

## Les conditions

Jusqu'à présent tout ce que nous avons vu nous permet de créer des programmes tout à fait fonctionnels mais pas vraiment adaptables. Imaginez que vous ayez un programme qui fait sonner une alarme à 7h00 pour les jours où vous travaillez mais que vous ayez besoin d'un autre programme pour vous réveiller quand êtes en vacances. Pas très pratique n'est-ce pas ? Heureusement une solution existe en programmation qui va nous permettre d'adapter les actions que notre programme prendra selon des conditions qui seront vérifiées, c'est ce qu'on appelle *les conditions*.

La structure de condition usuelle se compose toujours d'un mot-clef (on utilisera « if » qui se traduit par « si » en Français) suivi d'une condition entre parenthèses (voir les possibilités plus bas) qui devra être vérifiée. A la suite on utilisera des accolades, qui serviront à délimiter toutes les instructions à exécuter si la condition est vraie. Ce qui pourrait donner :

Ex : **var temp = 103**

```
if (temp >= 100) {  
  
    console.log("L'eau boue !") // Affiche à l'écran "L'eau boue !" car la variable temp  
vaut plus que 99  
  
}
```

Mais si je veux afficher un autre message dans le cas où la température vaut moins de 100 degrés je suis obligé de refaire un « if » alors ? Eh bien non ! On peut effectuer également des actions dans le cas où notre condition n'est pas validée, on utilisera alors un autre mot-clef (« else » pour nous, qui signifie « sinon ») suivi lui aussi des accolades et ce « else » se trouvera juste après les accolades du « if ». Voici notre exemple précédent amélioré :

Ex : **var temp = 93**

```
if (temp >= 100) {  
  
    console.log("L'eau boue !") // N'affiche pas à l'écran "L'eau boue !" car la variable  
temp vaut moins que 99  
  
} else {  
  
    console.log ("L'eau ne boue pas.") // Affiche à l'écran "L'eau ne boue pas." car la  
variable temp vaut moins que 100
```

}

Pour écrire une condition on a accès à plusieurs possibilités qu'on appelle *des opérateurs de comparaisons*. Ce sont eux qui vont se mettre entre 2 valeurs ou variables. Voici la liste :

Opérateur	Description
===	Opérateur d'égalité
!==	Opérateur de différence
<	Opérateur de strictement plus petit
>	Opérateur de strictement plus grand
<=	Opérateur de plus petit ou égal
>=	Opérateur de plus grand ou égal

Une condition avec ces opérateurs est exactement comme un calcul sauf que le résultat est toujours soit *true* soit *false*. Ça vous rappelle quelque chose ? En effet le résultat avec ces opérateurs de comparaisons est une valeur de type *Booléen*. En fait notre « if » ne va exécuter une condition que si ce qui est entre les parenthèses s'évalue à une valeur qui équivaut *true* (« Vrai »).

Pourquoi ai-je dit « s'évalue à une valeur qui équivaut *true* » et non « à la valeur *true* » ? Car les valeurs qui ne sont pas booléennes peuvent être transformées en type booléen si la situation le demande (ou *converties* pour être correct) et il y a quelques règles pour savoir si le résultat sera « true » ou « false ».

1) Toutes les valeurs sont vraies sauf si elles font parties des quelques valeurs qui sont *fausses*. Restez-assis je ne suis pas en train de me moquer de vous.

2) Est toujours fausse toute valeur qui est : le numérique zéro (0), une chaîne de caractères vides ("" ou ""), le booléen false, et toute autre valeur définie comme tel dans le langage utilisé. (Par exemple en Javascript les valeurs « null », « undefined » et « NaN »)

Il existe également *des opérateurs logiques*. Ce sont des opérateurs qu'on peut utiliser pour créer des conditions complexes en combinant deux ou plusieurs conditions simples.

Symbole	Description
&&	ET
	OU

Ex : `if (jour === 'lundi' && conge === false) {`

- Opérations à faire

`} // Les opérations seront lancées si le jour est lundi est que ce n'est pas un congé.`

Dernière chose à savoir, il existe un dernier mot-clef pour notre structure de condition qui nous permet de vérifier d'autres conditions si la première se révèle fausse. Par exemple si vous vouliez effectuer des actions si le jour est lundi mais que vous vouliez effectuer d'autres actions si le jour est mardi ainsi que des actions par défaut pour le reste des jours. Notre mot clef est « ELSE IF » (combinaison des deux précédents qui nous donne « SINON SI ») et il est également suivi d'accolades.

Ex : `if (jour === 'lundi') {`

- Opérations à faire

`} else if (jour === 'mardi') {`

- Opérations à faire

`} else {`

- Opérations à faire

`}`

A noter que dans une structure de condition, seulement un seul bloc d'instructions sera exécuté, une fois qu'une condition est validée et les opérations lancées le programme passe immédiatement au code sous la structure.

Ex : `var profondeurPiscine = 24`

`if (profondeurPiscine > 20) {`

`console.log("C'est très profond.") // Est affiché car la condition est vraie`

`} else if (profondeurPiscine > 10) {`

```
        console.log("C'est plutôt profond.") // N'est pas affiché car la condition précédente  
a été validée
```

```
    } else {
```

```
        console.log("C'est pas tellement profond.") // N'est pas affiché pour la même  
raison
```

```
    }
```

### Quizz

Conseil : Attention à ne pas créer de conditions impossibles. (Par exemple "maVar > 10 && maVar < 8")

1) Quels sont les constituants possibles pour une structure de condition ?

2) Comment écrire une structure de condition qui affiche la string « 1337 code » si la variable « programmer » vaut true ? (Incluez l'affectation de la variable)

3) Comment écrire une structure de condition qui affiche la string « Ceci est un calcul » si la variable appelée mystery contient le calcul « 5 + 10 » et la string « Ceci est une chaîne de caractères » si la variable mystery contient la string « 5 + 10 » ?

4) Comment écrire une structure de condition qui affiche « Super ! » si la variable note est supérieur à 14, « C'est pas mal. » si la variable entre 10 inclus et 14 inclus ainsi que « Aïe, va falloir étudier. » si la variable est inférieur à 10 ?

5) Comment écrire un programme qui affiche « Relax » si le contenu d'une variable day contient un chiffre plus grand que 4 mais affiche également « C'est le week-end » si le chiffre est plus grand que 5 ?