

Support de cours

Cours:

## Initiation à la programmation (en Java)

Vidéo:

### Init-JAVA-02-2-Conditions-pt2

Concepts (extraits des sous-titres générés automatiquement) :

**Évaluation de cette expression. Valeurs initiales. Utilisation de conditions simples. Cas d'un petit programme. Variante de la situation précédente. Opérateur de comparaison. Expressions de façon. Moyen d'opérateurs logiques. Moyen de conditions simples. Condition simple. Partie positive du if. Opérandes retourne. Moyen de petits exemples concrets. Opérateurs de comparaisons. Java des conditions simples.**



[vers la recherche de séquences vidéo](#)  
(dans Initiation à la programmation (en Java).)



[vers la vidéo](#)

Center for Digital Education. Plus de matériel de soutien pédagogique ici :

<https://www.epfl.ch/education/educational-initiatives/cede/educational-technologies-gallery/boocs-en/>

# Conditions

(Partie 2)

## Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s



```

→ int a = 1;
→ int b = 2;

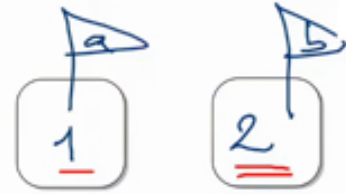
→ if (a == b) {
    System.out.println("Cas 1");
} else {
    System.out.println("Cas 2");
}

→ if (2 * a == b) {
    System.out.println("b est egal au double de a.");
}

```

*Handwritten annotations:*

- false* above the first `if` condition.
- true* above the second `if` condition.



$$2 * a \rightarrow 2$$

affiche

Cas 2

b est egal au double de a.

Illustrons maintenant, au moyen de petits exemples concrets, l'utilisation de conditions simples en Java. Nous avons ici le cas d'un petit programme qui commence par déclarer deux variables `a` et `b`, en leur affectant des valeurs initiales, respectivement 1 et 2. Nous souhaitons maintenant comparer les valeurs de `a` et `b`, pour savoir si elles sont égales, et pour cela nous avons utilisé une condition simple qui utilise l'opérateur de comparaison `==`. Ici les valeurs de `a` et `b` sont différentes, ce qui aura comme conséquence que l'évaluation de cette expression retournera `"false"`. Nous allons donc brancher sur la partie `else` de l'instruction conditionnelle, et donc exécuter cette instruction, qui aura pour conséquence d'afficher le message `"Cas 2"`, ce que nous voyons ici. Donc continuons ensuite l'exécution en séquence, et nous souhaitons maintenant comparer la valeur de deux fois `a` avec la valeur de `b`. Notons au passage que, au moyen de conditions simples, nous pouvons comparer non seulement les valeurs de deux variables, comme c'est le cas ici, mais également les valeurs de deux expressions de façon plus générale. Si j'évalue deux fois `a`, j'obtiens évidemment la valeur 2, la valeur 2 qui est évidemment égale à la valeur de `b`, par conséquent l'évaluation de cette expression retournera `"true"`, cette instruction, l'instruction qui suit, va donc être exécutée, et nous aurons donc l'affichage de `"b est égal au double de a."` Notez ici nous avons affaire à une expression qui est relativement simple, lorsque les expressions se complexifient, pour formuler les conditions, il est recommandé de parenthéser les termes des expressions pour les rendre plus lisibles.

notes

résumé

0m 1s



```
int a = 1;
```

```
int b = 2;
```

```
if (a != b) {
```

true

```
    System.out.println("Cas 2");
```

```
    } else {
```

```
        System.out.println("Cas 1");
```

```
    }
```

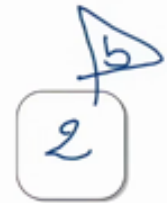
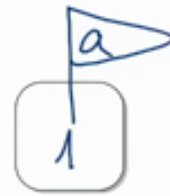
```
if (2 * a != b) {
```

```
    System.out.println("b est different du double de a.");
```

```
}
```

affiche

Cas 2



$$2 * a \rightarrow 2$$

Par exemple ici, nous aurions pu commencer à le faire en ajoutant une paire de parenthèses autour de ce terme. Variante de la situation précédente, nous voulons maintenant comparer les valeurs de a et b pour savoir si a est différent de b. Comme c'est le cas, l'évaluation de cette expression va retourner "true", nous allons donc brancher sur la partie positive du if, la partie vraie, et afficher "Cas 2".

notes

résumé

1m 49s





Si l'on continue en séquence, nous voulons maintenant comparer la valeur de deux  $a$  avec celle de  $b$  pour savoir si deux fois  $a$  est différent de  $b$ . Ça n'est pas le cas, donc l'évaluation de cette expression va retourner "false", par conséquent nous allons brancher l'exécution après l'instruction `if` puisqu'il n'y a pas de bloc `else`, donc rien de s'affiche. Nous venons donc de voir comment il est possible d'exprimer en Java des conditions simples en utilisant des opérateurs de comparaisons. En pratique, il est souvent nécessaire de formuler des conditions

#### notes

#### résumé

2m 13s



# Les opérateurs logiques

On peut relier des conditions simples par des opérateurs logiques.

