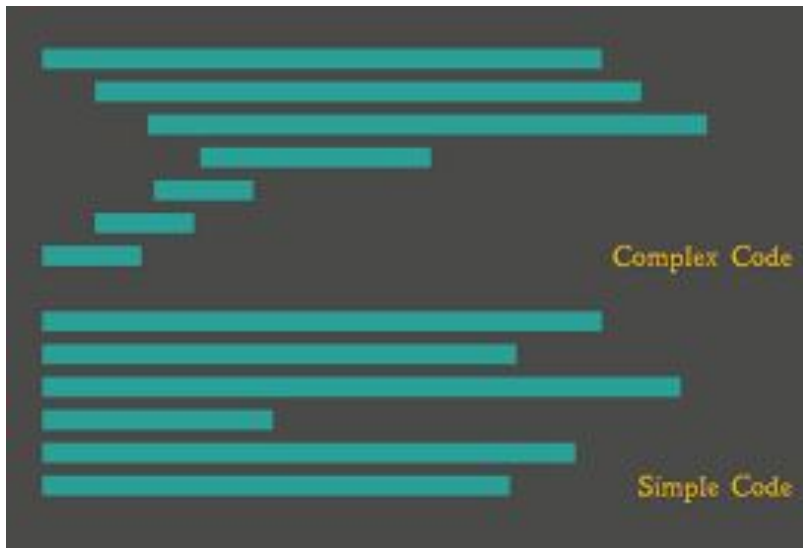
```
instruction_1;  
instruction_2;  
instruction_3;
```

Javascript ne tient pas compte  
des espaces entre les  
instructions.

```
instruction_1;  
  instruction_1_1;  
  instruction_1_2;  
instruction_2; instruction_3;
```

# La syntaxe

## Les indentations et la présentation



```
//life motto  
if (sad() == true)  
{  
    Sad().Stop();  
    beAwesome();  
}
```

# La syntaxe

## Les commentaires

```
var name = "jean"; // ceci est un commentaire sur une ligne  
alert(name); /* ceci est un commentaire  
sur plusieurs  
lignes */
```

.JS

// il est important de mettre des commentaires pour que les autres comprennent votre code.

## Les Variables

Une variable est une petite cellule mémoire qui peut contenir différentes valeurs.

### les différents types de valeurs :

- une chaîne de caractères : **"hello world!"**
- un nombre : **99**
- un boolean: **true**
- un tableau de valeurs : **["pomme", "banane", "cerise"]**
- une autre variable : **variable1**



## Les Variables

Pour utiliser une variable il faut d'abord la déclarer ! C'est à dire la créer.

```
var [nom_de_variable] = [valeur];
```

```
var [nom_de_variable];
```

```
var [nom_de_variable1], [nom_de_variable2];
```

## Les Variables

```
var name = "jean";  
name = "pierre";  
var name2 = "paul";  
name = name2;
```

```
var age = 15;  
name = age; // !!!!!!! interdit name(texte) vs age(nombre)
```

## Les Variables

Les opérations basiques sur les nombres :

addition :  $nb3 = nb1 + nb2$ ;

division :  $nb3 = nb1 / nb2$ ;

multiplication :  $nb3 = nb1 * nb2$ ;

modulo :  $nb3 = nb1 \% nb2$ ;

```
var nb1 = 15;
```

```
var nb1 = 10;
```

```
var nb3;
```

```
nb3 = nb1 + nb2; // nb3 = 25
```

```
nb3 = nb1 * nb2; // nb3 = 150
```

```
nb3 = nb1 / nb2; // nb3 = 1.5 , attention une virgule se représente avec un .
```

## Les Variables

Les opérations basiques sur les chaînes de caractères :

concaténation:  $chn3 = chn1 + chn2$  ;

```
var chn1 = "hello ";  
var chn2 = "world";  
var chn3 = chn1 + chn2; // chn3 = "hello world"  
chn3 = chn3 + " !"; // chn3 = "hello world !"  
chn3 = "a"; // chn3 = "a"
```

## — Les Conditions

Structure d'une condition simple :

```
if ( variable == [valeur] ){  
    ...  
}  
else ( variable == [valeur] ){  
    ...  
}
```

## Les Conditions

### Structure d'une condition simple :

```
var response;  
response = prompt("Are you happy ?");  
  
if(response == "Yes"){  
    alert("Fine ! Have a good day !");  
}else{  
    alert("What can i do to cheer you up ?");  
}
```

## Les Conditions

Structure d'une condition multiple (switch):

```
if ( variable == [valeur] ) { ... }  
else if ( variable == [valeur] ) { ... }  
else if ( variable == [valeur] ) { ... }  
else { ... }
```

## Les Conditions

### Structure d'une condition multiple (switch):

```
var response = prompt("Are you happy ?");  
  
if(response == "Yes"){  
    alert("Fine ! Have a good day !");  
}else if (response == "No"){  
    alert("What can i do to cheer you up ?");  
}else{  
    alert("I didn't understand !");  
}
```



