



Support de cours

Cours:

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

Vidéo:

### Variables - lecture, écriture (partie 2)

Concepts (extraits des sous-titres générés automatiquement) :

**Entier n. Variable n. Instructions d'affichage. Début du programme. Expression de la variable n. T-d. Points cout. Dernier message. Variable n de type entier. Mémoire n. Affichage suivant. Lettre n. Carré de expression. Points endl. Ligne grâce.**



[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 1/6

# Variables : lecture/écriture

## (Partie 2)

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

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s



## Remarque

On peut écrire cout et endl simplement car le début du programme contenait la ligne:

```
using namespace std;
```

En l'absence de cette ligne, il faudrait écrire std::cout et std::endl.

Par exemple:

```
std::cout << "La variable n contient " << n << "." << std::endl;
```

Cout et endl s'appellent en fait std::cout (S-T-D deux points deux points cout) et std::endl (S-T-D deux points deux points endl). On peut les écrire simplement cout et endl parce qu'au début du programme, on a mis l'instruction : « using namespace std », ce qui permet d'éviter d'avoir à chaque fois à écrire std: devant les mots réservés qui sont dans ce qu'on appelle « le namespace de la bibliothèque standard, le namespace std ». Si on n'avait pas mis cette ligne using namespace std, alors il aurait fallu en toute rigueur écrire : std::cout et std::endl.

### notes

### résumé

0m 1s



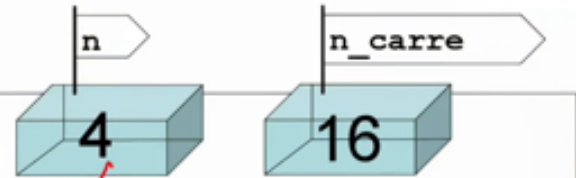
```
int n(4);
int n_carre;
```

```
n_carre = n * n;
```

```
cout << "La variable n contient " << n << "." << endl;
```

```
cout << "Le carre de " << n << " est " << n_carre << "." << endl;
```

```
cout << "Le double de n est " << 2 * n << "." << endl;
```



Ce qui s'affiche :

La variable n contient 4.

Le carre de 4 est 16.

Le double de n est 8.

Nous faisons cette remarque car parfois dans du code provenant de l'extérieur, vous pourriez rencontrer cette syntaxe. Reprenons maintenant pour illustrer tout ceci, le déroulement pas à pas de notre programme d'exemple. Au tout début, on commence par déclarer une variable  $n$  de type entier, que l'on initialise avec la valeur 4. Ceci a pour but de créer en mémoire un espace dans lequel on va pouvoir stocker un entier  $n$  et dans lequel on met la valeur 4. Ensuite, on va déclarer une variable, ici  $n^2$ , de type entier, mais comme elle n'est pas initialisée, on ne sait pas, a priori, quelle est la valeur qu'elle contient au départ. Puis, on exécute l'instruction  $n * n$  ( $n$  fois  $n$ ) va dans  $n^2$  c'est-à-dire,  $n^2 = n * n$  je vous rappelle que l'écriture du signe égal ici signifie que l'on évalue l'expression qui était à droite et qu'on la recopie dans la variable qui est ici, à gauche. Donc la variable  $n^2$  reçoit  $4 * 4$  c'est-à-dire 16. Puis nous passons aux instructions d'affichage, tout d'abord, on va afficher ici la variable  $n$  contient, puis on va afficher l'expression de la variable  $n$ , c'est-à-dire 4, sa valeur, puis on va ensuite afficher un point, et enfin, ici, on saute à la ligne grâce à ce endl. On passe à l'affichage suivant qui affiche le carré de l'expression recherchée en mémoire  $n$  qui vaut 4 est de nouveau une expression recherchée en mémoire, qui est  $n^2$ , qui vaut 16, puis saut à la ligne et enfin, on affiche le dernier message, « Le double de  $n$  est » Notez bien ici que  $n$  est une lettre comprise dans le message, elle va s'afficher telle quelle. C'est le mot, la lettre  $n$  qui va s'afficher ici, et non pas une expression comme on avait pu l'avoir sur la ligne précédente, ou comme on peut en avoir une ici. Donc le

notes

résumé

0m 49s



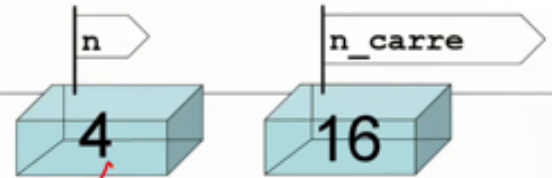
```
int n(4);
int n_carre;
```

```
n_carre = n * n;
```

```
cout << "La variable n contient " << n << "." << endl;
```

```
cout << "Le carre de " << n << " est " << n_carre << "." << endl;
```

```
cout << "Le double de n est " << 2 * n << "." << endl;
```



Ce qui s'affiche :

La variable n contient 4.

Le carre de 4 est 16.

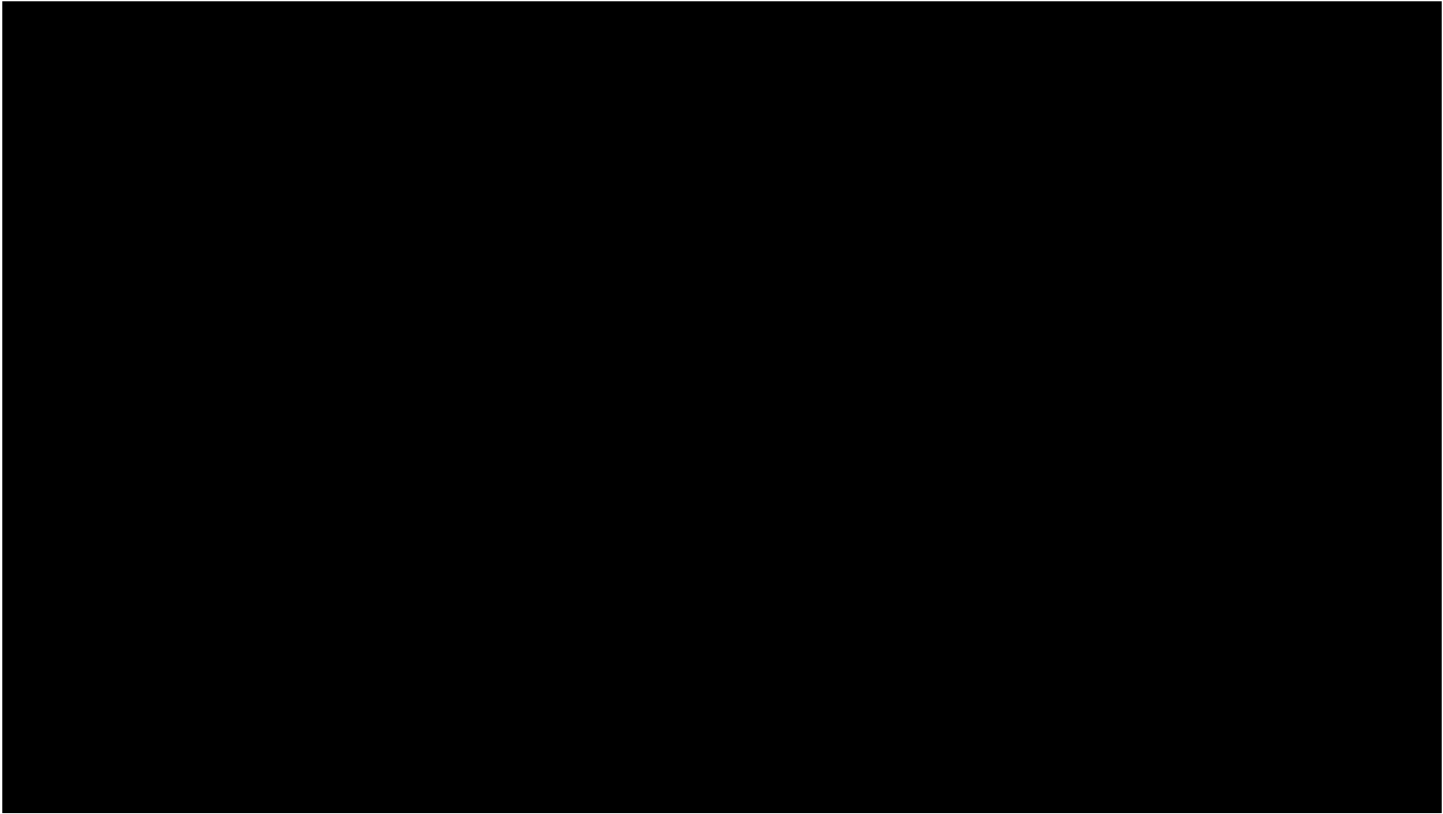
Le double de n est 8.

!

double de n est, puis affichage de l'expression 2 \* n

notes

résumé



donc on va chercher ici la valeur de  $n4$ ,  $2 * 4$  qui, bien sûr, va s'évaluer à 8, pour nous afficher ici 8. Enfin, affichage du point en tant que lettre, le message point et affichage du retour à la ligne. Une question, maintenant, regardez le code suivant et dites-nous ce qu'il affiche. et dites-nous ce qu'il affiche.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

3m 13s



.....

.....

.....

.....

.....