

# Crash course de C++

Pour programmeurs en n'importe quel langage raisonnable

beOI Training



OLYMPIADE BELGE D'INFORMATIQUE  
BELGISCHE INFORMATICA-OLYMPIADE

1<sup>er</sup> mars 2017

# Table des matières

Les bases

# Hello world !

```
// Ceci est un commentaire.  
#include <bits/stdc++.h> // importe toute la STL  
using namespace std; // évite de taper std::  
  
// Déclaration de fonction : type nom() { [...] }  
int main() {  
    // Toutes les instructions terminent par ";"  
    cout << "Hello World!" << endl;  
    // endl = retour à la ligne  
}
```

- ▶ La fonction **int** main() est appelée au lancement.
- ▶ Le **int** veut dire que main() retourne un entier, mais le **return** est implicite.
- ▶ Le sens des chevrons << indique que les mots vont vers cout, la "sortie standard".

# Variables et opérations

Une variable doit avoir un type spécifié

```
int i = 5; // entiers dans  $[-2^{31}, 2^{31}[$ 
double j = 5.4; // nombres à virgules
bool b = true; // booléen (vrai ou faux)
char ch = 'D'; // caractères seuls
string s = "abcd"; // chaîne de caractères

// On peut les initialiser plus tard
int k, l; // plusieurs variables du même type

k = i + 2;
l = 7 / 3; // division entière  $\Rightarrow l = 2$ 
s += ch; // ajout d'un caractère
j /= 3; // divise j par 3
i++; l--; // ajoute 1, enlève 1
```

# Conditions

```
int age;
cout << "Quel est votre âge ? ";
cin >> age; // lecture d'input
if (age < 18)
    cout << "Vous êtes mineur." << endl;
else if (age <= 120)
    cout << "Vous êtes majeur." << endl;
else {
    int a=3, b=4, c=5;
    bool rectangle = (a*a + b*b == c*c);
    if (rectangle && !(a == 0 || b == 0))
        cout << "Hypoténuse = " << c << endl;
}
```

- ▶ Les accolades {} sont facultatives pour une seule ligne.
- ▶ Le sens des chevrons >> indique que l'entier vient de cin, "l'entrée standard".

# Boucles

```
// Imprime les nombres de 1 à 5
for (int i=1; i<=5; i++)
    cout << i << endl;

string s; // initialement vide
while (s != "oui") { // pas égal
    cout << "Aimez-vous programmer ? ";
    cin >> s;
}
```

- ▶ Les boucles **for** (;;) ont trois parties :
  - ▶ Initialisation : initialise une ou plusieurs variables
  - ▶ Condition : la boucle s'arrête quand elle est fausse
  - ▶ Incrémentation : exécutée à la fin de chaque itération

# Fonctions

```
// Type obligatoire pour résultat et paramètres
int square(int x) {
    return x*x;
}
void sayHello(string s) { // void = ne renvoie rien
    cout << "Bonjour " << s << endl;
}
int main() {
    int y = square(4); // y = 16
    sayHello("Victor");
}
```

- ▶ Pas imbriquables, et toujours placées *avant* leur appel. Sinon il faut les déclarer avec **void** sayHello(string s); et les implémenter après.
- ▶ Quand une fonction n'est pas **void**, toutes les exécutions possibles doivent terminer avec un **return**.

# Tableaux

Toutes les cases d'un tableau doivent avoir le même type.

```
int maxi(int tab[], int n) { // le [] est toujours
    int ma = 0;              // après le nom
    for (int i=0; i<n; i++)
        ma = max(ma, tab[i]);
    return ma;
}

int main() {
    int a[5], b[4][3]; // 4 lignes et 3 colonnes
    for (int i=0; i<5; i++)
        cin >> a[i];
    cout << maxi(a, 5) << endl;
}
```

- ▶ Le premier élément est à l'indice [0].
- ▶ Un tableau ne connaît pas sa taille ! Il faut la donner à part quand on l'envoie à une fonction.