

Support de cours

Cours:

## Initiation à la programmation (en Java)

Vidéo:

### Init-JAVA-02-1-Branchements-pt3

Concepts (extraits des sous-titres générés automatiquement) :

**Bloc d'instructions. Bloc optionnel. Bloc de conditions vraies. Premier test. Branchements conditionnels. Blocs d'instructions. Bout de programme java. Bloc de conditions fausses. Deuxième if. Instructions if. Bloc du else. Effet vrai. Condition. Fait possible. Accolades rondes.**



[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/>

# Branchements conditionnels

## (Partie 3)

### Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

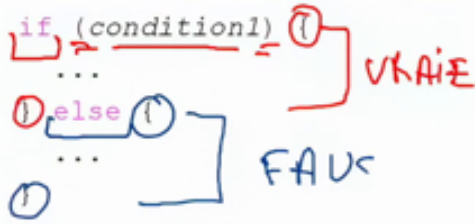
résumé

0m 0s



# Les choix imbriqués

L'instruction `if` suit donc le schéma:



Les instructions figurant dans les blocs sont absolument quelconques. *Il peut donc s'agir d'autres instructions `if`.*

Les branchements conditionnels en Java suivent donc le schéma général suivant: on commence par le mot réservé `if`. Ensuite entre accolades rondes ici, on écrit une condition que l'on veut évaluer. Ensuite, on met un bloc d'instructions ici qui sera évalué si la condition, qui sera exécuté si la condition est évaluée à vrai, puis éventuellement on peut mettre ici un bloc optionnel `else` ici avec un bloc

## notes

## résumé

0m 1s



# Choix imbriqués: exemple



```

if (x == y) {
    if (y == z) {
        System.out.println("Les trois valeurs sont egales.");
    } else {
        System.out.println("Seules les deux premieres valeurs sont egales.");
    }
} else {
    if (x == z) {
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");
    } else {
        if (y == z) {
            System.out.println("Seules les deux dernieres valeurs sont egales.");
        } else {
            System.out.println("Les trois valeurs sont differentes.");
        }
    }
}

```

d'instructions qui sera exécuté si la condition est évaluée comme fausse. Ces blocs d'instructions peuvent contenir n'importe quel bout de programme Java et en particulier peuvent aussi contenir d'autres instructions if. C'est ce qu'on appelle à ce moment-là des choix imbriqués. Prenons un exemple. Supposons que l'on ait trois variables x, y et z dont on veuille tester l'égalité par exemple est-ce que x est égal à y, est-ce que y est égal à z ou est-ce qu'elles sont les trois égales? Ce qu'on va pouvoir faire, c'est donc commencer par un bloc ici d'instructions qui va tester si x est égal à y. Ce bloc s'étend jusqu'ici avec ici son else associé et le bloc du else. Et donc dans le bloc par exemple, contrôlé par le premier test, lorsque le premier test est vrai, on voit ici, on a de même un deuxième if, ce qui est tout à fait possible, lui-même donc avec ici son bloc de conditions vraies et son bloc de conditions fausses. De même dans le premier else, on voit ici de nouveau un if avec son bloc de conditions vraies et son bloc de conditions fausses et on voit que dans le dernier bloc de conditions, on peut même avoir un dernier if.

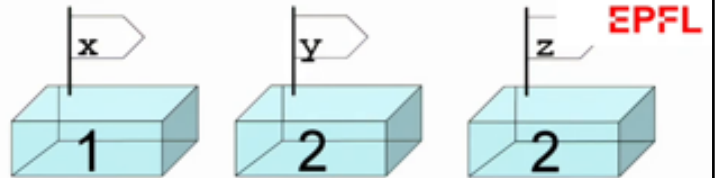
## notes

## résumé

0m 37s



Supposons:



```
if (x == y) {  
    if (y == z) {  
        System.out.println("Les trois valeurs sont egales.");  
    } else {  
        System.out.println("Seules les deux premieres valeurs sont egales.");  
    }  
} else {  
    if (x == z) {  
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");  
    } else {  
        if (y == z) {  
            → System.out.println("Seules les deux dernieres valeurs sont egales.");  
        } else {  
            System.out.println("Seules les deux dernieres valeurs sont egales. est affiche");  
        }  
    }  
}
```

