Guía de instalación

Programación Avanzada

Python 3.12

- Para Mac y windows recomendamos descargar el instalador:

https://www.python.org/downloads/release/python-3129/

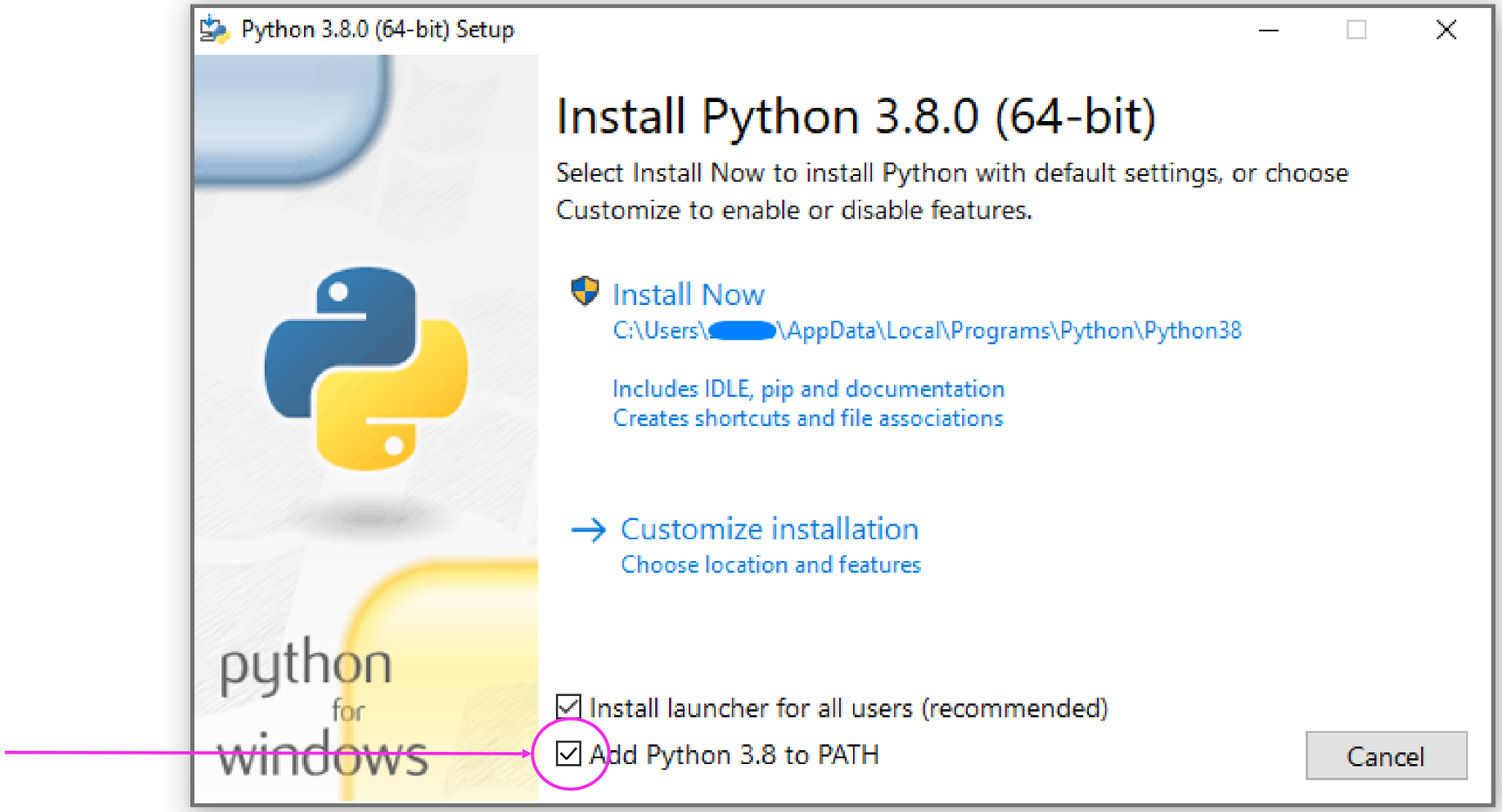
- Linux Distribuciones Debian-like:

```
sudo apt update
sudo apt install python3.12 python3-pip
```