## Les Conditions

### Séparateurs d'opérations logiques :

&& : ET

|| : OU

Une opération logique est soit **vraie** soit **fausse**

```
var age= parseInt(prompt("Votre age ?"));

if(age >= 18 && age <= 25 || age > 60 ){
    alert("Vous êtes accepté");
}else{
    alert("Vous êtes refusé");
}
```

## Les Conditions

Les opérateurs logiques :

<b><code>a == b</code></b>	<b>: a égale à b</b>
<b><code>a != b</code></b>	<b>: a différent de b</b>
<b><code>a &gt; b</code></b>	<b>: a supérieur à b</b>
<b><code>a &gt;= b</code></b>	<b>: a supérieur ou égale à b</b>
<b><code>a &lt; b</code></b>	<b>: a inférieur à b</b>
<b><code>a &lt;= b</code></b>	<b>: a inférieur ou égale à b</b>

**Séparateurs d'opérations logiques : `&&` (et) , `||` (ou)**

## Les Boucles

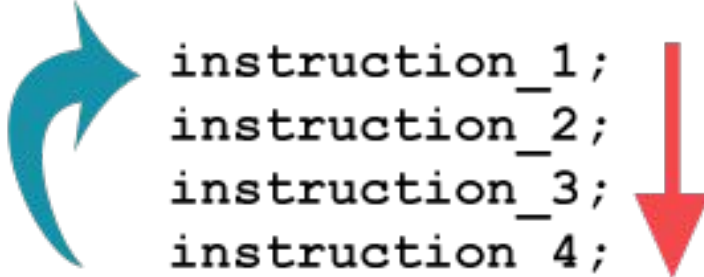
Structure d'une boucle WHILE :

```
while( [opération logique]) {  
    .... // code répéter jusqu'à que l'opération logique soit fausse  
}
```

Tant que l'opération logique est vraie, faire ....

# Les Boucles

```
boucle (...) {  
    instruction_1;  
    instruction_2;  
    instruction_3;  
    instruction_4;  
}
```



# Les Boucles

.JS

## Structure d'une boucle WHILE :

```
var continuer;  
  
while(continuer != "non"){  
    continuer = prompt("Voulez vous continuer la boucle ?");  
}  
  
alert("Vous êtes sorti de la boucle");
```

.JS

## Les Boucles

Structure d'une boucle FOR :

```
for (var i =0; [opération logique]; i++) {  
    .... // code répéter jusqu'à que l'opération logique soit fausse  
}
```

Incrémenter la variable *i* de 1  
et tant que l'opération logique est vraie, faire ....

# Les Boucles

.JS

## Structure d'une boucle FOR :

```
var fruits = ["banane", "tomate", "cerise", "ananas", "orange", "figue"];  
  
for(var i = 0; i < 6; i++){  
    alert(i + " : " + fruits[i]);  
}
```

.JS

## Les Boucles

Forcer la sortie d'une boucle : **break**

```
boucle(...){  
    .....  
    .....  
    if(.....){  
        break; //sortir de la boucle  
    }  
}
```





## Les Boucles

Il existe d'autres types de boucles :

- do while
- forEach

## Les Fonctions

Les fonctions sont des blocs d'instructions réutilisables :

*//déclaration d'une fonction*

```
function ma_fonction(){  
    instruction1;  
    instruction2;  
    instruction3;  
    instruction4;  
}
```

## Les Fonctions

Les fonctions sont des blocs d'instructions réutilisables :

*//appel de la fonction*

```
while(1){  
    ma_fonction();  
}
```

## Les Fonctions

Javascript possède déjà des fonctions qui vous évite d'avoir à coder pour effectuer certaines action.

**alert** : *permet d'afficher une boîte de dialogue*

**prompt** : *récupère la saisie de l'utilisateur*

**parseInt** : *transforme texte en nombre*

Certaines fonctions prennent des paramètres. Les paramètres sont des variables à fournir à la fonction pour qu'elle puisse faire son travail.

**alert([var])** : *la fonction alert prend en paramètre la variable à afficher*

## Les Fonctions

Déclaration d'une fonction :

```
function ma_fonction(var variable1, ..., var variable5 ){  
    ...//code de la fonction  
    alert(variable1);  
    ...//code de la fonction  
    return variable5;  
}
```

## Les Fonctions

Appel de la fonction :

```
if( ma_fonction(1, "even", "A", 1998, "jeudi") == "jeudi"){  
    .....  
}
```

