



Anthony Menghi contact@antocrea.dev

Mehdi Ghoulam mehdi@gelk.fr

C - Manifest

Code source

A. Indentation

Bien qu'en web, nous avons pour habitude d'utiliser 2 espaces pour 1 tabulation car l'imbrication d'instruction est plus courante.

Nous allons, pour le langage C, fixer l'indentation à 4 espaces par tabulation.

B. Convention de nommage

En C, la convention la plus répandue est l'écriture au format `snake_case`. Les mots sont écrits en minuscules et sont séparés par des `underscore`.

B-1. Constantes et Macros

Elles utilisent la directive de préprocesseur `#define`, par convention, on les écrits en majuscules, chaque mots séparés par des `underscore`.

C. Organisation du projet

C-1. Structure des dossiers

```
.
├── Makefile
├── bin
│   └── sum
├── obj
│   └── sum.o
└── src
    ├── include
    │   └── header.h
    └── sum.c
```

C-2. Code source

```
/**
 * Documentation section
 * @Description: Untitled-1
 * @Author: Mehdi Ghoulam<mehdi@gelk.fr>
 * @Created: 2024-02-02
 */

/* Inclusion des librairies ===== */
/* Macros et constantes ===== */
/* Variables globales ===== */
/* Déclaration des prototypes ===== */
/* Fonction principale ===== */
/* Fonctions secondaires ===== */
```

Référence : M-L Nivet -> StructureProgramme.c pour les L3

C - Cheat Sheet

Project base

```
/* Standard Input-Output library */
#include<stdio.h>

int main(void)
{
    printf("Hello, world!\n");
}
```

Data types

type	bytes	bits
int	4 bytes	32 bits
char	1 byte	8 bits
float	4 bytes	32 bits
double	8 bytes	64 bits

Note : The **boolean** data type is include in the `stdbool.h` library.

You can also create your own `bool` type using `typedef`

```
typedef enum {false, true} bool;
```

Or you can define other types.

```
typedef unsigned char BYTE;
typedef struct Person {
    char *name;
} Person;
```

Declaring variables

The structure is simple : `type variable_name; .`

```
int my_integer;
```

You can also declare multiple variables at once in a single line.

```
int price, total;
```

You can assign and initialize values with `=` .

```
int my_age = 21;
float product_price = 69.42;
char letter = 'A';
char *name = "Anthony";
```

Note : Working with chars must imply using single quotes eg. 'A' . Strings uses double quotes eg. "Mehdi" .

Conditionals

- Classic if-else syntax

```
if (my_age > your_age)
{
    printf("I'm older");
}
else if (my_age < your_age)
{
    print("I'm younger");
}
else
{
    printf("We have the same age");
}
```

- Ternary operation

```
int message = age >= 18 ? "You can pass" : "Not allowed";
```

- Switch-case

```
switch (response)
{
    case 'Y':
        printf("yes");
        break;
    case 'N':
        printf("no");
        break;
    default:
        printf("Invalid response");
}
```

Loops

```
while (some_boolean_expression)
{
    printf("Do something");
}

do
{
    printf("Something");
} while (condition);

for (int i = 0; i < 10; i++)
{
    printf("Counting ...");
}
```

Credits

1. [CS50 Cheat Sheet by Ruan](#)

Car j'ai moi-même suivi le cours de CS50, et j'ai appris le C avec ce cours. Il explique vraiment les fondamentaux de la programmation et des concepts du langage, de la gestion de mémoire, des algorithmes, des structures de données, etc ...

2. [Bonnes pratiques C](#)

Lecture approfondie, inspiration des parties