Alors vous comprenez bien que ce genre de construction, il ne faut pas en abuser. Ça devient vite un petit peu difficile à comprendre, difficile à lire. Au-delà de trois niveaux, je trouve que ça devient difficilement compréhensible, sinon bien totalement illisible. Regardons maintenant l'exemple vraiment en détail pas à pas. Supposons qu'on ait ici trois variables : x, y et z, qui contiennent chacune la valeur 1. Donc on commence par la première instruction if ici qui va tester: "est-ce que x est égal à y?" On a ici effectivement un 1 qui est égal à un 1. La condition est vraie, ce qui fait que l'on arrive à ce deuxième, cette deuxième instruction if ici qui va évaluer est-ce que la valeur de y est égal à la valeur de z. Donc on repose la question ici: est-ce que un est égal à un, ce qui est en effet vrai ce qui fait qu'on va entrer dans le bloc ici et afficher donc le message : "les trois valeurs sont égales". Puis, on va continuer l'exécution à la fin du bloc en question. Ici, il n'y a rien de particulier à faire. Puis, on va aussi continuer l'exécution après ce bloc ici. C'est-à-dire qu'on arrive ici à la fin du programme. Regardons maintenant ce qui se passe lorsque l'on a, par exemple, les valeurs y et z qui valent toutes les deux 2 et qui sont différentes de la valeur de x. Le programme commence donc comme tout à l'heure par évaluer est-ce que la valeur de x est égale à la valeur de y. Et dans ce cas-ci, bien sur, la réponse est fausse et donc la condition ici étant fausse, on va exécuter le bloc else correspondant. Donc, on rentre ici dans le bloc else du bas du programme, ce qui fait que on va rentrer ici dans le test de cette deuxième condition : «est-ce

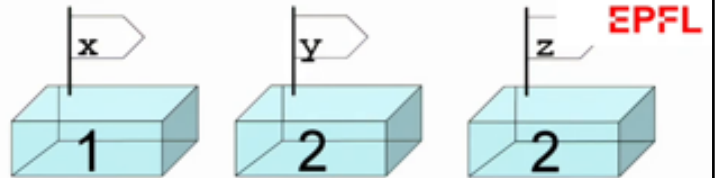
notes

résumé

1m 49s



Supposons:

