

# INF155 - Fonctions

Hugo Leblanc

Cours 3

## Fonctions

### Création de fonctions

- Une fonction doit faire une seule et unique tâche. Sinon, elle devrait être séparée en deux fonctions.
- Une fonction contient l'identificateur (son nom), des paramètres (les entrées), et un retour (la sortie).
- Les paramètres et le retour sont facultatifs. L'absence d'élément est représentée par le type void.

### Définition

Les fonctions et leurs instructions doivent être définies dans notre code. Elle se présente comme la fonction main. La différence est qu'elles devront être appelées pour être utilisées.

### Exemple d'une définition fonction

```
int ma_fonction(int para1, int para2){  
    int x;  
    int y;  
    x = para1 + para2;  
    y = x * para1;  
  
    return x - y;  
}
```

## Déclaration

Le compilateur regarde de manière séquentielle les fichiers pour l'utilisation de fonctions. Si la fonction est définie plus bas que son utilisation. Le compilateur renvoie une erreur. Nous utiliserons la déclaration de fonctions pour présenter les fonctions au compilateur avant leur définition.

## Exemple de déclaration avec définition

```
// Les déclarations se font avant les définitions.
double fcn1(double a, double b);

// Définition du main
int main(void) {return 0;}

// Les définitions suivent la fonction main
double fcn1(double a, double b){
    return a * b;
}
```

## Paramètres de fonction

### Liste des paramètres

- Les paramètres constituent les informations primordiales à l'utilisation de la fonction.
- Chaque paramètre inclut son type et identificateur.
- Plusieurs paramètres sont délimités par des virgules.

## Retour de fonction

L'instruction return permet de sélectionner la valeur à retourner par la fonction. Dès que la fonction rencontre une instruction return, la fonction se termine et la valeur de retour est renvoyée.

## Appels de fonctions

### Les appels

- On utilise le nom de la fonction pour en faire son appel.

- L'appel de la fonction sera remplacé par sa valeur de retour dans l'expression où la fonction est appelée.

## Les librairies

- Une multitude de fonctions sont disponibles dans les librairies standards du langage C

Librairie	Utilisation	Fonction
assert.h	Test de condition	assert
math.h	Calcul mathématique	abs, exp, pow, sqrt, ...
stdbool.h	Utilisation du type bool	type bool, true, false
string.h	Chaine de caractères	strcat, strcmp, strlen, ...

## Durée de vie

- Tout ce qui se passe à l'intérieur des fonctions est détruit après l'appel de la fonction.
- Toute déclaration de variables à l'intérieur d'une fonction est détruite après l'appel de la fonction.
- Seul le retour est renvoyé.

## Passage par valeur

- Les paramètres et les retours sont renommés pour la durée de la fonction.
- Seules leurs valeurs seront transférées entre la fonction et l'appelant.
- Les noms des paramètres et des retours n'ont aucune incidence.
- L'ordre des paramètres et des retours est ce qui sera considéré.

## Variables globales

- Une variable globale permet de lier une variable de l'espace de travail normal à l'espace de travail d'une fonction.
- La variable doit être indiquée comme globale en étant initialisée à l'extérieur des blocs de code de fonctions.

- Sauf avis contraire, l'utilisation de variable globale est interdite dans le cours.