- Linux Distribuciones Arch-like:

```
sudo pacman -S python python3-pip
```

Windows: importante

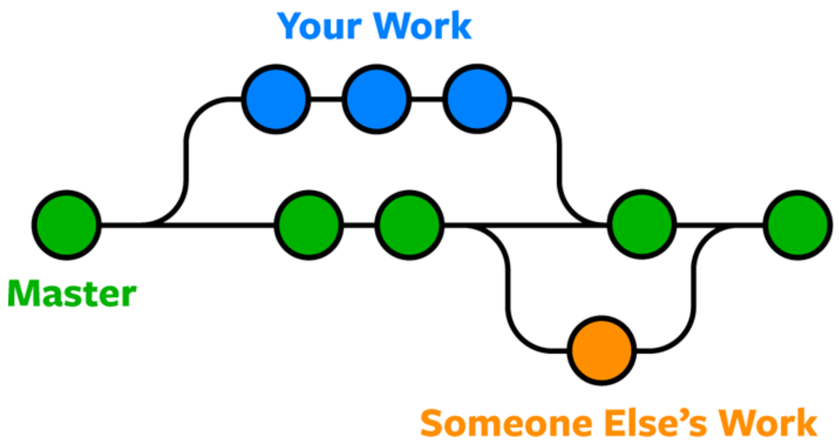


¿Qué es git ?

- Sistema de control de versiones, utilizado para gestionar y rastrear cambios en el código durante el desarrollo de software.

- En resumen, git es una herramienta que permite guardar y gestionar diferentes versiones de un proyecto.

- Git nos permite trabajar en un proyecto y mantener copias de seguridad a medida que hacemos cambios, lo cual es ideal para trabajos colaborativos.



¿Cómo saber si tengo git?

- Puedes ejecutar el siguiente comando:

```
git --version
```

- Si está instalado, debería aparecer algo del estilo:

```
git version 2.9.2
```

Cómo instalar git en Windows

- Puedes descargar el instalador desde aquí y seguir los pasos que te indican.



Cómo instalar git en MacOS

- Utilizaremos la terminal. Primero necesitamos instalar las herramientas de desarrollo de xcode:

```
xcode-select --install
```

- Luego instalamos Homebrew:

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

- Finalmente instalamos git:

```
brew install git
```

Cómo instalar git en Linux

- Distribuciones Debian-Like:

```
sudo apt install git
```

- Distribuciones Arch-Like:

```
sudo pacman -S git
```

Configurar git

- Para habilitar los colores en el terminal:

```
git config --global color.ui auto
```

- Configuración de credenciales:

```
git config --global user.name "NOMBRE APELLIDO"
git config --global user.email ejemplo@uc.cl
git config --global credential.helper store
```

Por favor...

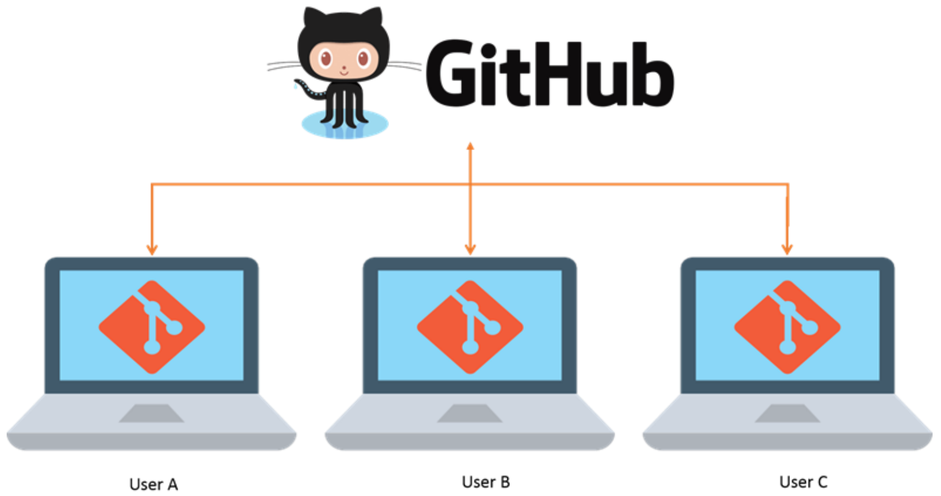
- Donde dice “NOMBRE APELLIDO” y ejemplo@uc.cl reemplázenlo por SU nombre y apellido, y su correo asociado a su cuenta de github.

- Por favor.



¿Qué es GitHub?

- Es una plataforma en la nube que utiliza git como sistema de control subyacente.
- Funciona como Google Drive, los archivos se almacenan en la nube y pueden ser accedidos desde cualquier lugar.
- A diferencia de Google Drive, GitHub está optimizado para la gestión de proyectos de software.
- Es el estándar de la industria.



Repositorios

- Son como una carpeta en la nube de Github donde se guarda todo lo relacionado al proyecto. En el curso, trabajaremos con tres repositorios distintos.

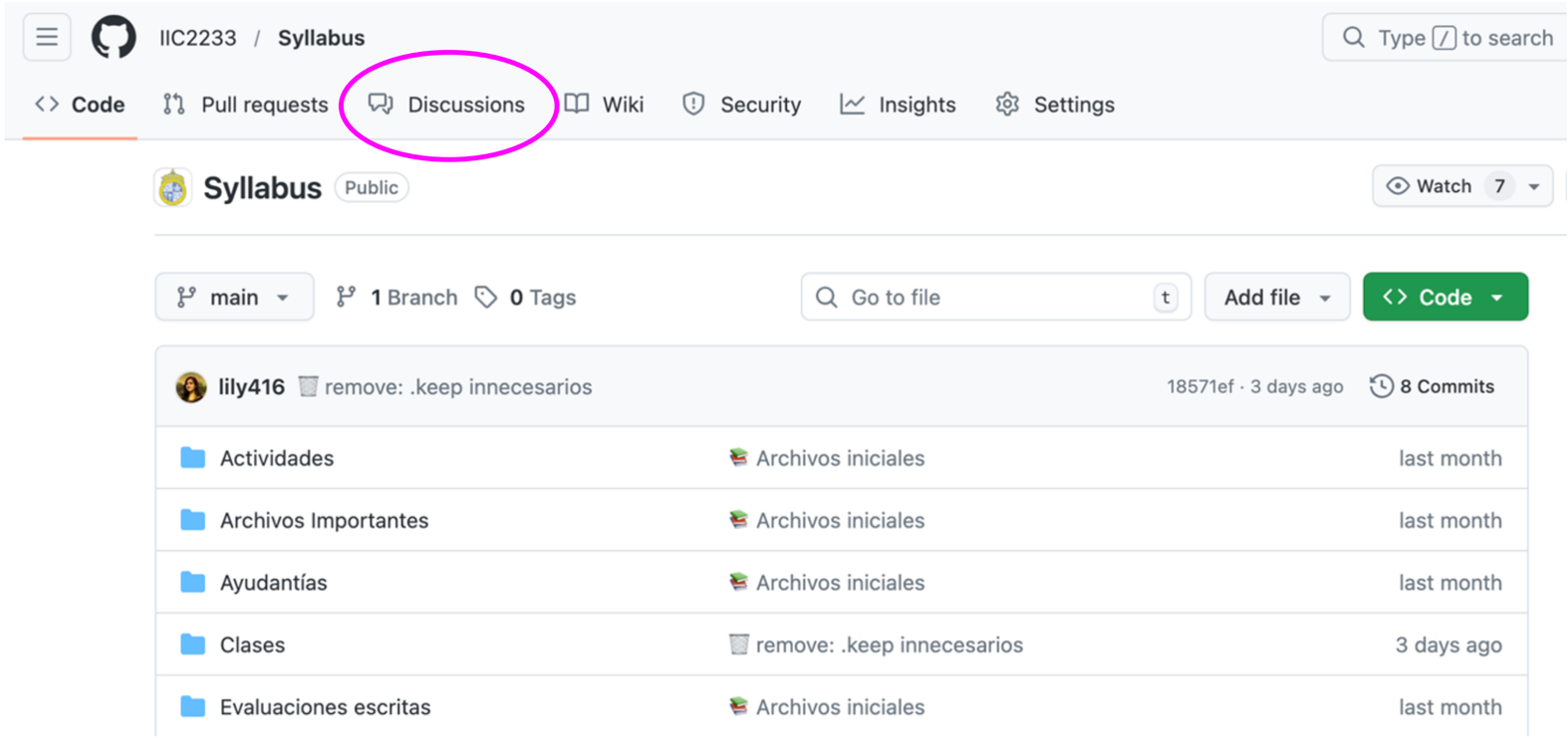
- Syllabus: aquí encontrarás las clases, ayudantías y tareas.

