

UE INFORMATIQUE

Programmation 1 : Fondamentaux

Laureen Fabre

TABLEAUX STATIQUES

```
// le tableau que l'on connait pour une chaine de caractere
{
    char chars[6] = "Hello"; // Inclut automatiquement le caractère nul '\0' à la fin
    printf("Chaine : %s\n", chars);
}
printf("\n");
// le tableau d'entiers
{
    int integers[3] = { 10, 20, 30 };
    for (int i = 0; i < 3 ; i++) {
        printf("Entier %d: %d\n", i + 1, integers[i]);
    }
}
```

Chaine : Hello

Entier 1: 10

Entier 2: 20

Entier 3: 30

Float 1: 1.1

Float 2: 2.2

Float 3: 3.3

TABLEAUX STATIQUES

```
// le tableau de flottant (idem pour les double on change juste le type)
{
    float floats[3] = { 1.1f, 2.2f, 3.3f };
    for (int i = 0; i < 3; i++) {
        printf("Float %d: %.1f\n", i + 1, floats[i]);
    }
}

// Déclaration d'un tableau de chaînes de caractères
char phrases[6][200] = {"Bonjour", "le", "monde", "en", "C"};

// Calcul du nombre de chaînes dans le tableau
int count = sizeof(phrases) / sizeof(phrases[0]);

// Affichage de chaque chaîne de caractères
for (int i = 0; i < count; i++) {
    printf("%s\n", phrases[i]);
}
}
```

Float 1: 1.1
Float 2: 2.2
Float 3: 3.3

Bonjour
le
monde
en
C

C
eu

LA PORTÉE

- Une variable a une portée définie.
- Elle dépend de là où elle est créée.
- Pour schématiser sa portée ne dépasse pas les accolades dans lesquelles elle a été créée.

```
#include <stdio.h>

int varGlobal = 8; // variable globale au fichier, on y accède de partout

int main()
{
    int varFonctionMain = 2; // variable accessible dans toute la fonction main

    {
        int varAccolade = 5; // variable varAccolade accessible seulement ici
    }
    varAccolade = 3; // ERREUR

    for (int i = 0; i < varGlobal; i++)
    {
        printf("%d ", i); // variable i accessible seulement ici
    }

    i = 8; // ERREUR

    if (varFonctionMain == 3)
    {
        float age = 25.0f; // variable age accessible seulement ici
    }
    else
    {
        age = 32.0f; // ERREUR il faudrait de nouveau déclarer age
    }

    return 0;
}
```

STRUCTURER LE CODE GRÂCE AUX FONCTIONS

- Pour plus de clarté dans le programme, on peut le découper en fonctions.
- Une fonction permet de faire une action « spécifique ».
- Un gros avantage à utiliser une fonction est de pouvoir la réutiliser sans avoir à refaire le même code encore et encore.
- Elle va avoir un *nom*, des *paramètres* et une *valeur de retour*.
- Les fonctions sont déclarées en dehors du *main* et **au dessus** de celui-ci.
- Le code qu'effectue la fonction se trouve entre **accolades**.
- Le mot clé *return* permet de renvoyer la valeur de retour.

```
float multiplication(float a, float b)
{
    float resultat = a * b;
    return resultat;
}
```

STRUCTURER LE CODE GRÂCE AUX FONCTIONS

- Pour *appeler* une fonction, il suffit de l'écrire dans le programme en lui donnant ses paramètres. Si elle a une valeur de retour on *l'affecte* dans une variable.

```
#include <stdio.h>

int multi(int a, int b)
{
    return a * b;
}

int main() {

    int var = 4;
    int var2 = 5;

    int result = multi(var, var2);

    printf("resultat = %d", result);

    return 0;    resultat = 20
}
```

LES HEADERS

- En C, les headers (.h) servent de **contrats** entre différentes entités compilées séparément.
- Les headers ne contiennent pas d'implémentations
- Ils contiennent principalement la **déclaration** des fonctions et des structures.
- C'est le .c qui contiendra leurs **définitions** et donc leurs implémentations.
- On va utiliser une *directive de préprocesseur* pour que les fichiers sources ne soient inclus qu'une seule fois à la compilation. On fait cela pour éviter que le compilateur tourne en rond.

LES HEADERS

```
menu.h
1 #ifndef MENU_HPP
2 #define MENU_HPP
3
4 #include <stdio.h>
5 #include <string.h>
6
7 int menu();
8 float paiement(float argent, int burger);
9
10 #endif

PreparationCoursC - main.c
1 #include "menu.h"
2
3 int main(int argc, char* argv[])
4 {
5
6     printf("Combien d'argent avez vous ?\n");
7     float argent = 0;
8     scanf_s("%f", &argent);
9     int choix = menu();
10    argent = paiement(argent, choix);
11
12 }
```

```
menu.c
1
2 #include "menu.h"
3
4 int menu()
5 {
6     printf("== = Menu = ==\n");
7     printf("1. Royal Cheese \n");
8     printf("2. Mc Deluxe \n");
9     printf("3. Mc Bacon \n");
10    printf("4. Big Mac \n");
11    printf("Votre choix ? \n");
12    int choix = 0;
13    scanf_s("%d", &choix);
14    return choix;
15 }
16
17
18 float paiement(float argent, int burger)
19 {
20     if (burger == 1)
21     {
22         float prix = 4.95;
23         if (argent >= prix)
24             argent -= prix;
25     }
26     else
27     {
28         printf("pas assez d'argent");
29         return argent;
30     }
31     printf("Votre Royal Cheese coute %f euros, il ne vous reste plus que %f euros", argent, argent - prix);
32 }
```

```
Console de débogage Micros
Combien d'argent avez vous ?
45.2
== = Menu = ==
1. Royal Cheese
2. Mc Deluxe
3. Mc Bacon
4. Big Mac
Votre choix ?
1
Votre Royal Cheese coute 4.950000 euros, il ne vous reste plus que 40.250000 euros
```




MERCI