

les conditions

Les conditions font partie des structures de contrôle.

Une condition est basée sur une proposition, elle peut donc être soit vraie, soit fausse.

Elle s'exprime à l'aide de l'instruction:

si <proposition>

alors <je fais ça>

(sinon <je fais autre chose>)

Dans le cas où la proposition est vraie, on passe dans le *alors*, dans le cas où la proposition est fausse, on passe dans le *sinon*.

A quoi ça sert ?

Suivant la valeur de vérité de certaines propositions, il sera possible d'adapter son algorithme.

Les conditions vont servir à tester les valeurs de certaines variables et ainsi permettre d'être plus précis dans nos algorithmes en exécutant des instructions différentes au cas par cas.

Explication technique

Une condition permet de créer un embranchement dans notre algorithme en proposant une alternative suivant certains cas.

```
if (expression) {  
    console.log("la valeur de vérité de l'expression est vraie")  
}  
else {  
    console.log("la valeur de vérité de l'expression est fausse")  
}
```

Il est ensuite possible d'imbriquer les if (if(...) { if(...) { ... } } else if() { ... } ^^

Explication technique

Lorsque l'on souhaite effectuer différentes instructions suivant la valeur d'une variables et que la liste des valeurs possible de cette variable est connue, il est possible d'utiliser le switch. Il évite une succession de plusieurs if et simplifie la lecture.

```
switch(variable){  
    case valeur1: { ... }  
    case valeur2: { ... }  
    default: { ... } // dans tous les autres cas
```

attention, on utilise le break dans les différents cas sinon toutes les instructions des cas suivant seront exécutées dès lors qu'un cas sera vrai.

<https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Instructions/switch>

Les comparateurs

Egalité :

`==` pour l'égalité de valeur [`1 == 1` est vrai ; `1 == "1"` est vrai]

`===` pour l'égalité de valeur et de type [`1 === 1` est vrai ; `"1" === "1"` est vrai ; `1 === "1"` est faux]

Comparaison

`<` [`1 < 2`]

`>` [`2 > 1`]

`<=` [`1 <= 2` ; `2 <= 2`]

`>=` [`2 >= 1` ; `2 >= 2`]

Différents

`!=`

`!==`

Pratique

exercice conditions