

URL du tuto : <http://fr.openclassrooms.com/informatique/cours/apprenez-a-programmer-en-c/la-programmation-modulaire>

Concept : Séparer les fonctions dans plusieurs fichiers

Les prototypes

A placer avant le *main.c* afin d'indiquer la présence de cette fonction à l'ordinateur.

```
Ex : double aireRectangle(double, double);
```

Les headers

Contient les prototypes des fonctions. Chaque fichier *.c* à son équivalent en *.h*.

Il faut inclure les *.h* grâce à une directive de préprocesseur *#include* dans le *.c*

Pour inclure les fichiers se trouvant dans le répertoire :

- *include* de l'IDE on utilise des chevrons < >
- du projet on utilise les guillemets " "

La compilation séparée

1. Préprocesseur : démarre avant la compilation et exécute les directives de préprocesseur (ex : *#include* -> inclus les *.h* dans les *.c*).
2. Compilation : transforme fichiers sources en code binaire en générant un fichier *.o* qui sont temporaires
3. Editions de liens : assemble les *.o* et les bibliothèques en un seul fichier : l'exécutable

La portée des fonctions et variables

- Les variables propres aux fonctions

Celles-ci sont supprimées de la mémoire à la fin de la fonction. On ne peut pas y accéder depuis une autre fonction.

- Les variables globales accessibles dans tous les fichiers (à éviter)

La déclaration se fait en dehors des fonctions, après les *#includes*.

Ce type de variables est généralement à bannir. Il vaut mieux utiliser le *return*.

- Les variables globales accessibles dans un seul fichier

Il faut simplement rajouter le mot clé *static* devant la déclaration de la variable.

- Les variables statiques à une fonction

Ces variables ne seront pas supprimées à la fin de l'exécution de la fonction.

- Les fonctions locales à un fichier

Il faut rajouter le mot clé *static* devant la déclaration de la fonction (même dans le prototype).