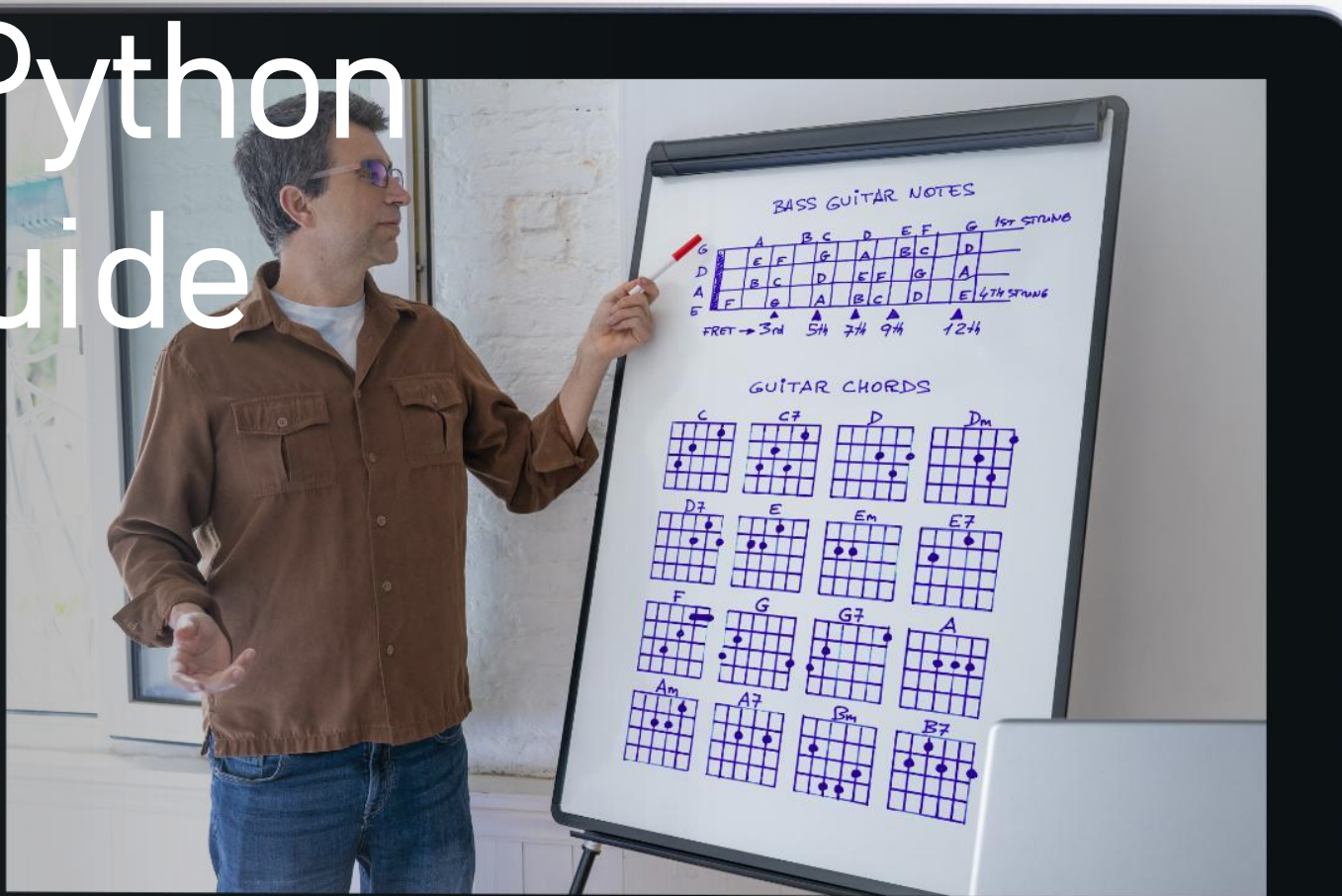


# PEP 8 Python Style Guide



MEJORES PRÁCTICAS PARA  
PROGRAMAR EN PYTHON



# Presentation Overview

## PEP 8

PEP 8 es una guía de estilo para la escritura de código Python que describe las mejores prácticas recomendadas para mejorar la legibilidad y mantenibilidad del código.

