



Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

Expressions (partie 2)

Concepts (extraits des sous-titres générés automatiquement) :

Cas des int. Exemple. Variable i. Cas des boucles for. Opérateurs d'incrément. Reste de la division entière. Opérateurs. Droit. Résultat du modulo. Valeur de la variable a. Leçon suivante. Opérateur. Modulo. Caractère. Décrément.



[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 1/7

Expressions et opérateurs

(Partie 2)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s



On dispose aussi des opérateurs +=, -=, *=, /=

En C++, on dispose également des opérateurs += (plus égal) -= (moins égal) *= (multiplié égal) et /= (divisé égal).

notes

résumé

0m 1s



On dispose aussi des opérateurs +=, -=, *=, /=

$a += 5;$

$a = a + 5;$

$a *= b;$

$a = a$

Par exemple, j'ai le droit d'écrire $a += 5$ et c'est équivalent à écrire $a = a + 5$ Un autre exemple, je peux écrire $a *= b$

notes

résumé

0m 13s



Opérateur modulo (%)

Dans le cas des `int`, il existe aussi:

- un opérateur modulo, noté `%`, qui renvoie le reste de la division entière:

$$\begin{array}{lcl}
 11 \% 4 & \text{vaut} & 3 \\
 72 \% 4 & \text{vaut} & 0
 \end{array}
 \qquad
 11 = 2 \times 4 + \underline{3}$$

et c'est équivalent à écrire $a = a * b$ Dans le cas des `int` et dans le cas des `int` seulement, on dispose également d'un opérateur qui s'appelle modulo et qui se note avec le caractère `%` (pourcent) et qui renvoie le reste de la division entière. Par exemple, 11 modulo pour cent 4 vaut 3 Pourquoi ? Parce que 11 est égal à $2 * 4 + 3$ et c'est ce reste qui correspond au résultat du modulo. Un autre exemple, 12 modulo 4... vaut, à votre avis... ?

notes

résumé

0m 37s



Opérateurs ++ et --

Il existe aussi:

- deux opérateurs notés ++ et --, qui permettent respectivement d'incrémenter et de décrémenter, c'est-à-dire d'ajouter et de soustraire 1 à une variable.

Par exemple, l'instruction:

```
++i;
```

est équivalente à :

```
i = i + 1;
```

Ces deux opérateurs sont souvent utilisés avec l'instruction **for**, que nous verrons par la suite.

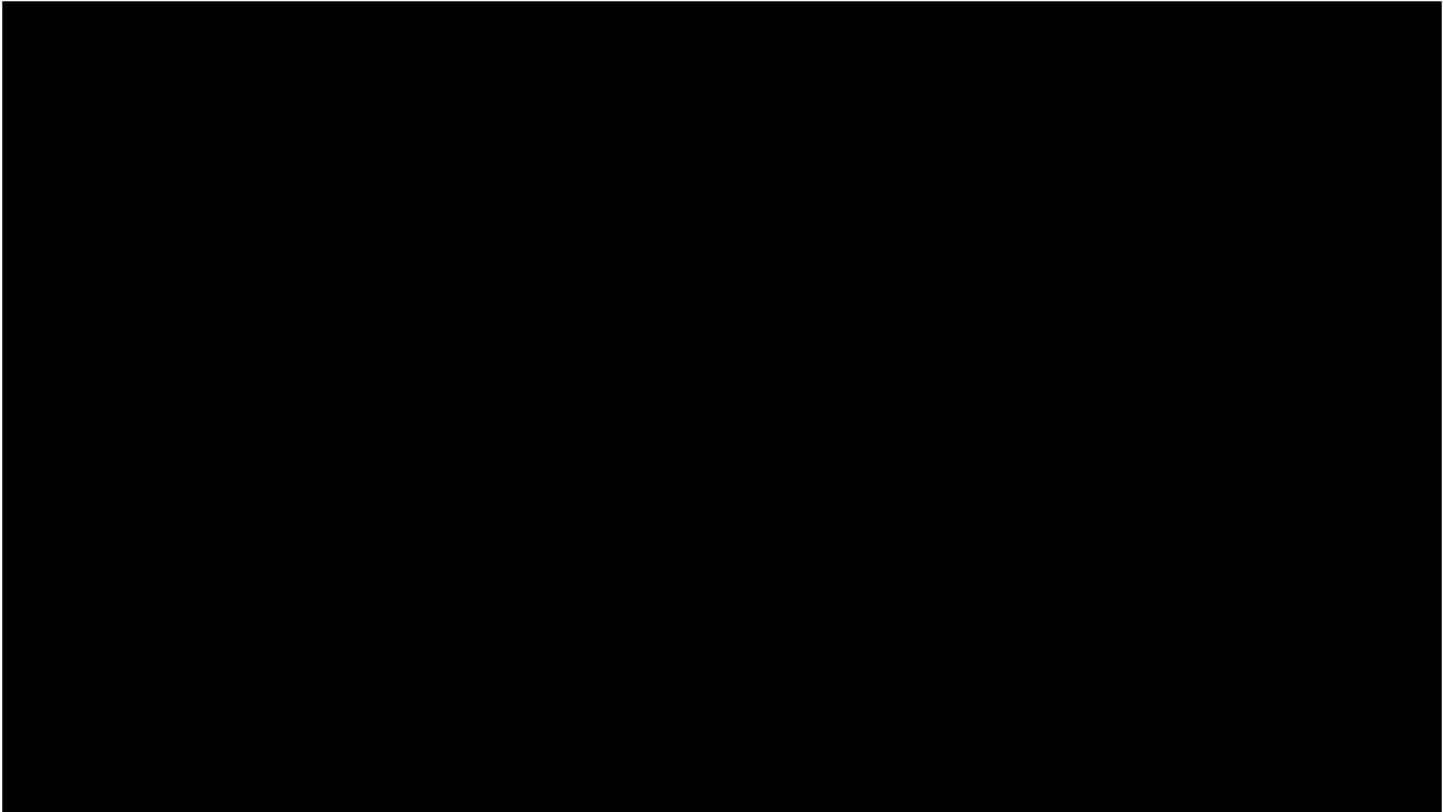
Alors pourquoi ? Parce que $12 \text{ est égal à } 3 * 4 + 0$ et c'est ce 0 qui est le résultat du modulo. Il existe aussi des opérateurs qui se notent ++ (plus plus) et -- (moins moins) qui s'appellent des opérateurs d'incrément et de décrément, c'est-à-dire qu'ils vont permettre d'ajouter 1 ou de soustraire 1 à des variables. Par exemple, j'ai le droit d'écrire ++i et c'est équivalent à écrire $i = i + 1$ qui va donc ajouter 1 à la variable i.

notes

résumé

1m 25s





J'aurais pu aussi, par exemple, écrire `--a`, qui sera équivalent à `a = a - 1` ce qui va retirer 1 à la valeur de la variable `a`. Ces opérateurs sont très utilisés dans le cas des boucles `for`, que nous verrons dans une leçon suivante. que nous verrons dans une leçon suivante.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

2m 1s

