

# MÉMO2 - DÉVELOPPEMENT JS Jour2

*Cours Javascript*  
*Frédéric LOSSIGNOL*

## Rappel

Nous avons vu en JOUR 1 :

- les variables et les constantes
- les tableaux
- la concaténation
- Les types de variable (string / number) et les conversions (parseInt, parseFloat)
- la class Date()

**Voici maintenant 1 notion fondamentale que nous retrouverons dans tous les langages de programmation :**

- Les **Conditions**

## A retenir :

**Les conditions** vous permettent d'exécuter une ou plusieurs instruction(s), **SI** une condition que vous avez déterminée est remplie.

Nous verrons ci-dessous la syntaxe d'une condition.

*Note : Tous les bouts de code que vous trouverez dans ce document sont directement exécutables et testables depuis la console de votre navigateur (Firebug ou la console sur Chrome), ou même sur <https://jsfiddle.net>*

# SOMMAIRE

## 1. Les conditions

### 1.1. Pourquoi utiliser des conditions

### 1.2. La syntaxe d'une condition

### 1.3. La syntaxe du switch (remplace les else if)

**Avant d'aborder les conditions**, on a besoin de connaître les opérateurs de comparaison, qui permettent de comparer 2 valeurs, ou 2 expressions. Un opérateur de comparaison renvoie un booléen (**TRUE** ou **FALSE**), c'est son seul rôle.

#### *Les opérateurs de comparaison :*

<b>a == b</b>	Vérifie l'égalité de valeur
<b>a === b</b>	Vérifie la stricte égalité (valeur et type)
<b>a &lt; b</b>	Vérifie que a est inférieur à b
<b>a &lt;= b</b>	Vérifie que a est inférieur ou égal à b
<b>a &gt; b</b>	Vérifie que a est supérieur à b
<b>a &gt;= b</b>	Vérifie que a est supérieur ou égal à b
<b>a != b</b>	Vérifie que a est différent de b
<b>a !== b</b>	Vérifie que a est différent de b en valeur et en type

**Si besoin, les opérateurs logique ET et OU** permettent de combiner plusieurs comparaisons ensemble :

<b>&amp;&amp;</b>	<b>ET</b>
<b>  </b>	<b>OU</b>

Voir => [https://fr.wikipedia.org/wiki/Table\\_de\\_v%C3%A9rit%C3%A9](https://fr.wikipedia.org/wiki/Table_de_v%C3%A9rit%C3%A9)

# 1 Les conditions

## 1.1 Pourquoi utiliser les conditions ?

Pour exécuter du code **SI** une condition déterminée est remplie.

Exemple 1 : Nous voulons tester si un utilisateur est majeur ou non.

Exemple 2 : Nous souhaitons réaliser un outil qui renvoie à l'utilisateur le résultat du mélange de 2 couleurs primaires.

## 1.2 La syntaxe d'une condition

Les mots-clés utilisés pour construire une condition sont **if**, **else if**, et **else** (respectivement si, sinon si, sinon). L'écriture d'une condition se fait comme ceci :

```
if(conditionAVerifier) {  
  // Code à exécuter si la condition est vérifiée  
}  
else {  
  // Code à exécuter si la condition n'est pas vérifiée  
}
```

### Exemple 1 : Tester l'âge du visiteur

```
// On crée une variable age qui stocke la saisie de l'utilisateur  
var age = parseInt(window.prompt('Quel âge avez-vous ?'));  
  
// si l'utilisateur a 18ans ou plus alors on lui affiche un message "Vous êtes majeur"  
if(age >= 18) {  
  alert('Vous êtes majeur');  
}  
// sinon on lui affiche un message "Vous êtes mineur"  
else {  
  alert('Vous êtes mineur');  
}  
  
// pour exemple, ci-dessous le même code conditionnel avec la syntaxe ternaire  
// (age>=18) ? alert('Vous êtes majeur') : alert('vous êtes mineur');
```

Note : exécutez ce code directement depuis la console firebug ou Chrome ou sur jsfiddle : <https://jsfiddle.net/learnlab/7c6p0obt>

Dans le cas où vous avez plusieurs conditions à tester, alors utilisez un ou plusieurs **else if** à la suite de votre **if**.

Exemple 2 : un script qui renvoie le résultat du mélange de 2 couleurs primaires.

```
/*
Utilisation de ELSE IF et de son alternative SWITCH en cas de conditions multiples.
*/
// Nous commençons par déclarer 2 variables d'entrée (color1, color2)
// et 1 tableau result contenant les résultats possibles
var color1;
var color2;
var result = ['violet', 'orange', 'vert'];

color1 = window.prompt('Choisissez une couleur primaire (rouge, bleu, ou jaune)');
color2 = window.prompt('Choisissez une couleur de mélange (rouge, bleu, ou jaune)');

// si on mélange les couleurs bleu et rouge
if((color1=='rouge' && color2=='bleu') || (color1=='bleu' && color2=='rouge')) {
    alert('En mélangeant du rouge et du bleu, vous obtenez du ' + result[0]);
}
// sinon si on mélange les couleurs rouge et jaune
else if((color1=='rouge' && color2=='jaune') || (color1=='jaune' && color2=='rouge')) {
    alert('En mélangeant du rouge et du jaune, vous obtenez du ' + result[1]);
}
// sinon si on mélange les couleur bleu et jaune
else if((color1=='bleu' && color2=='jaune') || (color1=='jaune' && color2=='bleu')) {
    alert('En mélangeant du jaune et du bleu, vous obtenez du ' + result[2]);
}
// sinon
else {
    alert('vous n\'avez pas renseigné une couleur acceptée');
}
```

Note : exécutez ce code directement depuis la console Firebug ou Chrome ou sur jsfiddle : <https://jsfiddle.net/learnlab/3t8ha39e>

Dans le code précédent, notez que les opérateurs logiques **ET** et **OU** (**&&** et **||**) sont très utiles pour éviter de répéter des **if** et des **else if**.

### 1.3 La syntaxe du switch (remplace les else if)

Les parenthèses qui suivent le mot clé **switch** indiquent une expression dont la valeur est testée successivement par chacun des **case**. Lorsque l'expression testée est égale à une des valeurs suivant un **case**, la liste d'instruction qui suit (après les **:**), celui-ci est exécuté.

Le mot clé **break** indique la sortie de la structure conditionnelle.

Le mot clé **default** précède la liste d'instructions qui sera exécutée si l'expression n'est jamais égale à une des valeurs.

```
switch (variableATester) {  
  case 1: // signifie : dans le cas où variableATester == 1  
    // Code à exécuter si la condition est vérifiée;  
    break; // le break permet de sortir de du switch  
  
  case 2: // signifie : dans le cas où variableATester == 2  
    // Code à exécuter si la condition est vérifiée;  
    break;  
  
  case 3: // signifie : dans le cas où variableATester == 3  
    // Code à exécuter si la condition est vérifiée;  
    break;  
  
  default: // (facultatif) signifie : dans tous les autres cas cas  
    // Code à exécuter si la condition est vérifiée;  
    break;  
}
```

Note : Retrouvez le code précédent (script des couleurs primaires) construit avec un switch et des if imbriqués sur jsfiddle : <https://jsfiddle.net/learnlab/nr0fnxsc>