

# Fonctions

---

## Question 1 :

Une fonction peut avoir \_\_\_\_\_ paramètres.

*zéros ou plus*

## Question 2 :

Les variables définies dans le corps d'une fonction qui sont visibles uniquement par la fonction sont appelées variables \_\_\_\_\_.

*locales*

## Question 3 :

Les variables définies dans les fonctions dont les valeurs persistent d'un appel à l'autre sont appelées variables \_\_\_\_\_.

*statiques*

## Question 4 :

La manière par défaut avec laquelle les paramètres sont transmis aux fonctions est obtenue en C++ \_\_\_\_\_.

*passage par valeur*

## Question 5 :

Qu'est-ce qui est affiché par le code suivant ?

```
#include <iostream>
using namespace std;

void func(int x, int y, int z) {
    x = y + z;
    y = 10;
    x = 20;
}

int main() {
    int a = 10, b = 20, c = 30;
    func(a, b, c);
    cout << a << " " << b << " " << c << endl;
    return 0;
}
```

*10 20 30*

**Question 6 :**

Qu'est-ce qui est affiché par le code suivant ?

```
#include <iostream>
using namespace std;

void func(int &x, int &y, int &z) {
    x = y + z;
    y = 10;
    x = 20;
}

int main() {
    int a = 10, b = 20, c = 30;
    func(a, b, c);
    cout << a << " " << b << " " << c << endl;
    return 0;
}
```

20 10 30

**Question 7 :**

Les arguments \_\_\_\_\_ peuvent être automatiquement fournis à une fonction lorsqu'aucun argument n'est fourni lors de l'appel de la fonction.

*défaut*

**Question 8 :**

Lorsqu'une fonction s'appelle, directement ou indirectement, cela est défini comme \_\_\_\_\_.

*récurtivité*

**Question 9 :**

Avant de pouvoir appeler une fonction en C++, elle doit être définie ou avoir un \_\_\_\_\_ fourni.

*prototype*

**Question 10 :**

La création de plusieurs versions du même nom de fonction qui accepte différents paramètres est appelée

*surcharge de fonctions*