

Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-03-3-IterationsQuizz-pt3

Concepts (extraits des sous-titres générés automatiquement) :

Première boucle for. Instruction d'incrémentation. Variable j. Variable i. Boucle for. Sortir de la boucle. Deuxième boucle for. Bonne réponse. Suite de cette boucle for. Suite de la boucle. Instruction print. Réponse a. Fin de la boucle for. Instruction for. Fin de la boucle for i.



[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/>

Itérations : quizz

(Partie 3)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s



Que s'affiche-t-il quand on exécute le code :

```
→ for(int i = 0; i < 3; ++i) {
  → for(int j = 0; j < i; ++j) {
    → System.out.print(j);
  }
  System.out.println("");
}
```

Handwritten notes:

i: 0 1 2
 ↑
 j: 0 1
 ↑
 0

A:
 0
 01

C:
 rien

B:
 0
 01
 012

D:
 0123
 0123
 0123

La bonne réponse est la réponse A et nous allons détailler pourquoi. Comme avant, la première boucle for initialise une variable i à zéro, la condition est i strictement inférieure à trois et l'instruction d'incrément est ++i. Encore une fois, comme avant, i va prendre les valeurs 0, 1 et 2. On commence par rentrer dans la boucle for i avec i qui vaut zéro et on exécute cette instruction-ci qui est une deuxième boucle for qui initialise une variable j à zéro, et la condition est j strictement inférieure à i. i vaut zéro, j vaut zéro également. Cette condition est fausse et on sort tout de suite de cette boucle for pour arriver ici. Arriver ici, ça veut simplement dire qu'on fait un retour à la ligne que je vais représenter de cette façon-ci. On revient à cette instruction-ci. i va donc maintenant prendre la valeur un, j va être initialisée à zéro, la condition est maintenant j strictement inférieure à i qui vaut un, donc j va prendre les valeurs ou plus exactement la valeur zéro et on va s'arrêter tout de suite de la boucle. Pour l'instant, j à la valeur zéro. On va exécuter cette instruction-ci, c'est-à-dire afficher la valeur de j qui est zéro. On sort de la boucle pour arriver à cette instruction-ci qui fait un retour à la ligne et on arrive à la fin de la boucle for i et on va revenir donc ici. i va donc prendre la valeur deux. On rentre dans cette boucle for i pour exécuter cette instruction for, qui, encore une fois, initialise j à zéro. La condition est maintenant j strictement inférieure à deux, donc j va prendre les valeurs zéro et un. On commence avec la valeur zéro dans j et on exécute cette instruction print,

notes

résumé

0m 1s



Que s'affiche-t-il quand on exécute le code :

```

-> for(int i = 0; i < 3; ++i) {
  -> for(int j = 0; j < i; ++j) {
    -> System.out.print(j);
  }
  - System.out.println("");
}

```

A:

0
01

C:
rien

B:

0
01
012

D:

0123
0123
0123

Handwritten notes illustrating the execution of the nested loops:

```

i: 0 1 2
   ↑
j: 0 1
   ↑
  0 1
  0 1

```

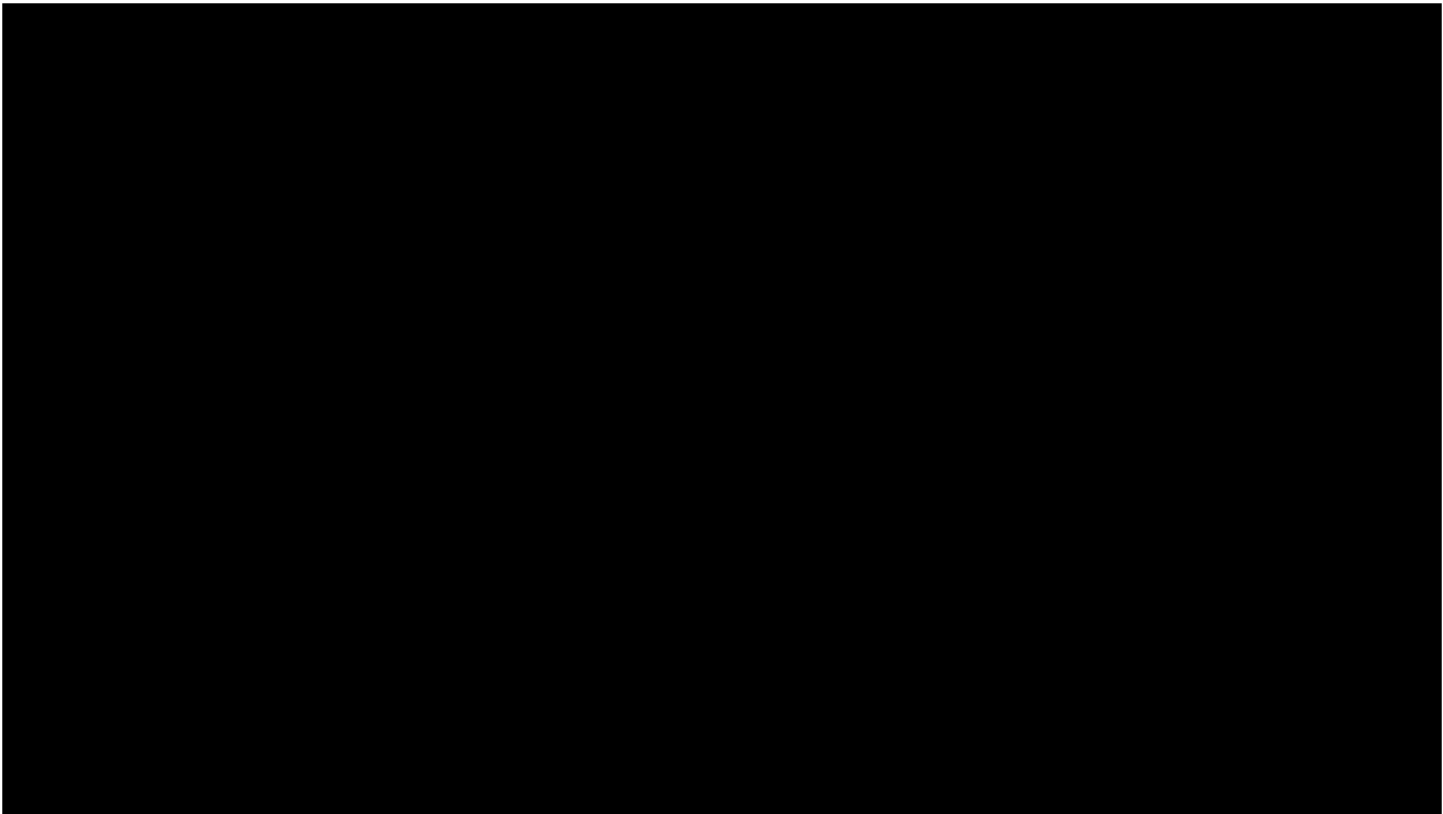
c'est-à-dire qu'on va afficher la valeur de j qui est zéro. On arrive à la fin de la boucle for j et on revient ici avec j qui va maintenant valoir un. On va exécuter cette instruction qui affiche la valeur de j, c'est-à-dire un, et on va sortir de la boucle.

notes

résumé

2m 49s





C'est-à-dire exécuter cette instruction-ci qui fait un retour à la ligne. On arrive à la fin de la boucle for i. i vaut maintenant trois, cette condition est fausse et on va sortir de la boucle pour arriver à la fin du code. On va donc bien afficher le résultat qui était donné. le résultat qui était donné.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

3m 13s

