

# Installer C sur son PC

## Installation de VSCode (5 min)

Quel que soit votre système d'exploitation, je vous recommande d'utiliser Visual Studio Code (VSCode) pour développer en C. Un tuto pour l'installer, et installer des extensions, est disponible sur [cette page WikiEtud](#).

De plus, pour faire du C, il vous faut installer au minimum les extensions VSCode suivante (vous pouvez en installer d'autre en plus selon vos préférences) :

- `C/C++` (par Microsoft)
- `C/C++ Extension Pack` (par Microsoft)
- `C/C++ Themes` (par Microsoft)

## Installation de C sur Windows (15 min)

Je vous recommande spécialement d'installer VSCode si vous êtes sur Windows. En effet, VSCode permet de se connecter directement sur un WSL, ce qui rendra l'expérience plus agréable.

La méthode vous est expliquée dans le titre précédent.

Il est difficile d'installer `gcc` sur Windows de manière native (la plus part des méthodes sont trop vieilles et / ou dépréciées). De ce fait, nous allons utiliser un WSL (Windows Subsystem Linux) pour nous simplifier la tâche.

### 1. Installation de WSL (10-15 min)

Si vous avez déjà installé WSL, vous pouvez sauter cette étape.

Un tutoriel pour installer WSL est disponible sur [cette page WikiEtud](#). Afin de simplifier le point suivant, je supposerais que vous avez installé Ubuntu ou Debian.

Si vous utilisez une autre distribution que Ubuntu ou Debian, il vous faudra adapter les commandes à votre distribution (voir la partie sur l'installation de C sur Linux).

## 2. Installation de `gcc`, `gdb` et `make` (2 min)

Afin d'installer ces 3 paquets qui vous seront nécessaires pour les TD et TP de C, suivez les points suivants :

1. Ouvrez un Terminal (recherchez Terminal dans la barre de recherche Windows)
2. Connectez-vous à votre WSL
  - Si vous voulez vous connecter à votre WSL par défaut, rentrez la commande `wsl`
  - Si vous voulez vous connecter à un autre WSL, rentrez la commande `wsl -d NomDeVotreWSL` (en remplaçant évidemment le nom)
3. Une fois dans votre WSL, rentrez ces commandes :

```
sudo apt update
sudo apt install gcc gdb make

# Cette commande permet de mettre à jour tous les paquets de votre WSL
# Vous pouvez la sauter, même si je la recommande
sudo apt upgrade
```

Pour maintenant utiliser ces commandes comme en salle de TP, vous devrez d'abord vous connecter à votre WSL, et rentrer les commandes dans le bon dossier depuis votre celui-ci.

## 3. Connectez vous à votre WSL dans VSCode

La méthode pour se connecter à un WSL est similaire à celle pour connecter VSCode à un ordinateur distant. Pour vous connecter à WSL, vous pouvez donc suivre [le tutoriel présent sur cette page WikiEtud](#), en prenant en compte le paragraphe en bas de page.

# Installation de C sur Mac (5 min)

Pour faciliter l'installation de nos paquets, nous allons utiliser le gestionnaire de paquets `homebrew`, non présent par défaut sur Mac.

## 1. Installation de Homebrew (2 min)

1. Ouvrez un terminal
2. Copiez-collez la commande suivante pour installer Homebrew

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

## 2. Installation de `gcc`, `gdb` et `make` (2 min)

Toujours dans votre terminal (réouvrez en un si vous l'avez fermé), rentrez les commandes suivantes:

```
brew install gcc  
brew install gdb  
brew install make
```

Vous pouvez à présent utiliser ces commandes comme en cours.

# Installation de C sur Linux (2 min)

Vous devez seulement rentrer quelques commandes dans votre terminal. Selon votre distribution, celles-ci peuvent légèrement changer.

Si vous ne savez pas quelle distribution vous avez, celle-ci se retrouve avec la commande suivante:

```
cat /etc/os-release
```

 (attention à l'espace après le égal)

Vous retrouverez également la version de cette dernière si besoin.

Si votre distribution n'apparaît pas dans la liste, elle a de grandes chances d'être basé sur une distribution présente dans celle-ci. Vous pouvez utiliser les mêmes commandes.

## Debian / Ubuntu / Mint

```
sudo apt update  
sudo apt install gcc gdb make
```

## RedHat / CentOS (Version $\leq 7$ )

```
sudo yum install gcc gdb make
```

## Fedora / CentOS (Version $\geq 8$ )

```
sudo dnf install gcc gdb make
```

## Arch / Manjaro

Que faites-vous ici ? Vous vous êtes perdus ?

```
sudo pacman -S gcc gdb make
```

## Gentoo

```
# Si les commandes suivantes sont erronées, merci de les corriger  
# Je n'ai jamais utilisé emerge auparavant :'(
```

```
sudo emerge -a sys-devel/gcc  
sudo emerge -a dev-debug/gdb  
sudo emerge -a dev-build/make
```

Si vous n'arrivez pas à installer ces outils, vous pouvez venir sur le [discord du Club Info](#) pour demander de l'aide.

---

Révision #3

Créé 29 juin 2025 21:52:07 par Clément Duran

Mis à jour 29 juin 2025 23:28:47 par Clément Duran