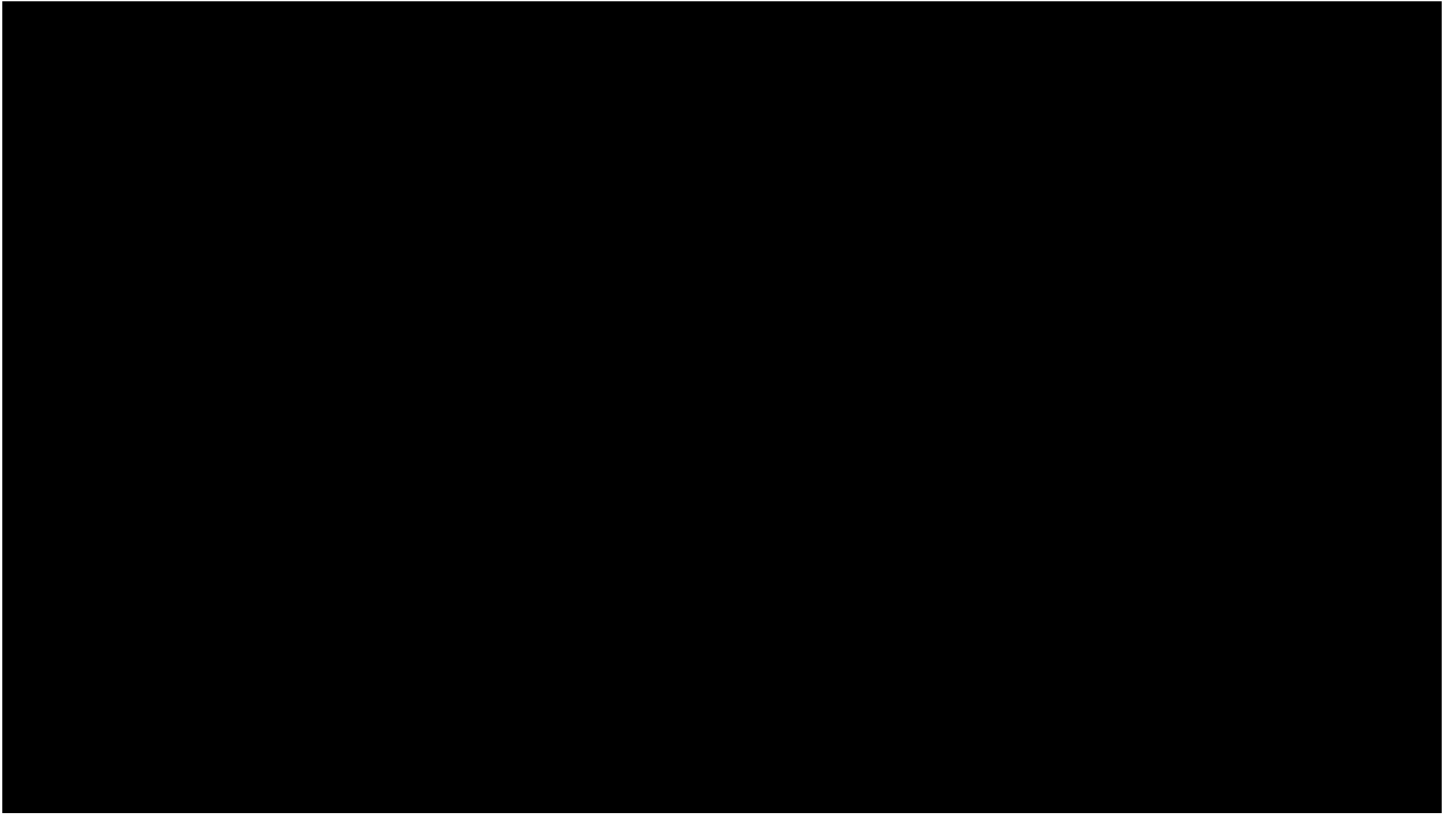


```
if (x == y) {  
    if (y == z) {  
        System.out.println("Les trois valeurs sont egales.");  
    } else {  
        System.out.println("Seules les deux premieres valeurs sont egales.");  
    }  
} else {  
    if (x == z) {  
        System.out.println("Seules la 1ere et la 3eme valeurs sont egales.");  
    } else {  
        if (y == z) {  
            → System.out.println("Seules les deux dernieres valeurs sont egales.");  
        } else {  
            System.out.println("Seules les deux dernieres valeurs sont egales. est affiche");  
        }  
    }  
}
```

que... ..la valeur de x (1) est égale à la valeur... ..de z (2) ?» Bien sur cette condition est fausse et donc ce if ici va lui-même se brancher sur son bloc else et donc on va se retrouver à évaluer cette troisième condition : est-ce que y a une valeur qui est égale à z ? 2 est égal à 2 : ici, on a bien sûr une condition qui est vraie ; ce qui fait que cette condition, ce bloc «if», ce branchement conditionnel va exécuter la condition, le bloc ici conditionné par le fait que la condition est vraie et va donc afficher : "seules les deux dernières valeurs sont égales".

notes

résumé



Puis on va continuer l'exécution à partir de là. On va continuer après ce bloc ici qu'on est en train d'exécuter. Il n'y a rien de particulier. Puis on va continuer l'exécution après ce branchement conditionnel ici. Il n'y a rien de particulier. Puis on va continuer l'exécution après ce branchement ici et on arrive donc à la fin de cet exemple. et on arrive donc à la fin de cet exemple.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

4m 13s

