

Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-02-2-Conditions-pt1

Concepts (extraits des sous-titres générés automatiquement) :

Petit détail syntaxique. Exemples de conditions. Notion de condition de façon. Valeur d'une variable. Séquence vidéo précédente. Telle instruction. Opérateurs de comparaison. Petit exemple. Condition simple. Exemple d'instruction de branchement. Variables x. Genre de tournure. Façon analogue. Petite mise. Instruction conditionnelle.



[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 1)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s



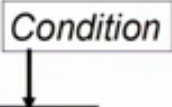
notes

0m 1s



Les conditions

L'instruction `if` fait apparaître une **condition** entre parenthèses



```
→ if (n < 5) {  
    System.out.println("Votre nombre est plus petit que 5.");  
} else {  
    System.out.println("Votre nombre est plus grand ou egal a 5.");  
}
```

Attention, la condition est toujours entourée de parenthèses.

Pour l'instant, nous n'avons rencontré qu'une condition simple, `n < 5`

Nous allons voir maintenant comment s'écrivent les conditions d'une façon générale.

petit peu plus détaillée et voir qu'en Java, il est possible d'exprimer des conditions relativement complexes. Vous avez donc ici sous les yeux un exemple d'instruction de branchement conditionnel tel que nous avons déjà eu l'occasion d'en rencontrer, et nous voyons que pour écrire cette instruction conditionnelle, nous avons besoin d'exprimer une condition, ce que nous voyons ici. La condition ici est une condition simple, qui consiste à comparer

notes

résumé

0m 25s



Les co

Les op

Attention:

L'opérateur pour tester si deux valeurs sont égales s'écrit avec deux signes égal ==

Une condition

Elle utilise un

Opérateurs d

Un seul signe = représente l'affectation

Par exemple, si on veut tester si la variable `n` est égale à 5, il faut écrire:

```
if (n == 5)
```

