

3. Manejo de Python con Visual Studio Code

Visual Studio Code (VS Code) es un editor de código fuente gratuito, de código abierto y multiplataforma desarrollado por Microsoft. Es una herramienta muy popular entre desarrolladores, que se utiliza para escribir, editar y depurar código en diversos lenguajes de programación.

Instalación de Python

Para instalar un interprete de Python lo descargamos e instalamos desde su página web:

<https://python.org/>

Esto nos instalará una versión de Python "limpia" sin ningún paquete externo.

Para comprobar que tenemos Python bien instalado abrimos un terminal y escribimos:

```
python --version
```

Para Macos usamos "python3" en lugar de "python"

Para mostrar la lista de paquetes externos instalados escribimos:

```
pip list
```

Instalación de VS Code

Para instalar VS Code lo descargamos e instalamos desde su página web:

<https://code.visualstudio.com/>

Configuración de VS Code para Python

Para poder usar VS Code con Python y cuadernos de jupyter es necesario instalar algunas extensiones a VS Code y algunos paquetes externos a Python. La forma más sencilla de hacerlo es abriendo un jupyter notebook en VS Code e intentando ejecutarlo.

Si hacemos esto nos pedirá varios permisos para instalar extensiones y paquetes y finalmente tendremos 8 extensiones instaladas en VS code y un conjunto de paquetes necesarios para

ejecutar cuadernos de jupyter con VS Code.

Crear entornos virtuales Python

Un entorno virtual en Python es un directorio o carpeta aislada donde se puede elegir una versión específica de Python e instalar un conjunto de paquetes que sólo serán válidos para ese directorio.

La gestión de los entornos virtuales de Python se puede realizar desde la terminal usando comandos de Python, pero también se puede gestionar de forma gráfica usando una extensión para VS Code de Microsoft llamada "Python Environments".

A modo de ejemplo vamos a crear un nuevo directorio llamado "pruebas_python" y crearemos un entorno virtual en el que se materializará en un directorio ".venv"

Ejecución del módulo principal

La sentencia `if __name__ == '__main__':` en Python se utiliza para definir qué código se ejecutará cuando un archivo se ejecuta directamente como un programa principal, en lugar de ser importado como un módulo.

Facilita la reutilización de código en otros programas al evitar que se ejecuten partes no deseadas al importar el archivo. Y no utilizar variables globales, que no es una buena práctica.

¿Cómo funciona?:

`__name__` es una variable especial en Python que indica cómo se está ejecutando el archivo. `"__main__"` es el valor de `__name__` cuando el archivo se ejecuta directamente como el programa principal. Si el módulo fuera importado desde otro archivo la variable `__name__` contendría el nombre del módulo.

Ejemplo:

```
def saludar():  
    print("Hola, mundo!")  
  
if __name__ == '__main__':  
    saludar()
```

Ejemplo de módulo Python

Vamos a crear un primer archivo de pruebas para ver las capacidades de VS Code como entorno de desarrollo en Python. LLamaremos al archivo "pruebas.py" y copiamos en el el siguiente código:

```
import numpy as np
import matplotlib.pyplot as plt

def grafica():
    # Ejemplo de gráfica
    x = np.array([1, 2.5, 5, 10])
    y = np.random.rand(4)
    plt.figure()
    plt.plot(x,y, label='y (x)')

    plt.legend()
    plt.xlabel('x')
    plt.ylabel('y')
    plt.title('Figura de ejemplo')
    plt.xlim([0, 11])
    plt.ylim([0, 1])
    plt.grid()
    plt.show()

if __name__ == '__main__':
    grafica()
```

Para que este código funcione debemos instalar los paquetes numpy y matplotlib, que instalaremos únicamente en nuestro entorno virtual.

Ahora ya podemos ejecutar o depurar (debug) el archivo de python con VS Code.