

Support de cours

Cours:

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

Vidéo:

### Conditions (partie 4)

Concepts (extraits des sous-titres générés automatiquement) :

**Évaluation de l'expression. Cas d'un petit programme. Moyen de l'opérateur logique. Petit exemple concret. Premier opérande. Expression globale. Dernier cas de figure. Cas présent. Évaluation de cette première expression. Premier temps. Évaluation de la seconde expression. Variables m. Évaluation de l'expression globale. Exemples d'utilisation de l'opérateur. Opérateur logique.**



[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/>

# Conditions

(Partie 4)

## Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s

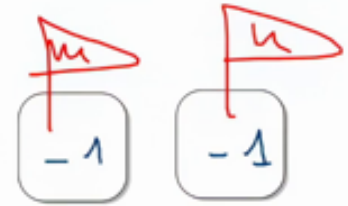


# Exemple avec l'opérateur logique `or`

```
cout << "Entrez deux valeurs:" << endl;
cin >> m >> n;
```

```
if ( (m >= 0) or (n >= 0) ) {
    cout << "au moins une valeur est positive" << endl;
} else {
    cout << "les deux valeurs sont negatives" << endl;
}
```

*Handwritten annotations: "false" above (m >= 0) and (n >= 0), and "true" above the 'or' operator.*



Illustrons maintenant l'utilisation du OU logique sur un petit exemple concret. Ici nous avons le cas d'un petit programme qui demande à l'utilisateur de choisir deux valeurs. Ces deux valeurs sont saisies depuis l'entrée standard et stockées dans deux variables `m` et `n`. Nous souhaitons dans le cas présent que l'une ou l'autre au moins soit positive, puisqu'il nous suffit que l'une ou l'autre des deux valeurs soit positives. Nous testons ces conditions au moyen de l'opérateur logique OU. Supposons dans un premier temps que l'utilisateur introduise les valeurs 1 et -1. Dans ce cas, le premier opérande sera évalué à "true" puisqu'il est effectivement positif, et donc l'évaluation de l'expression retourne "true" puisqu'il suffit que l'un ou l'autre des opérandes soient "true" pour que le résultat soit "true" pour l'expression globale. Et donc c'est bel et bien le message "au moins une valeur est positive" qui sera affiché dans le cas présent. Supposons maintenant que les valeurs saisies soient -1 et 1. Dans ce cas, l'évaluation de cette première expression retourne "false", l'évaluation de la seconde expression retourne quant à elle "true", puisqu'il suffit que l'un ou l'autre des opérandes soit "true" pour que l'ensemble soit "true", l'évaluation de l'expression globale retourne "true", et là encore c'est le message "au moins une valeur est positive" qui sera affiché. Dernier cas de figure: imaginons maintenant que l'utilisateur saisisse les valeurs moins 1 et -1 pour chacune des variables `m` et `n`, dans ce cas-ci l'évaluation de la première expression va retourner "false", l'évaluation de la seconde expression retourne également "false", puisque les deux opérandes

## notes

## résumé

0m 1s



# Les opérateurs logiques

L'opérateur logique `not` (NON):

par exemple, la condition

not (a < b)  
false

est vraie si (a < b) est fausse, et fausse si (a < b) est vraie.

L'opérateur `not` peut aussi s'écrire `!` : On aurait pu écrire

! (a < b)

Nous verrons des exemples d'utilisation de cet opérateur plus loin dans la suite du cours.

sont évalués à "false", l'expression globale retourne également "false" et dans ce cas c'est le message "les deux valeurs sont négatives" qui sera affiché. Dernier opérateur logique usuel, le NON, la négation, qui en C++ s'écrit soit au moyen du mot réservé `not`, soit au moyen du point d'exclamation. L'opérateur logique `not` est un opérateur logique unaire, c'est-à-dire qu'il n'attend qu'un seul opérande, qui à nouveau est une expression logique retournant "true" ou "false". La vocation de l'opérateur logique `not` est de retourner la négation de la valeur de son opérande. Par exemple, supposons que l'opérande soit évalué à "false" ce qui signifie que a est supérieur ou égal à b, l'évaluation de l'expression globale avec le

notes

résumé

1m 49s





not retournera avec la négation de "false", c'est-à-dire "true". Supposons maintenant que l'évaluation de cet opérande retourne "true", l'expression globale retournera cette fois la négation de "true", c'est-à-dire "false". Nous aurons l'occasion de voir plusieurs exemples d'utilisation de l'opérateur not dans la suite du cours. de l'opérateur not dans la suite du cours.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

