

Support de cours

Cours:

## Initiation à la programmation (en C++)

Vidéo:

### Blocs d'instructions (partie 1)

Concepts (extraits des sous-titres générés automatiquement) :

**Utilisation des variables. Instructions c. Portée des variables. Structures de contrôle. Propres variables locales. Variable de type entier. Séquence vidéo précédente. Variable de type double. Accolade ouvrante. Variable locale. Blocs. Grande autonomie. Réponse de l'utilisateur. Séquence d'instructions. Instruction if.**



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



[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 (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s





Dans une séquence vidéo précédente, nous avons vu comment les structures de contrôle, c'est à dire les instructions C++ qui

notes

.....

.....

.....

.....

.....

.....

.....

.....

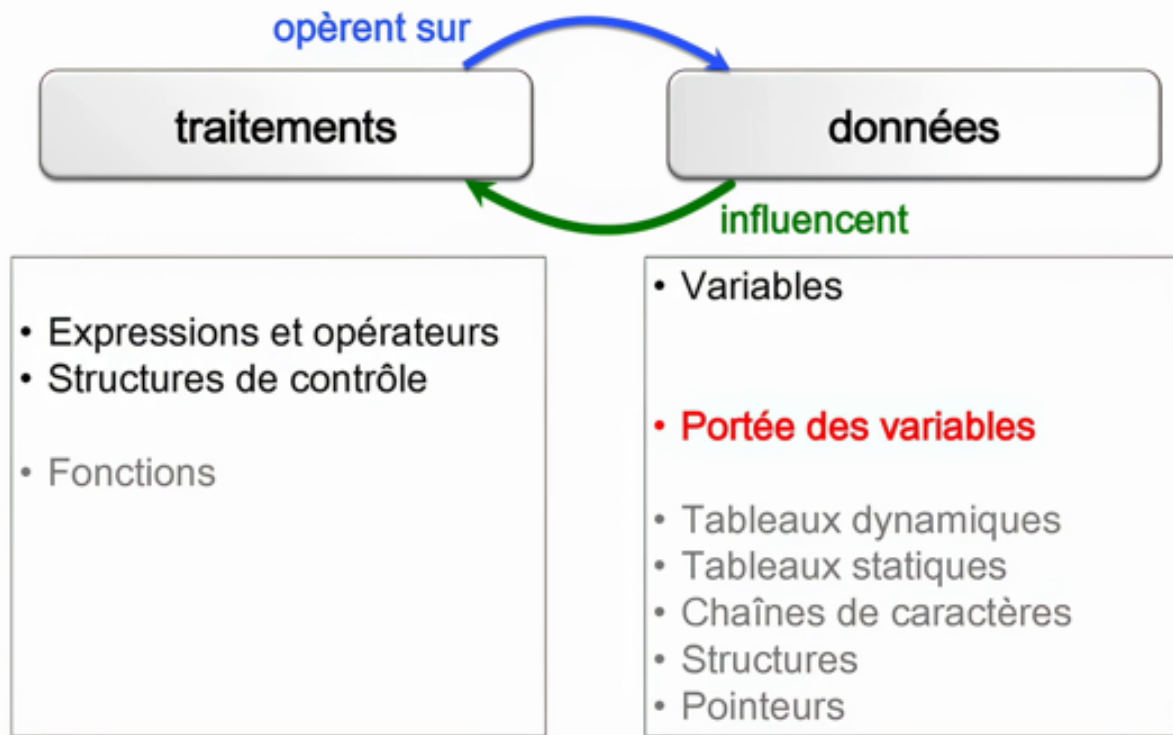
.....

.....

résumé

0m 1s





permettent aux données d'influencer, de choisir, de répéter les traitements effectués, comment ces structures de contrôle, justement, utilisaient des blocs. Ces blocs vont avoir un impact sur l'utilisation des variables,

