

UE INFORMATIQUE

Programmation 1 : Fondamentaux

Laureen Fabre

TABLEAUX STATIQUES

```
// le tableau que l'on connaît pour une chaîne de caractères
{
    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]);
    }
}

printf("Entier %d: %d\n", i + 1, integers[i]);
```

```
Chaine : Hello
Entier 1: 10
Entier 2: 20
Entier 3: 30
Entier 1: 1.1
Entier 2: 2.2
Entier 3: 3.3
Entier 4: 3.3
Entier 5: 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
#ifndef MENU_HPP
#define MENU_HPP

#include <stdio.h>
#include <string.h>

int menu();
float paiement(float argent, int burger);

#endif

main.c
#include "menu.h"

int main(int argc, char* argv[])
{
    printf("Combien d'argent avez vous ?\n");
    float argent = 0;
    scanf_s("%f", &argent);
    int choix = menu();
    argent = paiement(argent, choix);
}

menu.c
#include "menu.h"

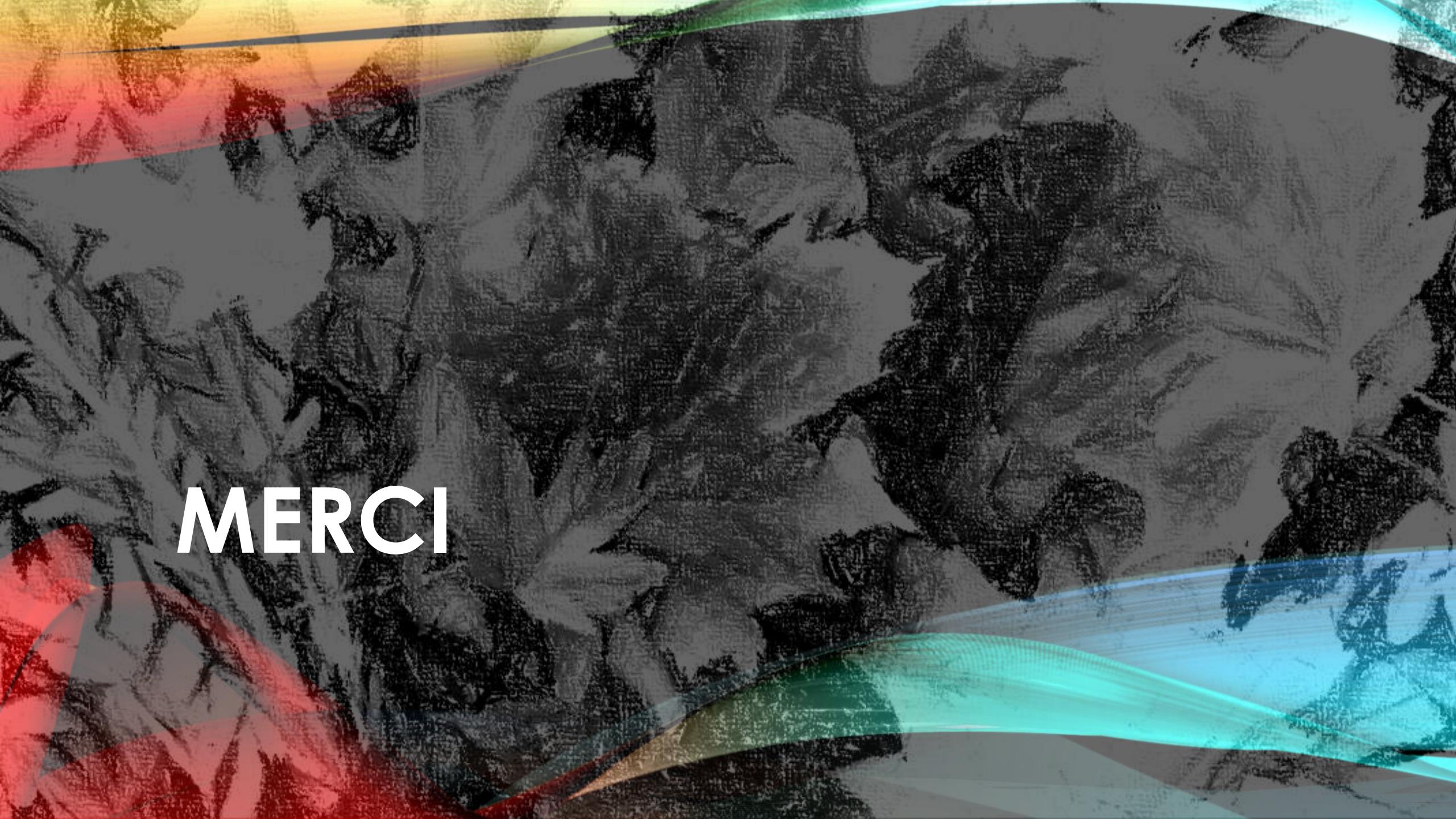
int menu()
{
    printf("== = Menu ==\n");
    printf("1. Royal Cheese\n");
    printf("2. Mc Deluxe\n");
    printf("3. Mc Bacon\n");
    printf("4. Big Mac\n");
    printf("Votre choix ?\n");
    int choix = 0;
    scanf_s("%d", &choix);
    return choix;
}

float paiement(float argent, int burger)
{
    if (burger == 1)
    {
        float prix = 4.95;
        if (argent >= prix)
            argent -= prix;
        else
            printf("pas assez d'argent");
        return argent;
    }
    printf("Votre Royal Cheese coute %f euros, il ne vous reste plus\nque %f euros\n", prix, argent - prix);
}
```

Console de débogage Micros... X + - □ X

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

Aucun problème détecté | Lig.: 74 Car.: 2 TABS CR LF



MERCI