

25 DE SEPTIEMBRE DE 2021

EMPEZAR CON PYTHON EN VSCODE

PABLO RAMOS DEL BUSTO

EMPEZAR CON PYTHON EN VSCODE

1. INTRODUCCIÓN

Este “manual” sirve como una ayuda básica y breve recordatorio para volver a programar en Python. Tan solo se anotarán unos pocos datos teóricos de la programación y se basará sobre todo en la instalación y configuración del entorno de programación.

Para hacer este manual se apoyó en:

- <https://code.visualstudio.com/docs/python/python-tutorial>
- <https://docs.python.org/3.7/using/windows.html>

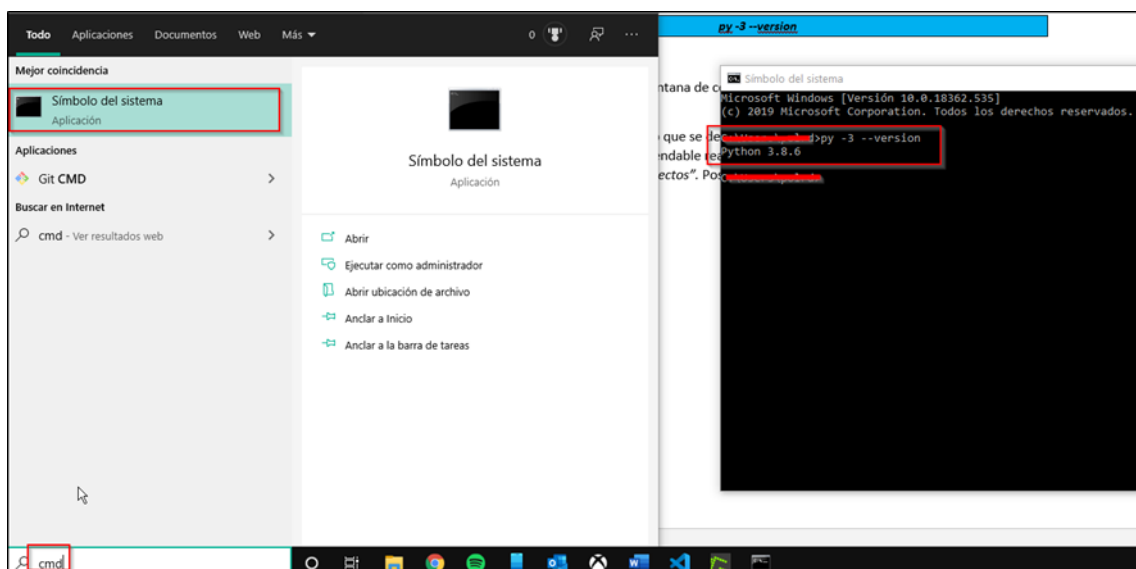
2. INSTALACIÓN PREVIA

Para programar Python en Visual Studio son necesarios tres requisitos.

1. Instalar [Microsoft Visual Studio](#)
2. Instalar [la extensión de Python](#)
3. Instalar [Python 3](#)

En Windows para verificar la instalación de Python se puede realizar a través de la ventana de comandos (buscar en Windows CMD). Escribiendo el comando siguiente:

```
py -3 --version
```



3. NUEVO WORKSPACE

Apoyándose de nuevo en la ventana de comandos se puede crear un espacio de trabajo para programar (**workspace**).

Para realizar esto es lo primero que se debe hacer es ir a la localización donde se quiere generar el proyecto. Es recomendable realizar esto en una carpeta localizada en una “parte alta”, por ejemplo, en “C:\Proyectos”. Posteriormente se copia la ruta y en la ventana de comandos se apunta hacia ella con el comando “cd” (**change directory**).

```
cd RUTA CARPETA
```

Una vez situado en esa posición, el siguiente paso es el de crear la carpeta donde se creará el proyecto e iniciarlo. Para crear la carpeta se usará el comando `mkdir` (***make directory***).

```
mkdir NOMBRE_PROYECTO
```

Se sitúa dentro de la carpeta recientemente creada, con `cd`. La tecla tabuladora sirve para navegar entre las distintas carpetas existentes en esa ruta.

```
cd NOMBRE_PROYECTO
```

Lanzamos el proyecto con **VSCode**.

```
code .
```

Cuando se lanza VSCode desde una carpeta, esta se convierte en nuestro espacio de trabajo **workspace**. VSCode almacenará todos los ajustes para este **workspace** en `.vscode/settings.json`. Estos ajustes estarán separados de los ajustes de usuario, almacenados de manera global.

3.1. SELECCIÓN DE PYTHON

A la hora de programar es necesario escoger el lenguaje en el que se va a realizar. De esta manera podremos ejecutar el código.

