

Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

Variables (partie 2)

Concepts (extraits des sous-titres générés automatiquement) :

Identificateur de la variable. Ligne de cette nature. Variable n. Type entier. Déclarations de variables. Valeur initiale. Variable. Travers de cet identificateur. Éléments de terminologie. Deuxième façon. Type de résultats. Variable de type double. Moment de la déclaration. Schéma de cette nature. Type.



[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/>
page 1/10

Variables

(Partie 2)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s



Déclarations de variables

Les lignes:

```
int n(4);  
int n_carre;
```

sont des **déclarations de variables**.

Une déclaration de variable permet de créer une variable.

Lorsque je rédige une ligne de cette nature dans un programme,

notes

résumé

0m 1s



Déclarations de variables

Les lignes:

```
int n(4);  
int n_carre;
```

identificateur

sont des **déclarations de variables**.

Une déclaration de variable permet de créer une variable.

comme je viens de le faire dans un éditeur, je suis en fait en train de déclarer une variable. Je déclare dans mon programme, que je souhaite utiliser une variable qui s'appelle n, pour reprendre les éléments de terminologie de tout à l'heure, l'identificateur de la variable est ici n.

notes

résumé

0m 5s



Déclarations de variables

Les lignes:

ty
↓
identificateur
↓
→ `int n(4);`
`int n_carre;`

sont des **déclarations de variables**.

Une déclaration de variable permet de créer une variable.

C'est au travers de cet identificateur que je vais pouvoir accéder à la donnée stockée dans cette variable. Et à cette donnée, est associé un type qui est le type entier,

notes

résumé

0m 25s



Déclarations de variables

Les lignes:

type
↓
→ int *identificateur*
n(4);
→ int n_carre;

sont des **déclarations de variables**.

Une déclaration de variable permet de créer une variable.

qui en c++ s'écrit int. En c++, je ne suis pas dans l'obligation absolue de donner une valeur initiale à ma variable quand je la déclare. Je peux déclarer une variable en la laissant vide de contenu,

notes

résumé

0m 35s

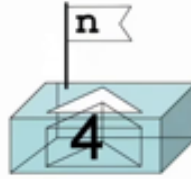


Initialisation

En même temps qu'elle est déclarée, une variable peut être initialisée, c'est-à-dire lui donner une valeur avant de l'utiliser.

La ligne:

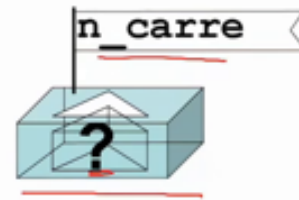
```
int n(4);
```



déclare donc une variable appelée `n` et lui donne la valeur 4.

Si une variable n'est pas initialisée, elle peut contenir n'importe quelle valeur!

```
→ int n_carre;
```



Il faut toujours initialiser une variable avant d'utiliser la valeur qu'elle contient !

ce que je fais ici. Ici j'ai un identificateur, qui est `n_carré`, un type qui est `int`, mais je n'ai pas de valeur initiale associée. Ces deux lignes constituent des déclarations de variables, valides en `c++`. Initialiser une variable consiste à lui donner une valeur de départ. Ceci se fait usuellement au moment de la déclaration. Ici, je déclare une variable `n`, de type entier, et je l'initialise au moyen de la valeur quatre, ce qui se traduit en mémoire par un schéma de cette nature. Je dispose désormais d'une zone identifiée par `n`, et qui contient une valeur initiale quatre. Ceci constitue la façon conseillée de procéder. Nous avons vu tout à l'heure qu'il était possible en `c++`, de déclarer une variable sans pour autant l'initialiser. Nous nous trouvons alors dans une situation de cette nature, où la variable `n_carré` peut contenir absolument n'importe quoi.

notes

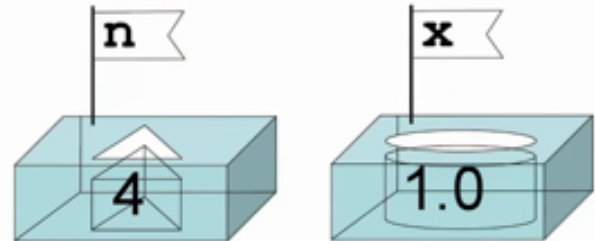
résumé

0m 49s



```
int n(4);  
double x(1.0);
```

decimal



Les traitements que l'on ferait avec cette variable `n_carré`, sont complètement imprédictibles, et peuvent produire n'importe quel type de résultats. Certainement des résultats erronés. C'est pour cela qu'il faut toujours initialiser une variable, avant d'utiliser la valeur qu'elle contient. Le type associé à la variable au moment de sa déclaration, est tout à fait fondamentale. C'est lui qui va conditionner le type de traitement que l'on peut réaliser avec la variable en question. Lorsque je fais une déclaration de cette nature dans un programme, j'indique que `n` ne peut contenir que des entiers, et rien d'autre. De même, si je déclare dans un programme une variable de type `double`, qui signifie décimale pour `c++`,

notes

résumé

1m 49s



Déclaration de variables

De façon générale, une déclaration de variable suit le schéma:

→ type identificateur(valeur_initiale);

ou éventuellement:

type identificateur;

Une fois défini, le type de la variable ne peut plus changer.

