

### LÍNEAS EN BLANCO

Dejar **dos líneas en blanco** entre **funciones** o **clases de nivel superior**.

Dejar **una línea en blanco** entre los **métodos de una clase**.

```
class MiClase:
    def metodo_uno(self):
        pass
    def metodo_dos(self):
        pass
def funcion():
    pass
```

### ESPACIOS EN BLANCO

**No añadir espacio** dentro de paréntesis, corchetes o llaves

```
lista = [1, 2, 3]
```

**No añadir espacios** antes o después de operadores como =, +, etc. (excepto en casos donde se usan para la alineación).

### COMPARACIONES

Usa `is` para comparar con `None`.

Usa `==` para comparar valores.

```
# Correcto:
if mi_variable is None:
    pass
```

```
# Incorrecto:
if mi_variable == None:
    pass
```

### CLÁUSULAS IF

Evitar hacer comparaciones innecesarias con `True`, `False`, o `None`.

```
# Correcto:
if mi_variable:
```

```
# Incorrecto:
if mi_variable == True:
```

### LONGITUD DE LAS LÍNEAS

Líneas de **79 caracteres** como máximo

### DOCSTRINGS

Las funciones, clases y métodos públicos deben tener un **docstring** que describa su propósito. Los **docstrings** deben usar comillas triples " " " .

```
def funcion(a, b):
    """
    Suma dos números.

    Args:
        a (int): Primer número.
        b (int): Segundo número.

    Returns:
        int: La suma de los dos.
    """
    return a + b
```

### ESPACIOS EN BLANCO

**No añadir espacio** dentro de **paréntesis, corchetes o llaves**

```
lista = [1, 2, 3]
```

**No añadir espacios** antes o después de **operadores** como =, +, etc. (excepto en casos donde se usan para la alineación).

```
x = 10 + 2
```

### COMENTARIOS

Los comentarios deben ser claros, completos y concisos. Usa comentarios **en línea** solo **si son necesarios**.

```
# Función que suma dos números.
def sumar(a, b):
    return a + b
```

### NOMBRES DE VARIABLES, FUNCIONES Y CLASES

Usa **snake\_case** para los nombres de variables y funciones.

```
mi_variable = 10
```

Usar **CamelCase** para los nombres de las clases.

```
class MiClase:
    pass
```

### IMPORTACIONES

Colocar las **importaciones** al inicio del archivo

Importa **una librería** por línea.

El **orden** de las importaciones debe ser:

1. Librerías **estándar** de Python
2. Librerías de **terceros**.
3. Librerías del **proyecto local**.

```
import os
import sys
```

```
from proyecto import mi_modulo
```

### EXCEPCIONES

Maneja las excepciones con bloques **try/except** específicos.

```
try:
    resultado = 10 / 0
except ZeroDivisionError:
    print("No dividi sib le")
```

### IDENTACIÓN

Usar **4 espacios** por cada nivel de indentación (no usar tabulaciones)

```
def funcion_ejemplo():
    if True:
        print(" ¡Co rre cto !")
```