En este caso al realizarse en Python, lo que se hará será seleccionar el intérprete en el **workspace**, donde se esté trabajando. Para realizar esto se pulsará **Ctrl+Shift+P** y se escribe "Python: Select Interpreter".

Si se realiza esto fuera de una **workspace** se pondrá este intérprete por defecto en los ajustes globales.

3.2. CREAR UN ARCHIVO

Para crear un archivo Python es necesario utilizar la extensión `.py` de esta manera se le está diciendo que este archivo está programado en Python para usar el intérprete y poder evaluar el contenido.

3.3. INSTALAR Y USAR PAQUETES

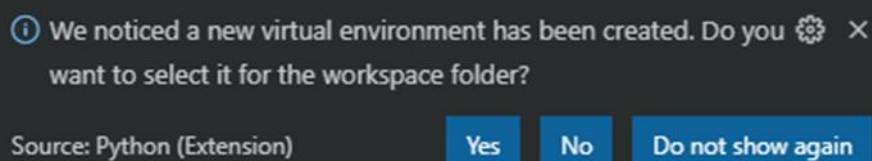
Normalmente los paquetes que se utilizarán se encuentran en el repositorio de [PyPI](#). Es importante trabajar en entornos virtuales e instalar estos paquetes en el entorno donde se esté trabajando. Los desarrolladores consideran una mala práctica instalar los paquetes en la configuración global (teoría sobre [enviroments](#)).

3.4. CREAR ENTORNO VIRTUAL

Para crear un entorno virtual en Windows, hay que escribir en la ventana de comandos (CMD) los siguientes comandos, valga la redundancia:

```
py -3 -m venv .venv
```

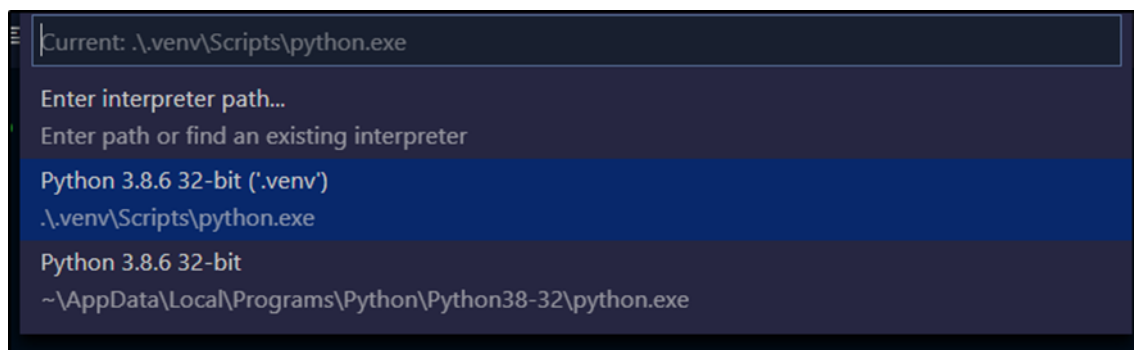
Cuando se crea un nuevo entorno virtual debería aparecer una ventana emergente en el VSCode que diga si se quiere establecer por defecto ese entorno para el **workspace** actual



Una vez creado el entorno se activa para que todo lo que se vaya a ejecutar o instalar sea dentro de este.

```
.venv\scripts\activate
```

Una vez creado el entorno, se selecciona usando el comando "Python: Select Interpreter" con **Ctrl+Shift+P** (*Command Palette*).



Se descargan los paquetes que se desee usar y se instalan, utilizando la ventana de comandos.

Se puede abrir la ventana de comandos directamente de VSCode con diferentes opciones, usando **Ctrl+ñ**, en la pestaña **View** y seleccionando la opción **Terminal** o con el **Command Palette** escribiendo "View: Toggle Integrated Terminal"

```
python -m pip install matplotlib
```

Cuando se está intentando ejecutar un entorno virtual es posible que se nos impida por política de **PowerShell**. Para evitar esto se requiere usar este código:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope Process
```

Una vez que se termina, se requiere desactivar el entorno virtual por medio del comando:

```
deactivate
```

4. INTELLISENSE

El **IntelliSense** es la función de autocompletado usada en el editor de código VSCode. Este se puede personalizar a través de estos dos enlaces:

- [Code analysis settings](#)
- [Autocomplete settings](#)

4.1. ACTIVAR INTELLISENSE PARA LOS PAQUETES

Para activar **IntelliSense** para paquetes que se encuentran instalados en otras localizaciones a la standard se tiene que añadir estas localizaciones al **python.autoComplete.extraPaths**. Por defecto esta colección se encontrará vacía.

```
"python.autoComplete.extraPaths": [
    "C:/Program Files (x86)/Google/google_appengine",
    "C:/Program Files
(x86)/Google/google_appengine/lib/flask-0.12"]
```