

Cours 3 - Fonctions

Notions : définition, paramètre avec valeur par défaut, passage d'argument par valeur, argument pointeur, passage d'argument par référence, documentation de fonctions.

Définitions de fonctions

Pour définir une fonction on écrit son type de retour, son nom, ses arguments précédés de leur type entre parenthèses séparés par des virgules puis son code entre accolades. Un argument peut avoir une valeur par défaut. Les arguments sont passés à la fonction par copie ou par référence. Plusieurs fonctions peuvent avoir le même nom si les arguments sont différents.

```
type_de_retour nom(arg1, arg2, ...) {  
    instructions  
}
```

Voici quelques définitions de fonctions.

Fonction sans argument et sans valeur de retour

1.

```
void f() {  
    int x = 1;  
}
```

2.

```
#include <iostream>  
  
void f() {  
    std::cout << "f" << std::endl;  
}
```

Fonction avec deux paramètres de types int qui retourne un int.

```
int f(int x1, int x2) {  
    return x1 + x2;  
}
```

Fonction avec un paramètre avec une valeur par défaut.

```
int f(int x = 0) {  
    return x + 1;  
}  
  
x = f(1)  
x = f()
```

Passage d'argument par valeur

L'argument utilisé dans la fonction est une copie. la variable en dehors de la fonction n'est pas modifiée.

prog.cpp

```
#include <iostream>  
  
void f(int x) {  
    x = 1;  
}  
  
int main() {  
    a = 2  
    f(a)  
    std::cout << a << endl;    // --> 2  
}
```

Arguments de type pointeur

Si l'argument est un pointeur, on ne peut toujours pas modifier sa valeur mais on peut modifier la valeur sur laquelle il pointe.

prog.cpp

```
#include <iostream>  
  
void f(int* x) {  
    *x = 1;  
}  
  
int main() {  
    int a = 2;  
    f(&a);  
    std::cout << a << std::endl;    // --> 1  
}
```

Passage d'argument par référence

L'argument utilisé dans la fonction est une référence vers une variable.

prog.cpp

```
#include <iostream>

void f(int& x) {
    x = 1;
}

int main() {
    a = 2
    f(a)
    std::cout << a << endl;    // --> 1
}
```

Documentation de fonctions

La documentation d'une fonction est un commentaire placé avant la définition de la fonction. Elle contient sa description, la description des arguments et de sa valeur de retour.

```
/*
    Incrémente x.

    param: x Un nombre entier à incrémenter.
    return: Le nombre entier incrémenté.
*/
int g(int x) {
    return x + 1;
}
```

Programme avec fonctions

prog.cpp

```
#include <iostream>

int g(int x) {
    return x + 1;
}

void print(int x) {
    std::cout << "-> " << g(x) << endl;
}

int main() {
    print(1);
    return 0;
}
```

Déclaration de fonctions

Pour utiliser une fonction dont la définition n'a pas encore été lue par le compilateur, il faut d'abord déclarer la fonction en écrivant son type de retour, son nom et ses arguments sans le code entre accolades.

prog.cpp

```
// Déclaration de f
void f(int x);

int main() {
    // Utilisation de f
    f(1);
    return 0;
}

// Définition de f
void f(int x) {
    int y = x + 1;
}
```