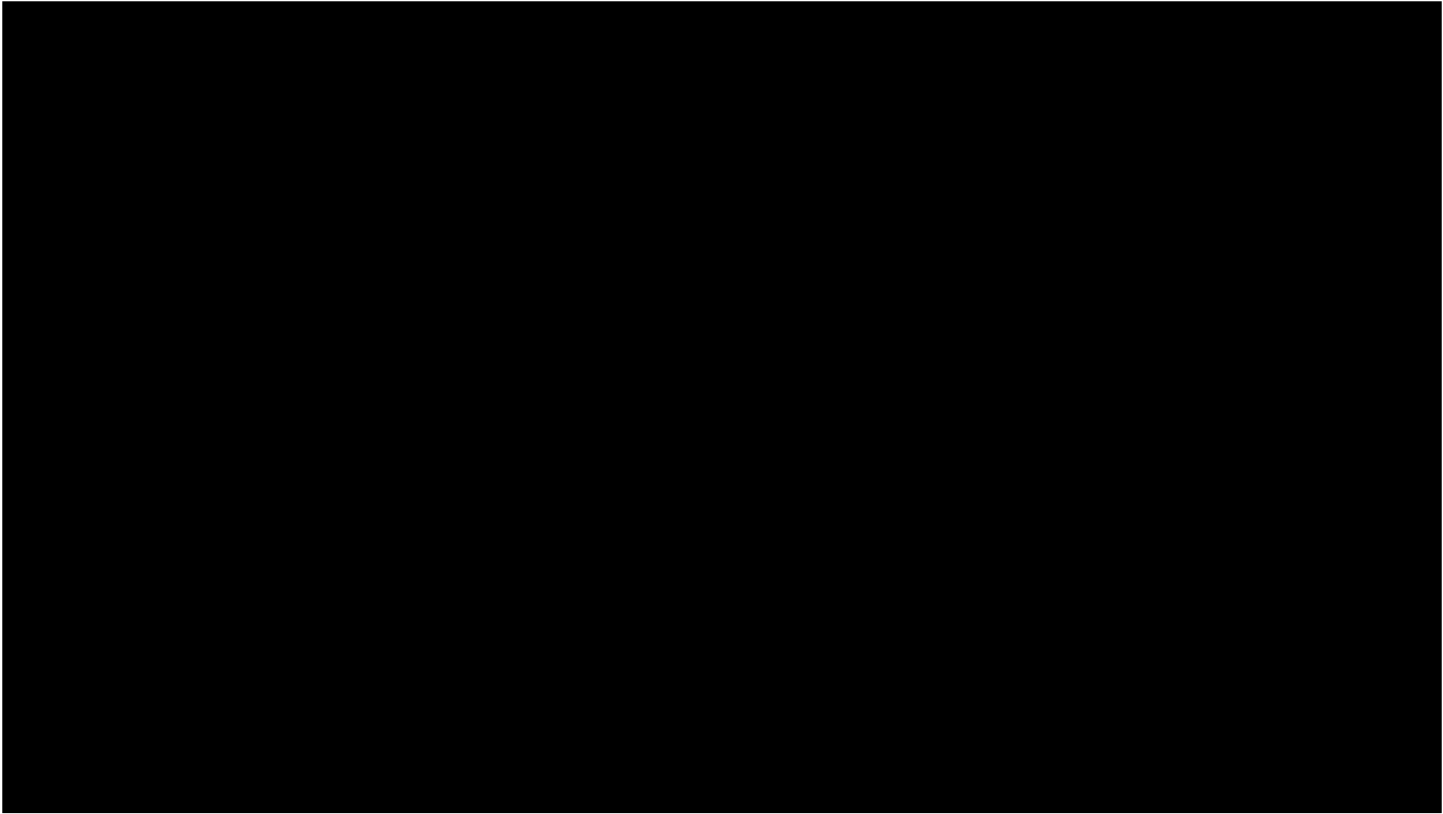
j'impose le fait que la variable x, ne peut contenir que des décimaux, et rien d'autre. Pour résumer, la déclaration d'une variable en c++ suit le schéma suivant. On indique le type de la variable, son identificateur, entre parenthèses, sa valeur initiale. Et ne pas oublier le point virgule à la fin.

notes

résumé

2m 37s





Cette deuxième façon de déclarer une variable est licite, mais déconseillée puisqu'on ne donne aucune valeur initiale à la variable, pour commencer. Il faut être attentif au fait que le type d'une variable ne peut pas changer. Lorsqu'on déclare une variable d'un certain type, cette variable va garder ce type jusqu'à la fin de son existence dans le programme. dans le programme.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

3m 1s