## Mejores prácticas para nombrar e identificar en Python

Las mejores prácticas para nombrar e identificar en Python incluyen convenciones de nomenclatura, recomendaciones de identificadores y estilos de nomenclatura.

## Recapitulación y recursos

En resumen, PEP 8 es una guía de estilo importante para mejorar la legibilidad y mantenibilidad del código Python. También aprenderemos sobre las mejores prácticas para nombrar e identificar en Python. Para obtener más información, puede consultar la documentación oficial.



# Introducción a la guía de estilo PEP 8 de Python

## Directrices para un código limpio y legible

PEP 8 proporciona directrices para escribir un código Python limpio y legible, lo que hace que sea más fácil para otros desarrolladores leer y entender tu código.

## Estándar de codificación para desarrolladores de Python

PEP 8 proporciona un estándar de codificación para desarrolladores de Python, asegurando que el código sea consistente y fácil de entender.

## Mejora de la calidad del código

PEP 8 ayuda a mejorar la calidad del código al establecer directrices para la escritura de un código claro y consistente, lo que facilita la detección y corrección de errores.

# ¿Qué es PEP 8?

PEP 8 es un conjunto de directrices para escribir código Python que es fácil de leer y mantener. Incluye directrices sobre espacios en blanco, convenciones de nomenclatura, estructura de código y más.

”



# ¿Cómo ayuda?

## Mejora la calidad del código

PEP 8 ayuda a mejorar la calidad del código al proporcionar un conjunto de pautas que ayudan a garantizar que el código sea más limpio, claro y fácil de entender.

## Consistencia

PEP 8 ayuda a garantizar la consistencia en todo el código de un proyecto, lo que a su vez hace que sea más fácil trabajar en equipo y mantener el código a largo plazo.

## Facilidad de comprensión

Con PEP 8, el código se hace más fácil de leer y comprender tanto para el programador que lo escribió como para otros que lo revisan o actualizan en el futuro.

# PEP 8 guía de estilo general

## Indentación

PEP 8 proporciona directrices específicas para la indentación de código en Python. La indentación adecuada mejora la legibilidad y facilita la comprensión del flujo de control de su programa.

## Convenciones de nomenclatura

PEP 8 establece convenciones de nomenclatura para variables, funciones y clases en Python para mejorar la legibilidad del código y hacer que su código sea más consistente con otras bibliotecas de Python.

## Espacios en blanco

PEP 8 establece pautas para el uso adecuado de espacios en blanco en su código Python. El uso adecuado de espacios en blanco ayuda a mejorar la legibilidad y a hacer que su código sea más fácil de entender.

