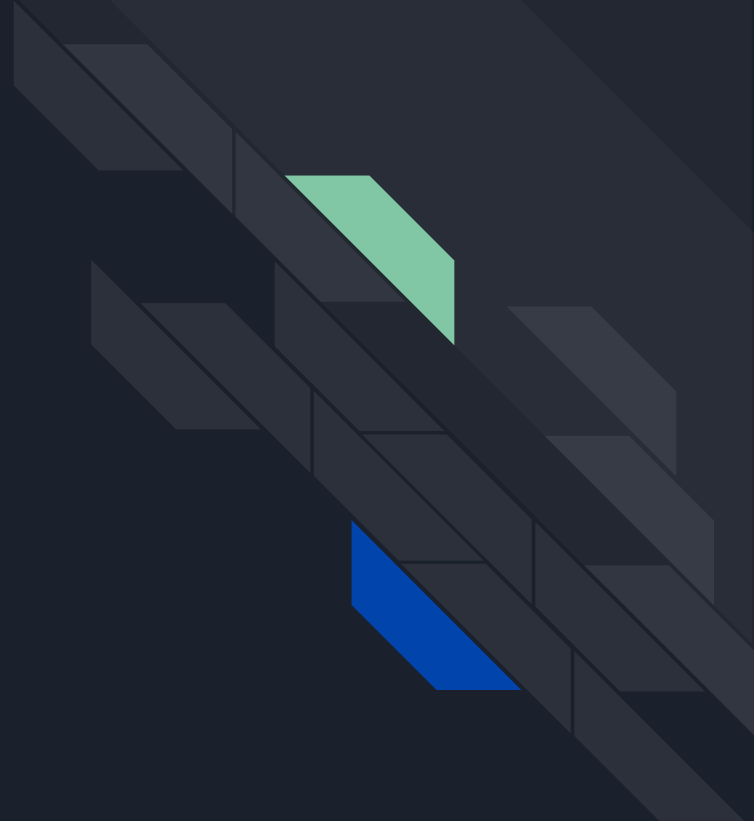




# 1h1 Outil Makefile





# Utilisation Générale

## Makefile sert à :

1. Compiler facilement son projet
2. Éviter de reconstruire des fichiers non modifiés

A large, stylized number '4' composed of three parts: a blue circular arc at the top left, a white vertical bar in the middle, and a green diagonal bar at the bottom right.

# Principe de fonctionnement



# Fonctionnement du makefile

## Principe de base :

- On créer un fichier nommé **Makefile** et l'exécute avec **make** [**<règle>**]
- Fonctionne avec des règles pour déterminer comment construire un fichier
- Se base sur les dates de modifications de fichiers pour savoir s'il a été changé



# Créer son Makefile

## Les règles :

**<cible>: [<prérequis>]**

### Commandes

La **cible** représente le fichier qu'on veut créer.

Le **prérequis** représente le fichier dont on a besoin.

Les commandes sont les instructions en shell pour passer des prérequis à la cible.

Les prérequis peuvent être aussi des règles



# Créer son Makefile

## Les variables :

Définition :

**NOM** := valeur

Utilisation :

**\$(NOM)**

## Les variables prédéfinies :

**CC** -> représente le compilateur

**CFLAGS** -> représente les flags du compilateur C



# Créer son Makefile

## Les variables de règle :

`$@` -> représente la cible de la règle

`$<` -> représente le premier prérequis

`$^` -> représente tous les prérequis

`%` -> représente une partie de texte à remplacer, comme un wildcard



# Créer son Makefile

## Manipulation des variables :

`+=` -> ajoute une valeur à une variable

Conditions :

`(ifdef / ifndef) VAR`

`else`

`endif`

`${VAR:old=new}` -> remplacement syntaxique de ce qu'il y a dans VAR





# Créer son Makefile

## Règles spécifiques :

**<nom>:** [**<prérequis>**]

Commandes

**.PHONY:** **<nom>**

Le .PHONY permet d'indiquer au makefile que la cible n'est pas un fichier. Cela nous assure que la règle se lancera tout le temps.

Par défaut, la règle qui est appelée lorsqu'on utilise **make** sans argument sera la première du Makefile.



# Créer son Makefile

## Quelques notions supplémentaires :

- Un Makefile peut appeler un autre makefile dans ses commandes, y compris lui même
- Il existe d'autres variables par défaut, comme **.CMAKEFLAGS** pour gérer le comportement du Makefile.
- En mettant @ devant nos commandes dans les règles, celle ci ne se voit plus dans le terminal

**Merci pour votre attention !**



**C4A**

Code For All, école 42