2023

Theo Tessier

Le plus beau

17/08/2023

Note C++ Maison



Table des matières

[**Opérateurs** 2](#_Toc143269496)

[**Arithmétiques** : 2](#_Toc143269497)

[**Booléens** : 2](#_Toc143269498)

[**Comparaison** : 2](#_Toc143269499)

[**Affectation** : 2](#_Toc143269500)

[**Préfixes** : 3](#_Toc143269501)

[**Bonus** : 3](#_Toc143269502)

[**Variables** 3](#_Toc143269503)

[**Types fondamentaux** : 3](#_Toc143269504)

[**Int** 3](#_Toc143269505)

[**Float** 3](#_Toc143269506)

[**Double** 3](#_Toc143269507)

[**Bool** 3](#_Toc143269508)

# **Opérateurs**

## **Arithmétiques** :

+ (addition),

- (soustraction),

\* (multiplication),

/ (division),

% (modulo/le reste d’une division).

## **Booléens** :

true (1),

false (0),

! (négation (exemple : !true = 0 = false, !false = 1 = true)),

&& (et/avec (si deux conditions remplies, alors && = 1) (exemple : true&&true = 1, true&&false = 0 )),

|| (ou (une des deux conditions est rempli) (exemple : true||true, true||false (si true||false alors c’est true)).

## **Comparaison** :

< (inférieur),

> (supérieur),

<= (inférieur ou égal),

>= (supérieur ou égal),

== (égal),

!= (différent).

## **Affectation** :

+= (affecter une valeur (exemple : int a = 1 ; a +=2 ; alors a =3)),

-= (enlever une valeur (exemple : int a = 1 ; a -=1 ; alors a = 0)),

\*= (multiplier par une valeur (exemple : int a = 1 ; a \*= 12 ; alors a = 12)),

/= (diviser par une valeur (exemple : int a = 6 ; a /= 2 ; alors a = 3)),

%= (affecter le reste).

## **Préfixes** :

Lorsque l’on cherche à afficher un préfixe (binaire, octal, hexa) celui-ci sera afficher en décimal dans le terminal (cela fait une traduction automatique).

0b = binaire (0/1) (exemple : int a = 0b10111011 ; a décimale = 187) (mettre 8 chiffres car 8 octets),

0 = octal (8) (exemple : int a = 0256 ; a décimale = 174),

0x = hexadécimale (16) (exemple : int a = 0x15bf ; a décimale = 5567),

Sans rien = décimale (exemple : int a = 12 ; a décimale = 12).

## **Bonus** :

Écrire a += 1 peut aussi s’écrire ++a (++a ne fonctionne que pour +1 (++a ⬄ 0 + 1 (si a = 0))),

Idem pour a -= 1 cela peut s’écrire --a(ne fonctionne qu’avec 1 (--a ⬄ 0 – 1 (si a = 0))).

# **Variables**

## **Types fondamentaux** :

**Int** : Int est un entier (de 32 bits) et permet donc de créer des entiers (exemple : int a = 0 ;) par conséquent int ne peut pas avoir de chiffre à virgule (car non entier) alors si int a = 1 ; et que a /= 12 ; alors le résultat ne sera pas a = 1/12 mais a = 0, car 1/12 = 0.0833… étant alors un chiffre à virgule. Int prend 4 octets et est un 32 bits. (Valeur : min : - 2 147 483 648, max : 2 147 483 647).

**Float** : Float est un flottant (de 32 bits) permettant de créer des flottant/chiffres a virgule. Cela s’écrit de la manière suivante : float a = 1.5f ; (f signifiant flottant) et donc en faisant : float a = 1.0f ; a /= 12, le résultat sera a = 0.833333. Remarque : si on écrit a = 1f alors le programme n e fonctionnera pas car 1 est un entier mais 1.0 est un flottant. Cependant float n’a une précision que de 7 chiffres après la virgule (car 4 octets). (Valeur : min : +- 1.17554e-38, max = +- 3.40282e+38).

**Double** : Double peut-être entier comme flottant, celui-ci a les mêmes caractéristiques que le int et le float. Mais contrairement au float, double à une précision de 15 chiffres après la virgule (car 8 octets (4x2), utiliser un double entant que int est possible mais cela reviendrai à gâcher de la mémoire (8 octets > 4 octets). De double est un 64 bits. (Valeur : min : +-2.22507e-308, max : +- 1.79769e+308).

**Bool**: Bool fonctionne de façon binaire, en false et true (0 et 1). Si un bool est utilisé il ne peut répondre que de deux manières. Il n’est donc utilisé que pour « valider ou non » une condition. Boll ne prend que 1 octet et est un 8 bits.

**Char** : Char permet d’afficher des caractères, que ce soit un emoji ou une lettre avec accent (😊/Â). Char ne prend que 1 octet et est un 8 bits. (Valeur : min : -128, max : 127).

**Auto** : Auto sert à laisser le compilateur déduire le type d’une variable via ce qui lui est assigné (auto a = 5, alors auto = int. auto b = 5.023f, alors auto = float). Idem pour du texte si auto a = « wesh » alors auto = string (string fait parti de la catégorie standard de la bibliothèque string). Donc auto est un peu comme un métamorphe, et sa place dépend du type qu’il prend.

## **Catégorie des standards :**

### **Cout** :