

Grafica Setup

Windows

Python

Descargamos la ultima version estable de Python aquí:

<https://www.python.org/downloads/windows/>

Python Releases for Windows

- [Latest Python 3 Release - Python 3.12.2](#)

Stable Releases

- [Python 3.9.19 - March 19, 2024](#)
Note that Python 3.9.19 cannot be used on Windows 7 or earlier.
 - No files for this release.
- [Python 3.8.19 - March 19, 2024](#)
Note that Python 3.8.19 cannot be used on Windows XP or earlier.
 - No files for this release.
- [Python 3.11.8 - Feb. 6, 2024](#)
Note that Python 3.11.8 cannot be used on Windows 7 or earlier.
 - [Download Windows installer \(64-bit\)](#)
 - [Download Windows installer \(ARM64\)](#)
 - [Download Windows embeddable package \(64-bit\)](#)
 - [Download Windows embeddable package \(32-bit\)](#)
 - [Download Windows embeddable package \(ARM64\)](#)
 - [Download Windows installer \(32-bit\)](#)
- [Python 3.12.2 - Feb. 6, 2024](#)

Pre-releases

- [Python 3.13.0a5 - March 12, 2024](#)
 - [Download Windows installer \(64-bit\)](#)
 - [Download Windows installer \(ARM64\)](#)
 - [Download Windows embeddable package \(64-bit\)](#)
 - [Download Windows embeddable package \(32-bit\)](#)
 - [Download Windows embeddable package \(ARM64\)](#)
 - [Download Windows installer \(32-bit\)](#)
- [Python 3.13.0a4 - Feb. 15, 2024](#)
 - [Download Windows installer \(64-bit\)](#)
 - [Download Windows installer \(ARM64\)](#)
 - [Download Windows embeddable package \(64-bit\)](#)
 - [Download Windows embeddable package \(32-bit\)](#)
 - [Download Windows embeddable package \(ARM64\)](#)
 - [Download Windows installer \(32-bit\)](#)
- [Python 3.13.0a3 - Jan. 17, 2024](#)
 - [Download Windows installer \(64-bit\)](#)

Abrimos el instalador y **TENEMOS** que seleccionar `Add python.exe to PATH`, si no lo hacen tienen que empezar de nuevo la instalación :)

Python 3.11.8 (64-bit) Setup

Install Python 3.11.8 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\Emilia Coloma\AppData\Local\Programs\Python\Python311

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ **Customize installation**
Choose location and features

☒ Use admin privileges when installing py.exe
☒ **Add python.exe to PATH**

Cancel

Cuando la instalación termine, habren una terminal y escriben:

```
python
```

Si todo está bien deberían obtener lo siguiente o algo similar.

```
Python 3.11.8 (tags/v3.11.8:db85d51, Feb 6 2024, 22:03:32) [MSC v.1937 64 bit  
(AMD64)] on win32  
Type "help", "copyright", "credits" or "license" for more information.  
>>
```

No queremos usar esto así que lo cerramos con

```
>>> exit()
```

Ahora, el instalador debería haber instalado pip junto a python. Prueben si lo tienen escribiendo en la terminal

```
pip
```

Si les aparece un texto largo con comandos están bien :).

Si les dice que el comando no existe (*no debería pasar pero si es que pasa...*) tienen que instalar pip como dicen por aquí: <https://www.liquidweb.com/kb/install-pip-windows/>

Creación del venv

(Para Powershell, si tienen otro shell me piden ayuda pero si tienen otro shell probablemente no necesiten mi ayuda)

Vamos a usar los virtual environments (*venv*) de Python para contener las librerías que necesitamos.

Primero en la terminal vamos a un lugar donde queramos guardar nuestro trabajo.

Al abrir una terminal nueva siempre se van a encontrar en el `home`.

Aquí voy a crear una carpeta llamada `Grafica` con el comando `mkdir`

```
mkdir Grafica
```

Luego con el comando `cd` me muevo dentro de esa carpeta:

```
cd Grafica
```

Ahora creamos un `venv` escribiendo lo siguiente:

```
python -m venv venv
```

Aquí el segundo `venv` es un nombre que ustedes pueden cambiar si quieren

Se debería haber creado una carpeta llamada `venv`

Ahora queremos ejecutar un script que **activa el `venv`**. Para ello, estando en la carpeta donde esta `venv` vamos a escribir:

```
venv\Scripts\Activate.ps1
```

Si esto le produce algún error, lo más probable es que por defecto windows no les permite ejecutar scripts (*medida de seguridad, no descarguen scripts random de internet porfavor*). Para bypasear eso ejecutamos lo siguiente:

```
Set-ExecutionPolicy AllSigned -Scope CurrentUser
```

y volvemos a intentar.