L'opérateur logique `&&` (ET):

par exemple, la condition

(a < b) `&&` (c < d)

est vraie **uniquement** si les deux conditions (a < b) et (c < d) sont toutes les deux vraies.

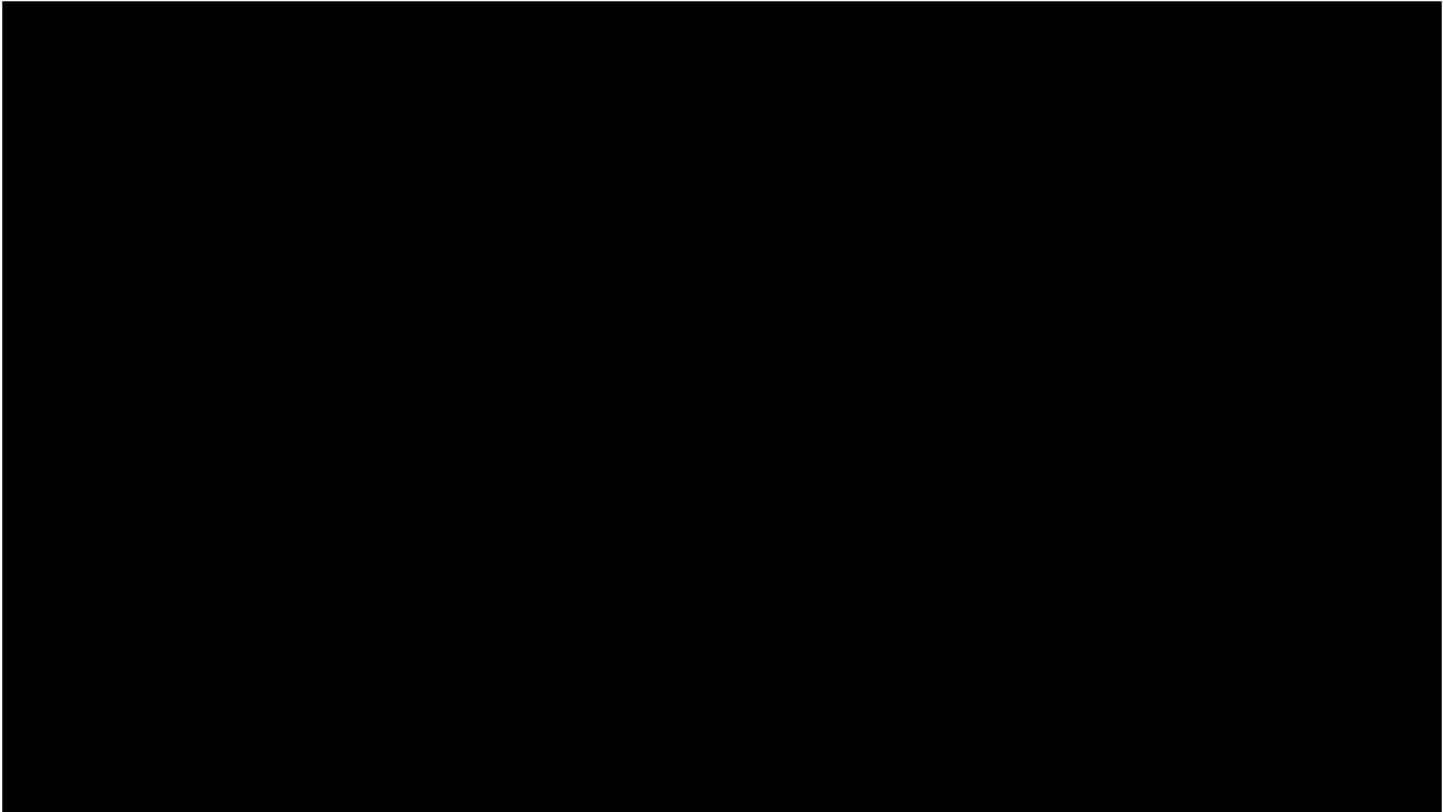
plus complexes, et donc de combiner des conditions simples. Ceci se fait au moyen d'opérateurs logiques. Par exemple, l'opérateur logique ET, qui en Java s'écrit au moyen du double symbole `&&`, permet de vérifier si deux conditions sont simultanément vérifiées. On voit donc que, l'opérateur logique

notes

résumé

2m 49s





requiert l'utilisation de deux opérandes, l'évaluation de l'expression avec le ET retournera "true" si et seulement si l'évaluation de chacun des opérandes retourne également "true". On voit donc qu'il suffit que l'un des deux opérandes retourne "false", par exemple celui-ci, et a fortiori les deux, bien sûr, pour que l'ensemble de l'expression retourne également "false". retourne également "false".

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

3m 13s