- Contenidos: aquí se subirán los contenidos del curso. Recuerda que el ramo tiene la metodología de Flipped Classroom, por lo que deberás estudiarlos por tu cuenta.

- Repositorio personal: aquí subirás tus tareas y actividades. Más información sobre esto será dada en la AC1.

Discussions

- Es el foro del curso, donde podrás realizar preguntas de instalación, contenidos, tareas, actividades, etcétera.



Reglamento de discussions

- Busca primero tu duda en Google (tip: si puedes, busca en inglés para mejores resultados).
- Revisa si alguien ya preguntó tu misma duda.
- Procura que el título de tu issue sea descriptivo.
- **No subir código de tareas.**
- No usar el foro como medio de reclamos.
- Política de descanso de ayudantes: no se responden discussions desde las 20:00 del sábado hasta las 8:00 del lunes.



Jupyter Notebook

- Son cuadernos interactivos que nos permiten mezclar texto y código de manera organizada. Los contenidos del curso y las ayudantías estarán en este formato.

- Para instalar jupyter usamos pip:

```
python3 -m pip install notebook
```

- También podemos abrir los cuadernos en VSCode.

- VSCode???

Interfaces de desarrollo (IDE’s)

- También conocidas como entornos de desarrollo o editores de código, son herramientas que usaremos para escribir, editar y gestionar nuestro código de manera más eficiente.

- Recomendamos usar VSCode, puedes instalarlo desde aquí.



Terminal

- Es una forma más directa de comunicarse con el computador. Comandos útiles:

- cd: moverse de directorio.
- dir/ls: mostrar contenido de un directorio.
- mkdir: crear un directorio.
- pwd: mostrar ruta absoluta del directorio.



Terminal: ejercicio

- Crea una nueva carpeta y muévete hacia ella.
- Crea un archivo de python vacío.
- Edítalo (con un VSCode o bloc de notas) para que imprima algo.
- Córrelo desde la terminal.
- Por último, entra a la consola de python y haz lo mismo.



Consejos para el curso :3

- Sigue la metodología, llega a las clases (e idealmente a las ayudantías) con los contenidos estudiados.

- Lee los enunciados de las tareas apenas salgan, te puede ayudar a hacerte una idea de lo que tienes que hacer antes de sentarte a programar.

- Dale tiempo a las tareas, si dejas todo a último minuto la puedes pasar mal :(.

- Antes de empezar a programar, planifica y diagrama. Divide la tarea en partes pequeñas y avanza desde ahí.

¡Éxito!

