

# Programmation Langage C

## Types de base

### Lecture/écriture

Youssef ALJ

24 février 2020

## 1 Variables et types de bases - entrée/sortie

- Variables et types de base
- Lecture et écriture
  - la fonction `printf`
  - la fonction `scanf`

- Quand on programme avec n'importe quel langage de programmation on veut utiliser des données.
- On veut stocker ces données dans ce qu'on appelle des variables.
- On veut programmer un jeu vidéo.
- On a un joueur de ce jeu vidéo qui veut manipuler un personnage.
- Ce personnage va avoir une position dans une carte.
- Cette position va changer en fonction du mouvement du personnage.
- On veut stocker cette information pour pouvoir interagir avec d'autres éléments du jeu.

Il y en a six : `void`, `int`, `char`, `float`, `double`, `long double`.

- `void` : c'est le type vide. Il est surtout utilisé pour définir les fonctions sans arguments ou sans valeur de retour.
- `int` : c'est un type qui se décline sous plusieurs formes.
  - `short` : un entier court.
  - `long` : un entier long.
  - `signed` pour dire que l'entier qu'on manipule peut être positif ou négatif.
  - `unsigned` pour dire que l'entier qu'on manipule est uniquement positif**N.B. Si on n'indique rien, le qualificatif `signed` est appliqué.**

- `char` : ce type permet de stocker les caractères. Il peut aussi représenter les entiers sur 8 bits. On, peut aussi trouver un char signé `signed char` ou non signé `unsigned char`.
- `float` : ce type permet de représenter les réels.
- `double` : même chose que `float` mais avec une précision plus importante.
- `long double` : pareil que `double` mais avec une précision encore plus grande.

- La fonction `printf` est très utile car permet d'afficher des messages et les valeurs des variables.
- Exemple :

```
printf("Bonjour tout le monde!");  
printf("%d kilogramme vaut %d grammes", 1, 1000);
```

- L'argument de la fonction `printf` dit **format** est une chaîne de caractère qui détermine ce qui sera affiché par `printf` et sous quelle forme.
- Dans l'exemple 1 c'est "Bonjour tout le monde!".

```
printf("Bonjour tout le monde!");  
printf("%d kilogramme vaut %d grammes", 1, 1000):
```

- Dans l'exemple 2 c'est "%d kilogramme vaut
- Dans l'exemple 2 : la chaîne de caractère qu'on veut afficher est composée d'un texte normal et de séquence de contrôle permettant d'inclure des variables.
- le premier %d sera donc remplacé par la valeur 1.
- le deuxième %d sera remplacé par la valeur 1000.
- On aura comme résultat en sortie l'affichage suivant :  
"1 kilogramme vaut 1000 grammes".

Les séquences de contrôle commencent par le caractère % suivi d'un caractère parmi les suivants :

- d ou i pour afficher un entier signé.
- f pour afficher un réel (float ou double).
- c pour afficher comme un caractère.
- s pour afficher une chaîne de caractères.



- Cette fonction permet de lire des données formatées à partir de l'entrée standard (clavier).
- `scanf` utilise les mêmes formats que `printf` mais on fait précéder le nom de la variable du caractère `&`.

```
scanf("%d",&i);
```

- seul le format est passé en paramètre.
- Il ne faut ajouter de message ni aucun autre caractère.

```
/* Exemple pour tester "scanf" */
#include <stdio.h>
int main () {
    int nb1 ;
    float nb2 ;
    printf("Saisissez une valeur entiere (positive ou negative) pour nb1 : ") ;
    scanf("%d",&nb1) ;
    printf("Saisissez une valeur reelle pour nb2 : ") ;
    scanf("%f", &nb2) ;
    printf("nb1 vaut %d ; nb2 vaut %f\n", nb1, nb2) ;
    return 0;
}
```

- Éditer, sauvegarder puis compiler ce code.
- Faites en sorte que le programme affiche "hello monde" après avoir affiché les valeurs saisies par l'utilisateur.