# Guía de Estilo PEP 8 para Python



## Indentación en Python

Utilizar 4 espacios por nivel de indentación



## Longitud de Líneas

Limitar a 79 caracteres por línea



## Estructura del Código

Separar funciones y clases con 2 líneas en blanco



## Codificación de Archivos

Preferir ASCII o UTF-8



## Importaciones

Una sentencia 'import' por módulo  
Ordenarlas al inicio del documento



## Formato y Nombres



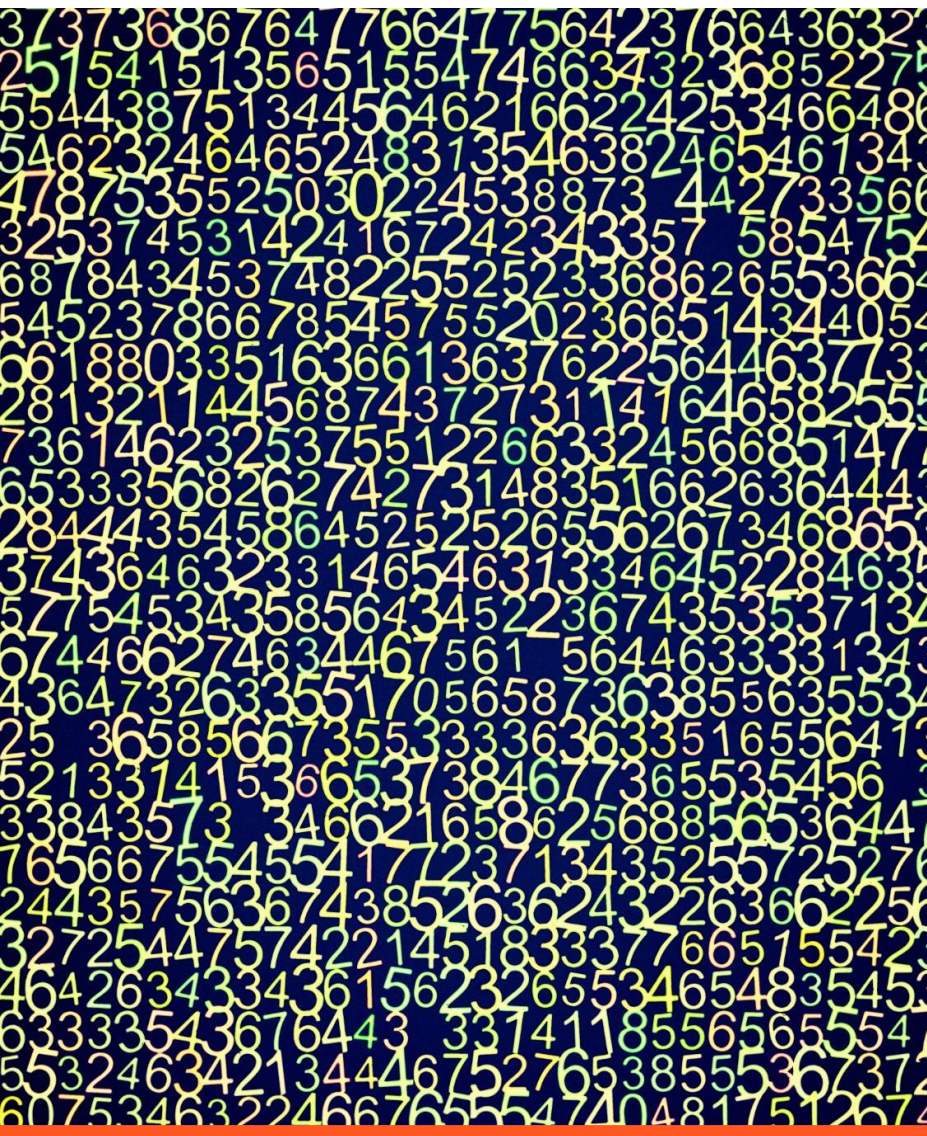


# Mejores prácticas para nombrar identificadores en Python

---

Tipo	Convención de nombres	Ejemplos
Funciones	Usar solo minúsculas y separar palabras con _ para mejorar la lectura.	function, my_function
Variables	Usar letras individuales, palabra o palabras en minúsculas y separar las palabras con _.	x, var, my_variable
Clases	Iniciar cada palabra con mayúsculas (CamelCase). No separar las palabras con _.	Model, MyClass
Métodos de clases	Usar solo minúsculas y separar palabras con _ para mejorar la lectura.	class_method, method
Módulos	Usar solo una o más palabras cortas en minúsculas y separar palabras con _ para mejorar la lectura.	module.py, my_module.py
Paquetes	Usar solo una o más palabras cortas en minúsculas. No separar las palabras con _.	package, mypackage





# Recomendaciones para identificadores

## Nombres Descriptivos

El uso de nombres descriptivos para los identificadores es una buena práctica de programación que mejora la legibilidad del código. Los nombres deben ser cortos pero significativos, y deben reflejar el propósito del código.

## Variables con sustantivos singulares

Es recomendable utilizar sustantivos singulares para los nombres de las variables, ya que esto ayuda a evitar confusiones y hace que el código sea más legible.

## Colecciones con sustantivos plurales

Es recomendable utilizar sustantivos plurales para los nombres de las colecciones, ya que esto ayuda a indicar que el identificador hace referencia a un conjunto de elementos.





# Conclusion

PEP 8 proporciona directrices para escribir código Python limpio, legible y fácil de mantener. Seguir estas directrices puede ayudar a escribir código que sea más fácil de entender, más coherente y de mayor calidad.