et non pas:

```
if (n = 5)
```

==

<=

>=

!=

égal à

inférieur ou égal à

supérieur ou égal à

différent de

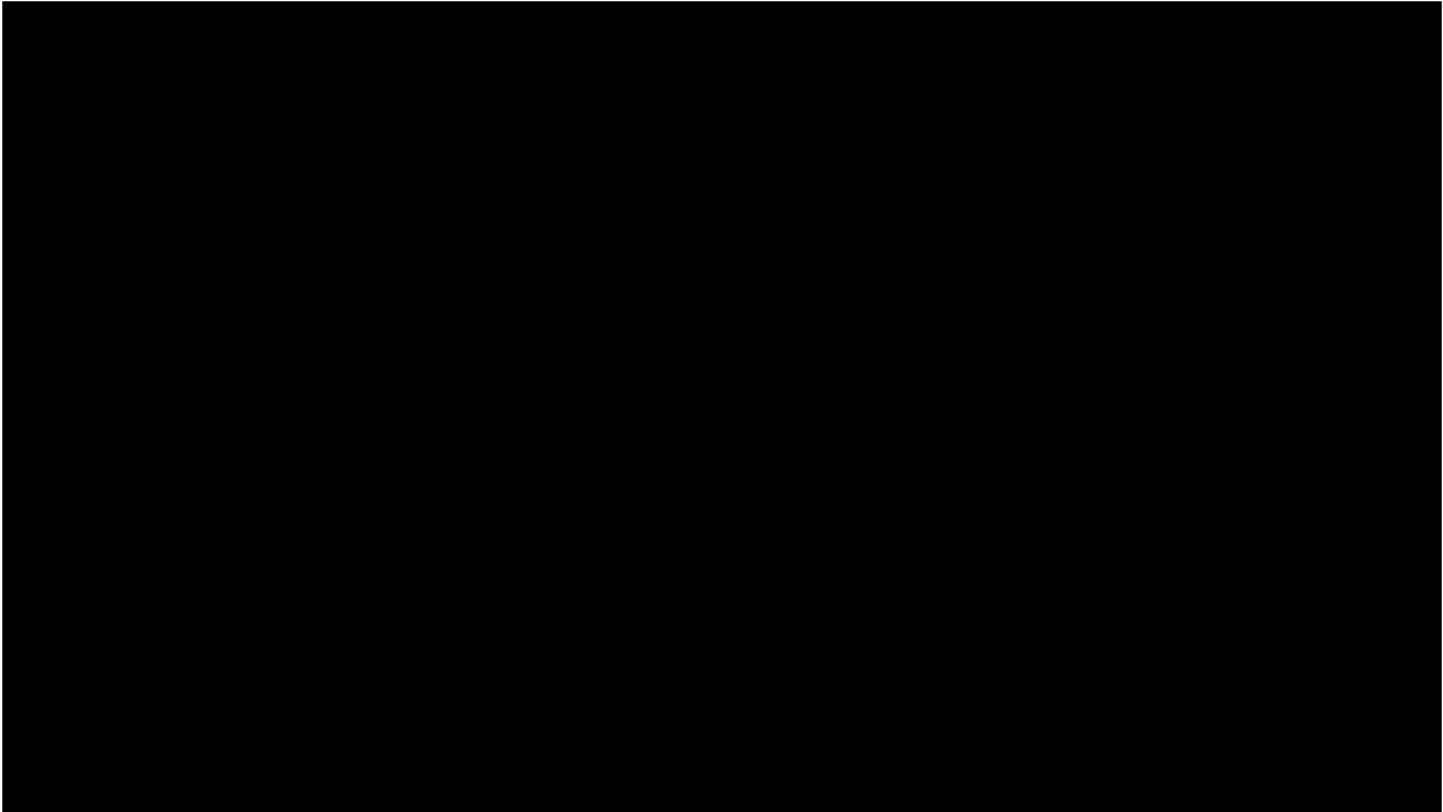
la valeur d'une variable `n` avec une constante, la constante 5. Petit détail syntaxique: les conditions dans une instruction conditionnelle sont toujours en Java entourées de parenthèses, ce que nous voyons concrètement ici, dans ce petit exemple. Commençons d'abord par examiner de façon un petit peu plus exhaustive comment on peut, en Java, exprimer des conditions simples. La vocation des conditions simples est de comparer deux expressions, et pour cela nous allons utiliser des opérateurs de comparaison. Nous avons déjà eu l'occasion d'en rencontrer à plusieurs reprises dans des exemples, comme inférieur ou supérieur. Java, en fait, offre une palette d'opérateurs de comparaisons prédéfinis, par exemple l'opérateur inférieur permet de comparer si la valeur d'une entité est inférieure à la valeur d'une autre entité. Supposons par exemple que je dispose de deux variables `x` et `y`, et que je souhaite savoir si la valeur de `x` est inférieure à celle de `y`, j'utiliserai l'opérateur de comparaison inférieur pour réaliser ce test, pour exprimer cette condition. De façon analogue, je peux utiliser l'opérateur `==` pour vérifier si deux entités ont la même valeur. Par exemple, à nouveau si j'ai deux variables `x` et `y`, si je veux tester que la valeur de `x` est la même que la valeur de `y`, j'utiliserai `==` et j'exprimerai ma condition de cette façon. De façon analogue, nous disposons des tenants supérieur ou égal, inférieur ou égal ou différent. Ce qu'il faut savoir, c'est que l'évaluation d'une telle expression retourne à un résultat booléen, c'est-à-dire la valeur "true" pour vrai ou encore la valeur "false" pour faux. Petite mise en garde de rigueur: l'opérateur de comparaison permettant de tester si deux valeurs sont égales s'exprime au moyen de deux symboles égal. À ne pas confondre donc avec le symbole `=` unique, qui lui représente l'affectation.

notes

résumé

0m 49s





Ici dans cet exemple, je teste si la valeur de n vaut 5. Ce n'est pas ce que je fais quand j'écris `n = 5`, qui signifie que j'affecte la valeur 5 à la variable n. Il faut toutefois noter que ce genre de tournure n'est pas tolérée par le compilateur, en Java, et donc ici vous aurez un échec de compilation, si vous rédigez ce genre d'instruction. d'instruction.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

2m 49s



.....

.....

.....

.....

.....