



préprocesseur  
Dorian.H Mekni  
13 | 02 | 2020

# Le préprocesseur

- Un sous programme qui va s'exécuter en tout premier
- Il opère les changements selon les directives qu'on lui donne
- Puis après on a l'étape de la compilation

# Xcode

```
#include <stdio.h>
#include "player.h"
#define MARGE 0.90 // 10 %

int main(void)
{
    int prix = 20;
    int prixTotal = 0;

    prixTotal = prix / MARGE;
    printf("le prix total est de %d euros\n", prixTotal);

    return 0;
}
```

le prix total est de 22 euros  
Program ended with exit code: 0

le prix total est de 22 euros  
Program ended with exit code: 0

# #define

- Une manière d'introduire une const en dessous des bibliothèques déjà connues telle que `#include <studio.h>`
- Pour changer sa valeur, on ne la change qu' 1 seule fois en haut du fichier ce qui modifiera sa valeur à travers toutes ses utilisations tout au long du fichier.
- `#define MARGE 0.90` | par exemple - cette dernière est souvent soumise à des augmentations ou bien des baisses.

# .h

Sur les fichier header, on placera nos prototype de fonctions à **l'intérieur** de:

```
#ifndef  
#define
```

```
void bonjour(void);
```

```
#endif
```

# Simplexifier

```
8
9 #include <stdio.h>
10 #include "player.h"
11 #define Hiya printf("Hiya !\n")
12
13 int main(void)
14 {
15     Hiya;|
16
17     return 0;
18 }
19
20
```

Hiya !  
Program ended with exit code: 0

Prédéfinis dans #define au préalable, on l'appelle par son alias.  
Si les noms des fonction sont longues, cela peut être très utile.

# Const prédéfinis

/\*

Const predef dans le C :

\_\_FILE\_\_ (nom du fichier)

\_\_LINE\_\_ (ligne du fichier)

\_\_DATE\_\_ (date de compilation)

\_\_TIME\_\_ (heure de compilation)

\*/