notes

résumé

0m 13s



# Les blocs

Les blocs ont en C++ 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
```

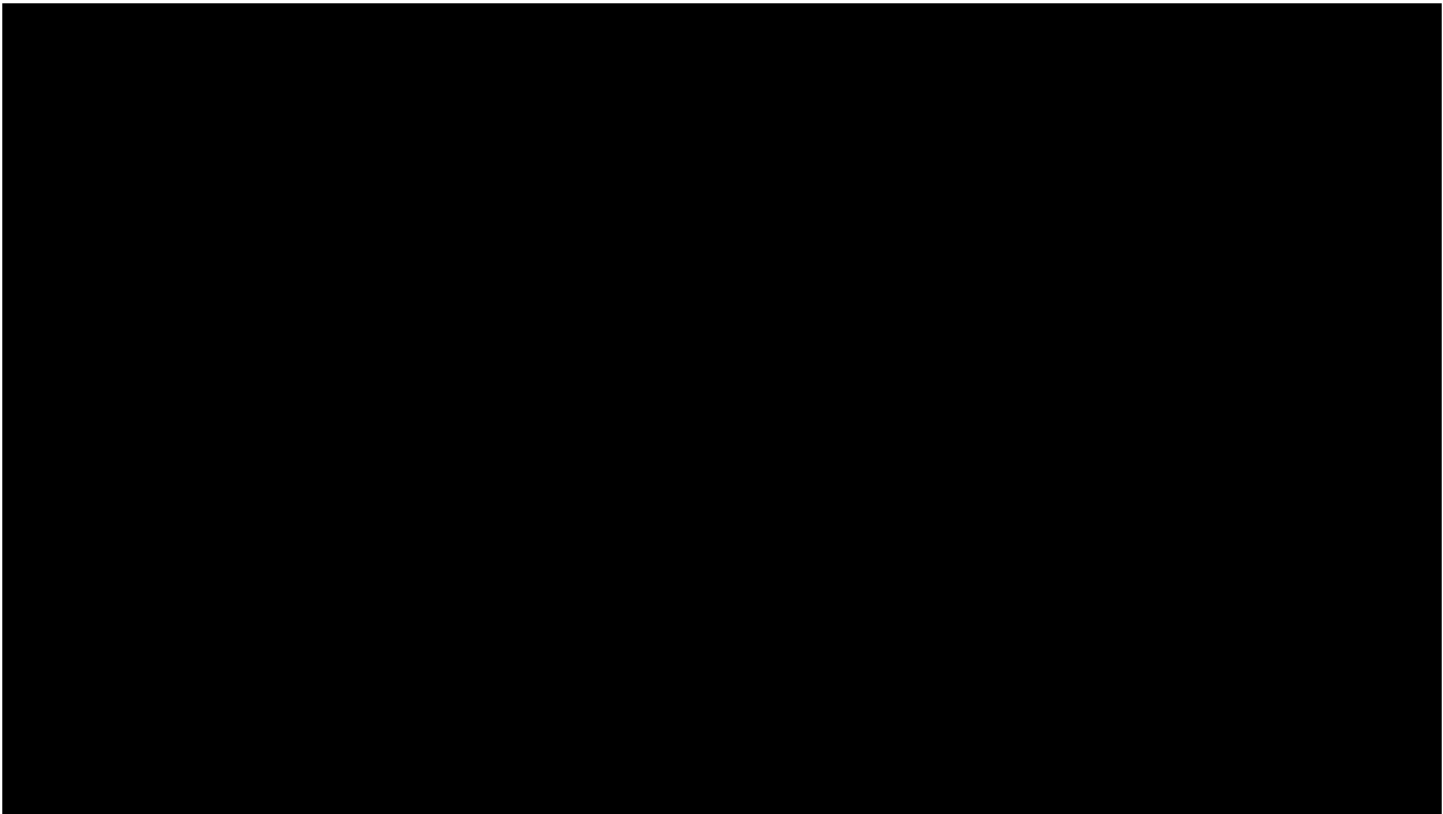
c'est ce que l'on appelle la portée des variables et que nous allons étudier donc dans cette séquence-ci. En C++, les instructions peuvent donc être regroupées dans des blocs et ceci indépendamment de toute structure de contrôle; il suffit pour cela d'avoir une accolade ouvrante et une accolade fermante qui entourent une séquence d'instructions. Par exemple, je peux donc ici commencer un bloc, déclarer une variable de type entier qui s'appelle i, déclarer une variable de type double qui s'appelle x, afficher un message pour pouvoir saisir la réponse de l'utilisateur pour saisir les valeurs de i et de x, et puis ensuite donc afficher ce qui a été saisi sur le terminal; et tout ceci donc dans un bloc qui est terminé, ici, par une accolade fermante. Les blocs en C++ ont vraiment une très grande autonomie, ils peuvent ainsi déclarer leur propres variables locales comme on a vu précédemment, ou même dans bloc ici associé à une instruction if, on peut tout à fait déclarer une variable, ici, j, qui va être utilisée dans le bloc en question et cette variable est locale, en ce sens que une fois qu'on a terminé le bloc, on ne peut plus utiliser j à partir du moment où on

notes

résumé

0m 25s





a quitté le bloc dans lequel elle avait été définie. Une variable comme ceci qui est définie dans un bloc, c'est ce qu'on appelle une variable locale, donc la variable j précédemment est une variable locale au bloc ici contrôlée par l'instruction if. l'instruction if.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

1m 37s



.....

.....

.....

.....

.....