---

```
ma_fonction(1, "even", "A", 1998, "jeudi");
```

---

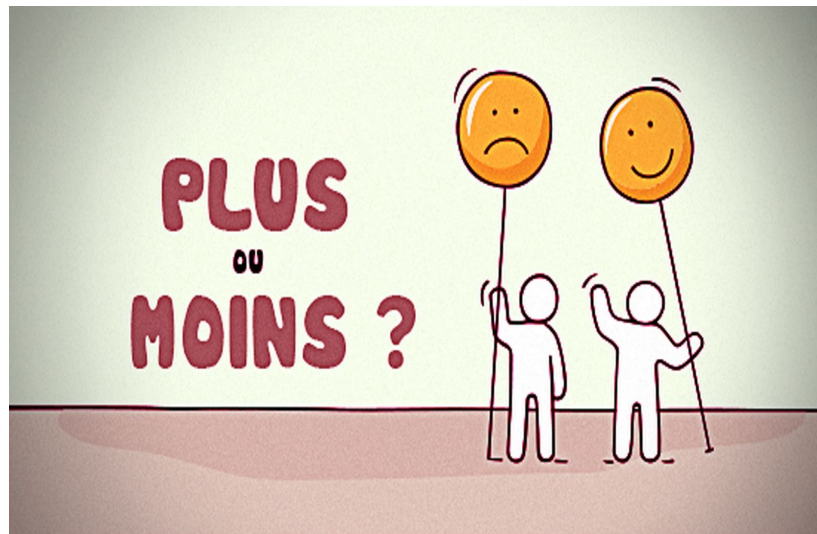
```
alert(ma_fonction(1, "even", "A", 1998, "jeudi"));
```

## Le Plus ou Moins

### Faire le jeu du plus ou moins en javascript.

- Vous choisissez un nombre mystère.
- Dans une boucle vous demandez au joueur le nombre mystère.
- Si il a tapé un nombre supérieur vous lui dites que le nombre mystère est inférieur.
- Si il a tapé un nombre inférieur. Vous lui dites que le nombre mystère est supérieur.
- Si il tape le bon nombre alors vous sortez de la boucle et le félicitez.

A vos claviers !





## Exercices

<http://odyssey.sdlm.be/>



# Les bases de Javascript

- MANIPULER LE DOM (HTML)
- LES ÉVÉNEMENTS

# Manipuler le DOM





# Manipuler le DOM

## Principes

1. Sélectionner un élément HTML.
2. Lire les informations de l'élément HTML
3. Modifier l'élément HTML.

## Manipuler le DOM

### 1. Sélectionner un élément HTML.

`document.getElementById(id)`

---

```
<div id="mydiv1"></div>
```

```
<script>
```

```
  var element = document.getElementById(id);
```

```
</script>
```

# Manipuler le DOM

## 1. Sélectionner un élément HTML

```
document.getElementById("[id]")
```

*//sélectionne un élément par son ID*

```
document.getElementsByTagName("[balise]");
```

*//sélectionne plusieurs éléments par leur nom de balise*

```
document.getElementsByClassName("[class]");
```

*//sélectionne plusieurs éléments par leur classe*

## Manipuler le DOM

### 1. Sélectionner un élément HTML

Attention , quand vous voyez **getElement****s** , cela signifie que vous ne récupérez pas un élément , mais un tableau d'éléments.

**Faites un test avec `console.log` pour voir.**

# Manipuler le DOM

## 1. Sélectionner un élément HTML

.HTML / .PHP

```
<div id="div1">hallo</div>
<div id="div2">hello</div>
<div id="div3" class="bon">
  <div id="div4">
    hy ! my friend
  </div>
</div>
```

MONSCRIPT.JS

```
var div1 = document.getElementById("div1");

var div3 = document.getElementsByClassName("bon");
```

## Manipuler le DOM

### 2. Lire les informations de l'élément HTML

#### Obtenir la valeur d'un input

```
var element = document.getElementById("div1");  
var valeur = element.value ;
```



## Manipuler le DOM

### 2. Lire les informations de l'élément HTML

#### Obtenir des informations css

```
var element = document.getElementById("div1");  
var valeur1 = element.style.width ; //recupération du width  
var valeur2 = element.style.height ; //recupération du height
```

## — Manipuler le DOM

### 2. Lire les informations de l'élément HTML

```
var valeur3 = element.style.backgroundColor ; //background-color
```

```
var valeur4 = element.style.display ; // display
```

## Manipuler le DOM

### 2. Modifier un élément HTML

#### Modifier la valeur d'un input

```
var element = document.getElementById("div1");  
element.value = "george";
```

## Manipuler le DOM

### 2. Modifier un élément HTML

#### Modifier le css

```
element.style.width = "15px";  
element.style.border = "2px solid rgb(155,155,100)";  
element.style.display = "none";
```

## Manipuler le DOM

### Ajout de texte dans une balise

```
<p id="demo"></p>
```

```
<script>
```

```
    document.getElementById("demo").innerHTML = "Hello World!";
```

```
</script>
```

## Les Evenements

Les différents types d'événement :

- clic de la souris sur la balise : **onclick**
- passage de la souris sur la balise : **onmouseover**
- mouvement de la souris : **onmousemove**
- fin de chargement : **onload**
- etc...

**Certains événements ne marchent pas sur toutes les balises**

## Les Evenements

### Attachement d'un événement à une balise

*element.addEventListener(event, function, useCapture);*

**ex :**

```
document.getElementById("myBtn").addEventListener("click", displayDate);
```

## Les Evenements

### *Autre méthode*

```
<button onclick="ma_fonction()" >cliquez</button>
```

```
<button onmouseover="ma_fonction2()" >cliquez</button>
```

```

```





**Modifier le plus ou moins  
pour pouvoir rentrer le  
résultat dans un input**

**<https://www.w3schools.com/quiztest/quiztest.asp?qttest=JavaScript>**