

referente a git y GitHub, por ahora hemos visto como guardar versiones de nuestros proyectos y devolvernos a versiones anteriores y como conectar nuestro proyecto a un repositorio de GitHub para que podamos trabajar con demás personas, aquí un paso a paso:

para todos los siguientes comandos ya debemos haber navegado a la carpeta de nuestro proyecto

iniciar un repositorio desde cero:

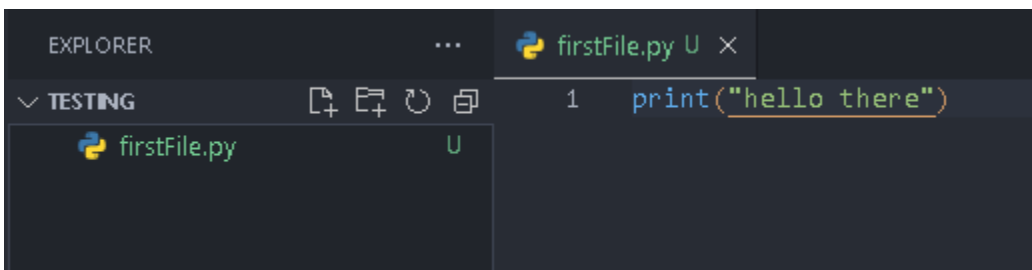
cuando queremos iniciar un nuevo proyecto y queremos manejarlo de manera que podamos tener distintas versiones de este, podemos hacer un repositorio de GitHub en dicho proyecto, ¿cómo? Necesitamos usar el siguiente comando:

```
git init
```

```
Administrator@PC-KEINER MINGW64 ~/Desktop/testing
$ git init
Initialized empty Git repository in C:/Users/Administrator/Desktop/testing/.git/
```

Podemos ver que efectivamente se creo un repositorio vacío en nuestra carpeta de proyecto, hay que tener en cuenta que no debe haber otro repositorio antes de esto.

Ya podemos ir haciendo cambios en nuestro proyecto, por ejemplo, añadamos un archivo:



Como podemos ver, automáticamente visual studio nos reconoce que el archivo no esta guardado en ninguna versión de nuestro repositorio (por el color verde del archivo y la letra U que viene de “untracked”)

También podemos ver esto desde nuestra consola con el comando:

```
git status
```

```
Administrator@PC-KEINER MINGW64 ~/Desktop/testing (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    firstFile.py

nothing added to commit but untracked files present (use "git add" to track)
```

Como vemos, aquí está en color rojo, pero nos indica lo mismo: ese archivo no lo tenemos guardado en ninguna versión, ahora, agreguémoslo, para esto, usamos el comando:

```
git add
```

Este comando lo podemos usar de 2 formas:

1. podemos poner el nombre de cada archivo para ir seleccionándolos 1 por uno, en este caso, podemos usar:

```
Administrator@PC-KEINER MINGW64 ~/Desktop/testing (master)
$ git add firstFile.py
```

2. si en vez del nombre de un archivo ponemos un punto, entonces se seleccionan todos los archivos para ser guardados.

Ahora que tenemos los archivos seleccionados, vamos a guardar una versión de nuestro proyecto, usamos el comando:

```
git commit
```

Como mencione antes, este comando nos sirve para guardar una versión de nuestro proyecto, el uso de este comando es el siguiente:

```
Administrator@PC-KEINER MINGW64 ~/Desktop/testing (master)
$ git commit -m "name of the version"
```

Siempre vamos a tener que usar este comando de la misma forma, lo único que cambiaremos será lo que va dentro de las comillas, que es el nombre de la versión que queremos agregar.

Perfecto, ya lo tenemos, si queremos guardar otros cambios que hagamos en una nueva versión simplemente tendremos que repetir los últimos 2 comandos, en realidad el comando de “git status” lo podemos usar cuando queramos para ver el estado en el que se encuentra nuestro repositorio, es opcional.

Por ahora estamos guardando los cambios localmente en la computadora, podemos vincular un proyecto a nuestro GitHub para poder colaborar con otras personas, pero por ahora acostúmbrate a usar estos comandos, hay que empezar por lo básico, ¿no se construye una pirámide desde arriba verdad?

En Python hemos hablado sobre los tipos de datos nativos principales, las conversiones entre estos, el uso y sintaxis del ciclo for en Python, a continuación, te presento unos ejercicios para que practiques:

1. Pide al usuario que ingrese un número entero y su nombre. Luego, imprime un saludo personalizado utilizando su nombre y el número ingresado multiplicado por 2. Por ejemplo, si el usuario ingresa "Juan" y "5", el programa debe imprimir "¡Hola, Juan! Tu número multiplicado por 2 es 10".
2. Crea una lista de números enteros y luego crea un programa que recorra la lista e imprima cada número. Ahora modifica el código para que dada la cantidad de números que se van a añadir, los números se le pidan al usuario y se vayan añadiendo a la lista (recuerda que para añadir un elemento a una lista usábamos la función `append()`)