Si salió bien deberían ver un parentesis de color al inicio de su línea en la terminal indicando que están en el `venv`.

```
(venv) PS C:\Users\su_nombre\Grafica
```

Git

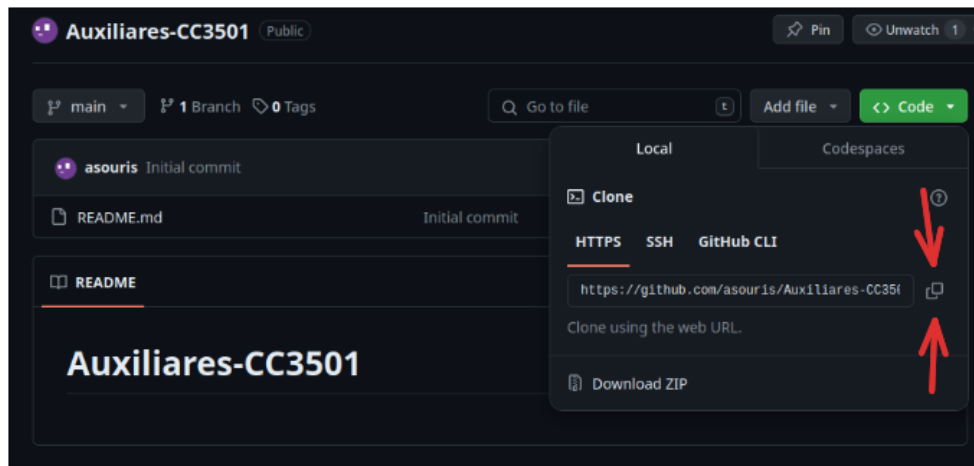
Instalen Git con

```
`winget install --id Git.Git -e --source winget`
```

Clonar repositorio

Durante las próximas semanas les voy a subir a un repo en github ejemplos, shaders, los códigos de los aux y otras cosas útiles.

Con git pueden clonar el repositorio en su máquina. Primero copian el link.



Al clonar el repo se les va a crear una carpeta, así que antes, **en la terminal** vayan donde quieren colocar el repo (*vayan dentro del venv*). Luego usan `git clone` para copiar el repo

```
cd Grafica/venv/  
git clone https://github.com/asouris/CC3501.git
```

MacOs

(Para Zsh, si tienen otro shell me piden ayuda pero si tienen otro shell probablemente no necesiten mi ayuda)

Python

Mac viene con una versión antigua de python. Si escriben en la terminal:

```
python3
```

Podría ser que

1. Tengan python y vean algo así:

```
Python 3.6.6 (default, Sep 12 2018, 18:26:19)  
[GCC 8.0.1 20180414 (experimental) [trunk revision 259383]] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>
```

2. No lo tengan y pero tengan xcode y se ponga a instalarlo
3. No lo tenga y punto

Si son el caso 3. vamos a instalarlo con **brew**. Vean si ya tienen brew con

```
brew
```

Si les dice que no conoce el comando lo instalan, si ya lo tenían sigan con la instalación. Instalar brew es muy simple, sigan las instrucciones que hay en <https://brew.sh/> (*copian lo que les dice y lo ponen en la terminal*)

Ahora instalan **python** con:

```
brew install python3
```

Creacion del venv

Vamos a usar los virtual environments (**venv**) de Python para contener las librerías que necesitamos.

Primero en la terminal vamos a un lugar donde queramos guardar nuestro trabajo.

Al abrir una terminal nueva siempre se van a encontrar en el `home`.

Aquí voy a crear una carpeta llamada `Grafica` con el comando `mkdir`

```
mkdir Grafica
```

Luego con el comando `cd` me muevo dentro de esa carpeta:

```
cd Grafica
```

Ahora creamos un `venv` escribiendo lo siguiente:

```
python3 -m venv venv
```

Aquí el segundo venv es un nombre que ustedes pueden cambiar si quieren

Se debería haber creado una carpeta llamada **venv**

Ahora queremos ejecutar un script que activa en **venv**. Para ello vamos a ejecutar:

```
source ~/Grafica/venv/bin/activate
```

(El path puede cambiar dependiendo de como organizaron sus carpetas)

Aparecerá (**venv**) al lado izquierdo de su prompt, indicando que dicho environment se encuentra activo.

Git

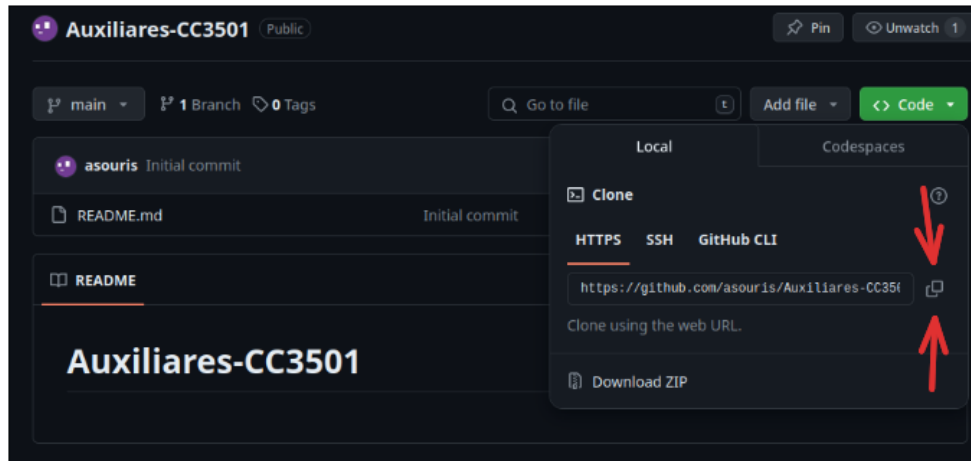
Instalen git ejecutando:

```
brew install git
```

Clonar repositorio

Durante las proximas semanas les voy a subir a un repo en github ejemplos, shaders, los códigos de los aux y otras cosas útiles.

