

Support de cours

Cours:

Initiation à la programmation (en Java)

Vidéo:

Init-JAVA-03-5-Blocs-pt1

Concepts (extraits des sous-titres générés automatiquement) :

Utilisation des variables. Notion de bloc. Séquence vidéo précédente. Portée des variables. Instructions java. Structures de contrôle. Accolade ouvrante. Accolade fermante. Grande autonomie. Propres variables. Variable j. Variable keyb de type scanner. Variable i de type int. Instruction if. Différentes parties du programme.



[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/>

Fonctions : blocs

(Partie 1)

Initiation à la programmation (Java)

Jamila Sam, Vincent Lepetit et Jean-Cédric Chappelier

...

notes

résumé

0m 0s





Dans une séquence vidéo précédente, nous avons vu les structures de contrôle, c'est-à-dire ces instructions Java qui permettent aux données d'influencer,

notes

résumé

0m 1s



Les blocs

En Java, les instructions peuvent être regroupées en **blocs**.

Les blocs sont identifiés par des délimiteurs de début et de fin : **{ et }**

Exemple:

```
{
Scanner keyb = new Scanner(System.in);
int i;
double x;

System.out.println("Valeurs pour i et x : ") ;
i = keyb.nextInt();
x = keyb.nextDouble();
System.out.println("Vous avez saisi : i = " + i +
                    ", x = " + x);
}
```

une conséquence sur l'utilisation des variables, c'est ce que l'on appelle la portée des variables que nous allons maintenant détailler dans cette séquence-ci. En Java les instructions peuvent donc être regroupées en bloc, et ceci indépendamment des structures de contrôle. Il suffit pour cela de regrouper donc les instructions entre une accolade ouvrante et une accolade fermante. Par exemple, ici, j'ai dans un bloc, donc avec une accolade ouvrante et une accolade fermante, déclaré une variable keyb de type scanner, une variable i de type int, une variable x de type double et je peux

notes

résumé

0m 25s



Les blocs

Les blocs ont en Java une grande autonomie.

Ils peuvent contenir leurs propres déclarations et initialisation de variables:

```
→ if (i != 0) {
→   int j = 0;

    ...
    j = 2 * i;
    ...
}
// A partir d'ici, on ne peut plus utiliser j
```

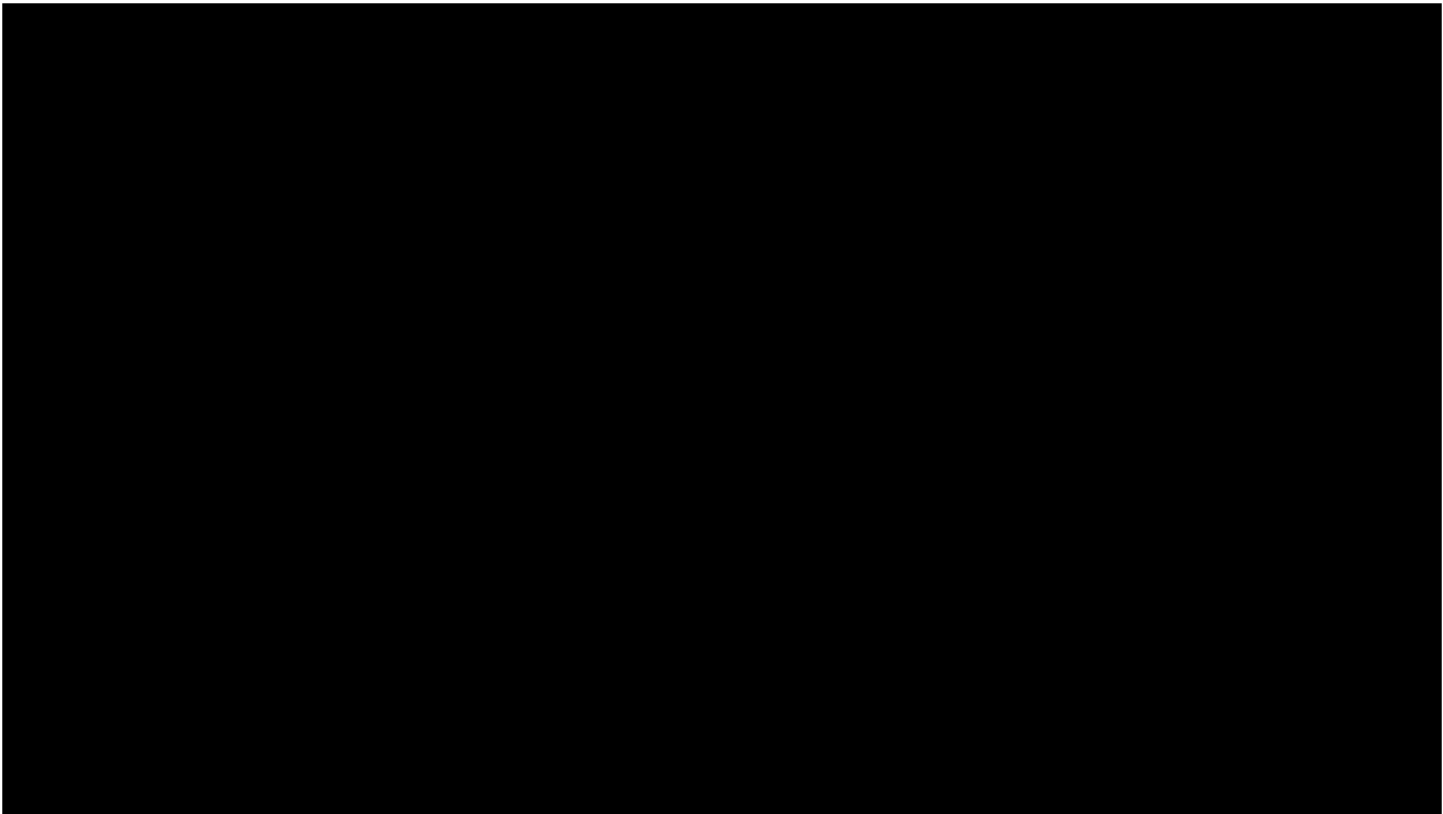
par exemple, les utiliser pour lire la valeur de i, lire la valeur de x ou par exemple pour afficher la valeur de i, la valeur de x. Ces variables appartiennent au bloc dans lequel elles ont été déclarées. Vous voyez donc qu'en Java, les blocs ont une grande autonomie, au sens où ils peuvent contenir leurs propres variables. Un deuxième ici, plus simple, où on a un bloc contrôlé par une instruction if. Le bloc commence donc ici, et se termine par l'accolade fermante ici, et dans lequel on déclare une variable j de type entier, que l'on peut donc ensuite utiliser dans ce bloc.

notes

résumé

1m 1s





Mais une fois que l'on a quitté le bloc, à partir de cet endroit-là, on ne peut plus utiliser cette variable j qui appartenait au bloc en question. appartenait au bloc en question.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

1m 37s



.....

.....

.....

.....

.....