

Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

Blocs d'instructions (partie 5)

Concepts (extraits des sous-titres générés automatiquement) :

**Variable i. I principal. Fonction des règles de résolution. Résolution de la portée. Instruction
incrément. Bloc main. Boucle. Fin de la boucle. Dernier affichage. Valeur. Condition.
Portée. Exécution de l'incrément. Résolution. Foies.**



[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 5)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s



Exemple (à ne pas suivre !)

```
#include <iostream>
using namespace std;

int main()
{
    int i(120);

    for(int i(0); i < 5; ++i) {
        cout << i << endl;
    }

    cout << i << endl;

    return 0;
}
```

Déroulons donc le programme pas à pas. Tout d'abord nous déclarons ici

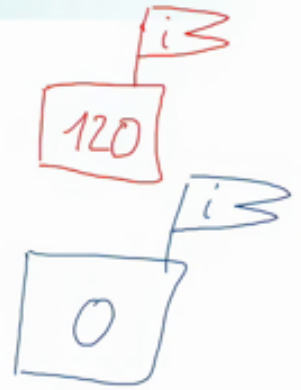
notes

résumé

0m 1s



Exemple (à ne pas suivre !)



```
#include <iostream>
using namespace std;

int main()
{
    → int i(120);
    → for(int i(0); i < 5; ++i) {
        cout << i << endl;
    }

    cout << i << endl;

    return 0;
}
```

donc une variable que l'on appelle `i` et que l'on initialise à la valeur cent vingt. Ensuite nous entrons dans la boucle, qui déclare ici une autre variable `i` qu'on initialise à la valeur zéro. On teste la condition. Est-ce que `i` est inférieur à cinq? De quel `i` s'agit-il ici? Il s'agit bien du `i` de la boucle, c'est-à-dire de ce `i` ici, donc cette valeur zéro est bien inférieure à cinq, donc effectivement on rentre dans la boucle et on affiche ici la valeur de `i`. Quel `i` une fois de plus?

notes

résumé

0m 5s

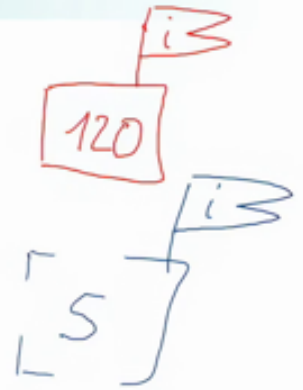


Exemple (à ne pas suivre !)

```
#include <iostream>
using namespace std;

int main()
{
    → int i(120);
    → for(int i(0); i < 5; ++i) {
        → cout << i << endl;
        cout << i << endl;
        return 0;
    }
```

0
1
2
3
4



Le `i` ici, résolution à la portée la plus proche, c'est le `i` de la boucle, c'est à dire ce `i`, ici et donc la ligne en question va nous afficher ici, la valeur zéro. Puis on arrive donc à l'exécution de l'incrément. A nouveau ici, de quel `i` s'agit-il? Il s'agit, résolution de la portée la plus proche du `i` de la boucle, de nouveau et donc c'est ce `i` ici, qui va passer à un, à la valeur un par le `++i` qui est là. Ensuite donc on re-teste la condition, donc on arrive ici pour re-tester la condition. Est-ce que ce `i`, est inférieur à cinq? Oui. Donc on refait l'affichage et on affiche donc, la valeur, cette fois-ci toujours du `i` de la boucle, mais qui vaut maintenant donc la valeur un, puis on incrémente toujours ce même `i`, qui va donc passer à la valeur deux, ce qui fait que la condition qu'on va tester va toujours être vraie et donc on va ici, de nouveau, afficher cette valeur deux. Et ainsi de suite, donc deux, trois, quatre. Qu'est-ce qu'il se passe quand maintenant ce `i` vaut quatre? Quand ce `i` vaut quatre on arrive à l'instruction incrémentation ici, qui le fait passer à cinq, donc on arrive à cinq. On teste la condition. La condition cette fois-ci est fausse puisque `i` ici, n'est pas strictement inférieur à cinq, cinq n'est pas strictement inférieur à cinq. La condition étant fausse, on termine la boucle et

notes

résumé

0m 49s

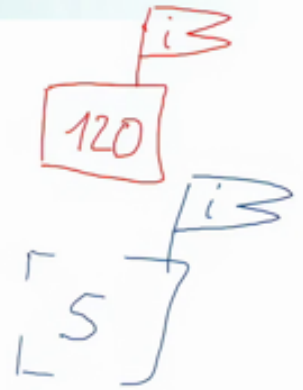


Exemple (à ne pas suivre !)

```
#include <iostream>
using namespace std;

int main()
{
    → int i(120);
    → for(int i(0); i < 5; ++i) {
        → cout << i << endl;
    }
    → cout << i << endl;
    return 0;
}
```