Con git pueden clonar el repositorio en su maquina. Primero copian el link.



Al clonar el repo se les va a crear una carpeta, así que antes, **en la terminal** vayan donde quieren colocar el repo (*vayan dentro del venv*). Luego usan `git clone` para copiar el repo

```
cd Grafica/venv
git clone https://github.com/asouris/CC3501.git
```

Linux (Debian/Ubuntu)

(Para bash, si tienen otro shell me piden ayuda pero si tienen otro shell probablemente no necesiten mi ayuda)

Python

Primero vemos si tenemos python. Ejecuten:

```
python3
```

Podría ser que

1. Tengan python y vean algo así:

```
Python 3.6.6 (default, Sep 12 2018, 18:26:19)
[GCC 8.0.1 20180414 (experimental) [trunk revision 259383]] on linux
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>>
```

2. Les tire error y no lo tienen

Si no tienen python instalado, ejecuten en el mismo orden lo siguiente:

```
sudo apt-get update
sudo apt-get install python3 python3-dev
sudo apt-get install python3-pip
sudo apt-get install python3-venv
```

Creacion del venv

Vamos a usar los virtual environments (**venv**) de Python para contener las librerías que necesitamos.

Primero en la terminal vamos a un lugar donde queramos guardar nuestro trabajo.

Al abrir una terminal nueva siempre se van a encontrar en el **home**.

Aquí voy a crear una carpeta llamada **Grafica** con el comando **mkdir**

```
mkdir Grafica
```

Luego con el comando **cd** me muevo dentro de esa carpeta:

```
cd Grafica
```

Ahora cree el venv ejecutando:

```
python3 -m venv venv
```

Aquí el segundo venv es un nombre que ustedes pueden cambiar si quieren

Se debería haber creado una carpeta llamada **venv**.

Ahora queremos ejecutar un script que **activa el venv**. Para ello vamos a escribir:

```
source ~/Grafica/venv/bin/activate
```

(El path puede cambiar dependiendo de como organizaron sus carpetas)

Aparecerá **(venv)** al lado izquierdo de su prompt, indicando que dicho environment se encuentra activo.

Git

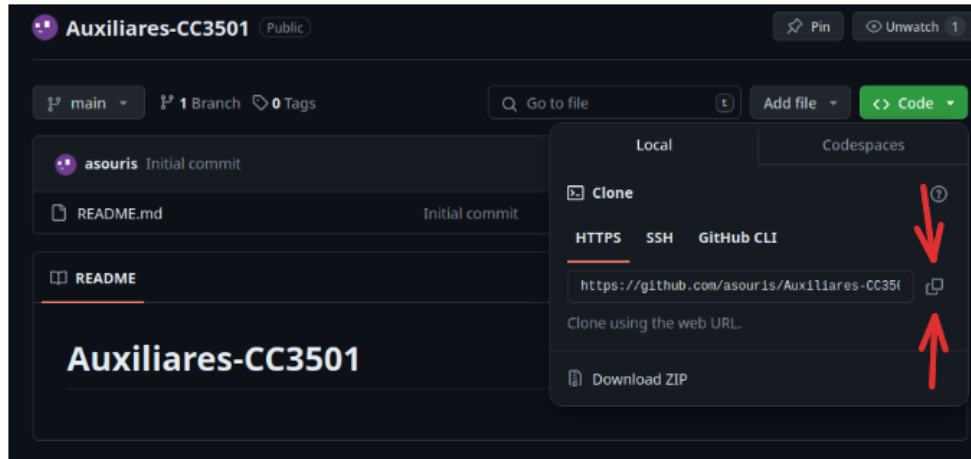
Instalen con

```
sudo apt-get install git
```

Clonar repositorio

Durante las proximas semanas les voy a subir a un repo en github ejemplos, shaders, los códigos de los aux y otras cosas útiles.

Con git pueden clonar el repositorio en su maquina. Primero copian el link.



Al clonar el repo se les va a crear una carpeta, así que antes, **en la terminal** vayan donde quieren colocar el repo (*vayan dentro del venv*). Luego usan `git clone` para copiar el repo

```
cd Grafica/venv
git clone https://github.com/asouris/CC3501.git
```

Probando un ejemplo (para todos)

Con nuestro **venv activo**, vamos a instalar las librerías:

```
pip install numpy pygame pyopengl
```

Y habiendo clonado el repositorio previamente, vamos a ejecutar un ejemplo llamado **ej_aux1.py**

```
python CC3501/ejemplos/ej_aux1.py
```

Si todo sale bien debería haberse una ventana y veríamos algo así:

