

1. Qué es PEP8.

- PEP8 es una guía que contiene una serie de **recomendaciones** para escribir código Python de forma que resulte claro y legible.
- No son de obligado cumplimiento, pero son usadas por la comunidad Python.
- Algunos problemas que debemos evitar al escribir código son los siguientes:
 - Líneas demasiado largas.
 - Nombres de variables poco explicativos.
 - Código mal comentado.
 - Uso incorrecto de espacios y líneas en blanco.
 - Código mal indentado o sangrado.

2. Cómo hacer nuestro código más legible.

- **Líneas en blanco.** Hacen el código más claro y mejoran la legibilidad. Recomendaciones.
 - Las funciones y clases deben rodearse con dos líneas en blanco.
 - Separar los métodos de una clase con una línea en blanco.
 - Separar con una línea en blanco los bloques de código que realicen una función concreta.
- **Espacios en blanco. Recomendaciones.**
 - Rodear con un espacio los operadores de asignación.
 - `a = 1` # Correcto
 - `a=1` # Incorrecto.
 - Rodear también los operadores relacionales.
 - `if x == 5:` # Correcto.
 - No usar espacios en argumentos por defecto de funciones.
 - `def suma(a, b=10):` # Correcto.
 - No usar espacios dentro de los paréntesis ni de corchete.
 - `def suma(a)` # Incorrecto.
 - `lista = [1, 2, 3, 4]` # Incorrecto.
 - `lista = [1, 2, 3, 4]` # Correcto.
 - No usar espacios antes de comas en argumentos de funciones.
 - `print(x , y)` # Incorrecto.
 - No usar espacios entre nombres de listas y corchetes.
 - `lista [1]` # Incorrecto.
- **Tamaño de la línea.**
 - No debe exceder de 79 caracteres.
 - Usar `\` para continuar en una línea nueva las líneas que no se puedan romper.

- **Operaciones largas.**
 - Mejora la legibilidad dividiéndolas en varias líneas, con el operador al inicio de cada línea.
- **Identificadores.**
 - Nombre de variables, métodos y funciones. Utilizar snake case. Los nombres se escriben en minúsculas y si son compuestos, irán separados por guión bajo.
 - Ejemplos: `a`, `contador_facturas`, `calcula_area`.
 - Nombres de constantes. Su nombre se escribe en mayúsculas. Si el nombre es compuesto cada palabra irá separada por guión bajo.
 - Ejemplos: `PI`, `VALOR_3`.
 - Nombres de clases. Utilizar CamelCase para nombrar a las clases. Primera letra en mayúsculas, resto en minúsculas. Si la palabra es compuesta, la primera letra de cada palabra se escribe en mayúsculas.
 - Ejemplos: `Empleado`, `GestionAlmacen`.
 - Módulos. Igual que funciones.
 - Paquetes. El nombre se escribe en minúsculas, sea compuesto o no.
- **Orden en la importación de paquetes/módulos/funciones.**
 - Separar los import en varias líneas.
 - Si se importan varios elementos de una misma librería, irán en la misma línea.
 - `from modulo import funcion_1, funcion_2. # Correcto.`
 - Los import deben ubicarse al principio del archivo.
 - Orden.
 - Librerías estándar.
 - Librerías externas.
 - Librerías locales.
 - Evitar el uso de "import *".
- **Comentarios.**
 - Comienzan con #.
 - Deben ser frases completas.
 - Docstrings.
 - Son cadenas encerradas entre comillas simples triples o dobles triples y que se sitúan tras la declaración de una función, método o clase para describir qué hace.

3. Extensiones VSC para PEP8.

- Como se observa, el escribir código legible conlleva memorizar gran cantidad de reglas.
- Esto puede hacerse de forma automática gracias a ciertas extensiones de VSC.
- **Flake8.** Esta extensión de VSC contiene un "linter", es decir, una herramienta que corrige el código y lo adapta a las recomendaciones indicadas en PEP8. Puede

configurarse para que al guardar los cambios del documento se apliquen las correcciones necesarias.