0
1
2
3
4



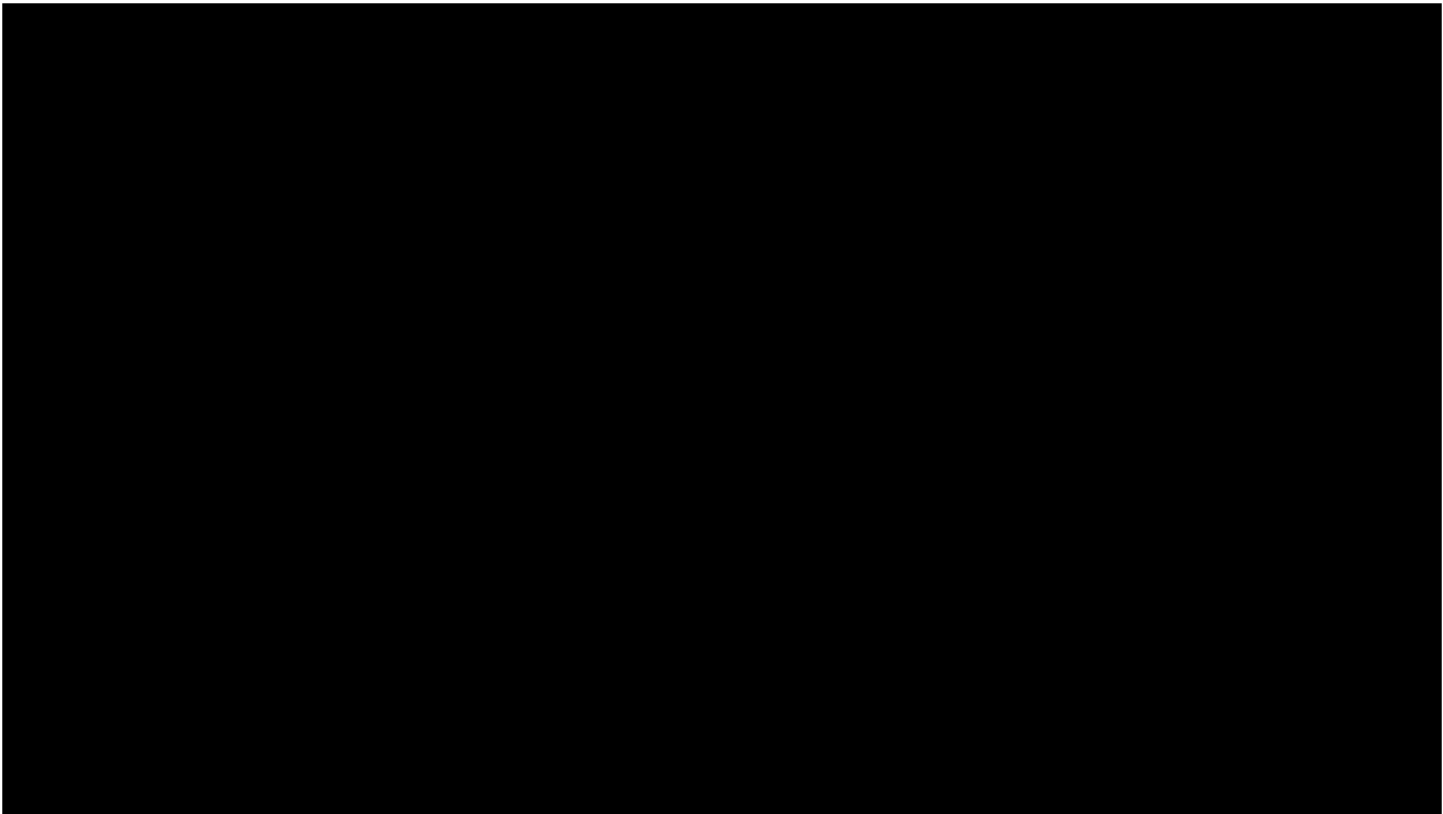
on arrive donc ici à la fin de la boucle. Ensuite donc on exécute l'instruction qui suit la boucle et on va afficher un i. La question c'est quel est ce i? En fonction des règles de résolution de portée, ce i c'est le i principal, le i du niveau dans lequel on est ici, puisque je vous rappelle que le i de la boucle

notes

résumé

2m 25s





ici, a une portée qui est réduite à la boucle. Ce i ici a une portée qui est réduite à la boucle, il n'existe plus à partir d'ici. Et donc le i dont on parle ici, dans le bloc main, c'est bien le i de départ ce qui fait que donc, le dernier affichage va être cent vingt. être cent vingt.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

2m 49s



.....

.....

.....

.....

.....