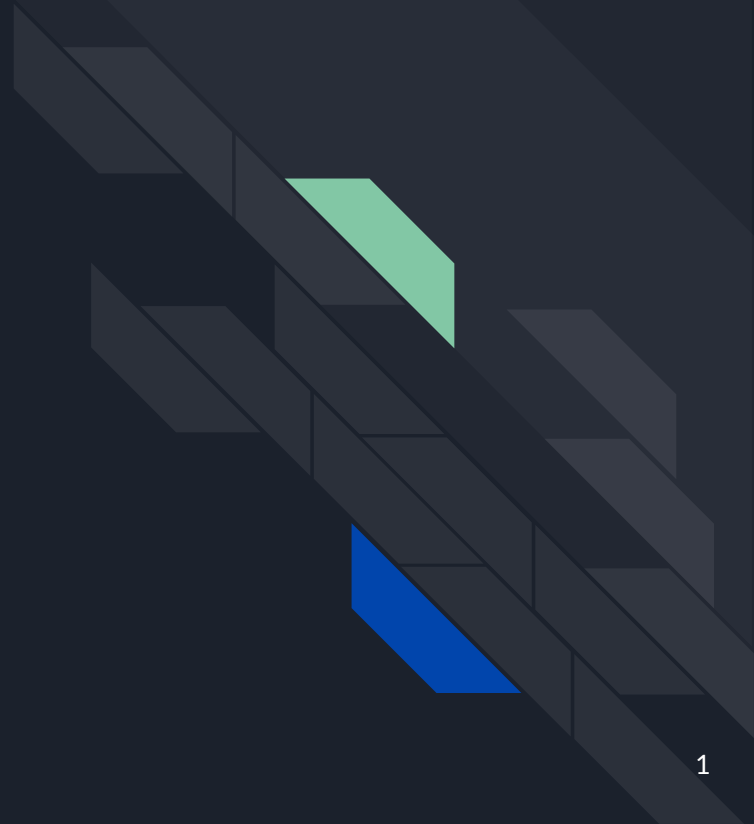


Fonctions





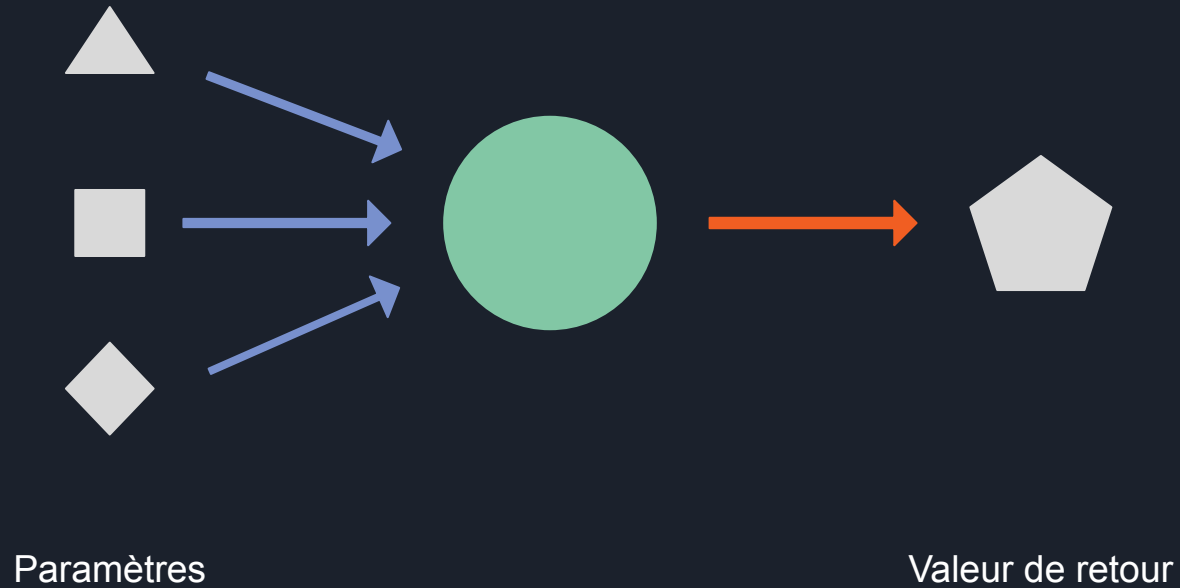
Fonction

Une **fonction** est un bloc d'instructions séparé portant un nom, dit **identifiant**.

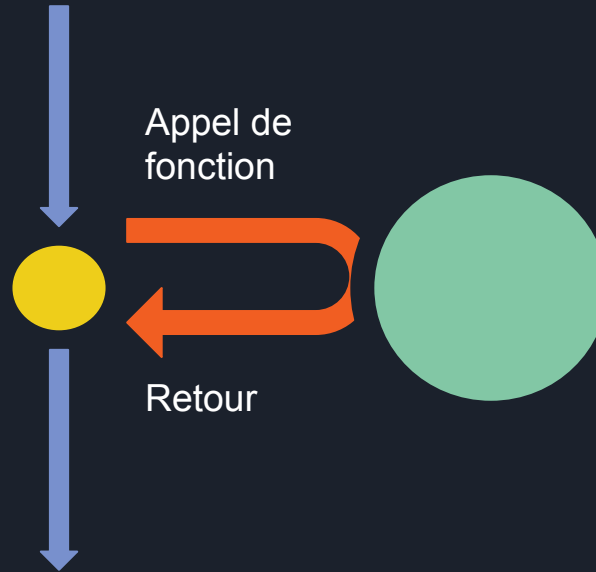
La fonction prend en entrée des **paramètres**.

La fonction renvoie à l'appelant une **valeur de retour**.

Entrée - Sortie



Retour à l'appelant





Définition

Définition d'une fonction :

```
int Add(int a, int b) {  
    int c = a + b;  
    printf("Added: %d + %d = %d", a, b, c);  
    return c;  
}
```



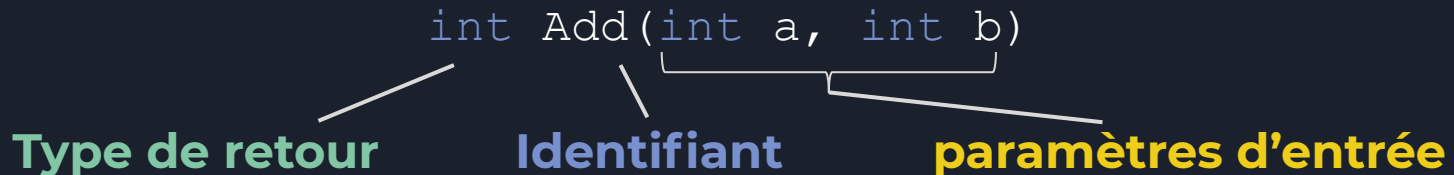
Signature

La première ligne de la fonction contient son identifiant et sa **signature**. Elle comporte :

- Le **type de retour**
- L'**identifiant**
- Les **paramètres d'entrée**.

`int Add(int a, int b)`

Type de retour **Identifiant** **paramètres d'entrée**





Paramètre

Un **paramètre** est une **valeur** envoyée en **entrée** de fonction.

Cette valeur est **copiée** dans une nouvelle variable.

```
int Add(int a, int b)
```

paramètres d'entrée



Retour

L'instruction **return** marque la **fin de l'exécution** de la fonction et précise la **valeur de retour**.

```
int Add(int a, int b) {  
    int c = a + b;  
    printf("Added: %d + %d = %d", a, b, c);  
    return c;  
}
```

Ici la valeur de retour est la valeur de la variable c.



Appel

La fonction peut être **appelée** par son identifiant n'importe où dans le code **après sa déclaration**.

Une fois la fonction exécutée, la **valeur de retour remplace l'appel** de fonction dans l'appelant.

```
int add(int a, int b) {  
    return a + b;  
}  
  
int main() {  
    int result = add(1, 2)*2; // add(1, 2) remplacé par la valeur de retour : 3.  
    printf("Result: %d", result); // Affiche "Result: 6"  
    return 0;  
}
```



Prototype

La fonction peut être définie après une autre fonction l'appelant. Dans ce cas, le **prototype** de la fonction doit être **déclaré avant** cet appel.

```
int add(int a, int b); // déclaration du prototype de la fonction

int main() {
    int result = add(1, 2)*2; // add(1, 2) remplacé par la valeur de retour : 3.
    printf("Result: %d", result); // Affiche "Result: 6"
    return 0;
}

int add(int a, int b) {
    return a + b;
}
```



Retour void

Une fonction peut **ne pas retourner de valeur** en utilisant le type de **retour void**.

```
// Définition d'une fonction qui affiche le résultat d'une addition.
void printAddition(int a, int b) {
    int c = a + b;
    printf("Added: %d + %d = %d", a, b, c);
    return; // ne renvoie pas de valeur
}

int main() {
    printAddition(1, 2); // La fonction est appelée mais
                        // aucune valeur de retour n'est récupérée.
    return 0;
}
```



Main

La fonction **main** est le point d'entrée d'un programme C++. Elle est généralement appelée par le système d'exploitation.

Sa valeur de retour indique si la fonction s'est correctement terminée ou non.

```
int main() { // main est une fonction !  
    return 0;  
}
```



Fonction sur variables (C++)

Certains types de variables (i.e. objets) **contiennent des** définitions de **fonctions** qui leurs sont propres.

Ces fonctions sont **appelées depuis une variable** de ce type.

```
Square mySquare;  
mySquare.SetSize(10.0f);
```

La **variable** appelant peut être **manipulée** par la fonction.