ESTÁNDARES DE GITHUB

Equipo 5
Jessica Treviño
Fernando Sánchez
Rodrigo Murillo
Eduardo Loya
Victoria Domínguez

COMMITS PEQUEÑOS

Hacer commits pequeños y frecuentes hace que sea más fácil ver el historial de commits y encontrar un commit específico que necesites. Es recomendado usar el comando "git add -a" con moderación, y hacer commits más frecuentemente.

DARLE COMMIT A CÓDIGO COMPLETO Y PROBADO

Es recomendado trabajar con pequeñas tareas y asegurar que cada una esté completa. Además, se sugiere que el código se pruebe antes del commit.

ESCRIBIR BUENOS MENSAJES DE COMMIT

Escribir mensajes de comit descriptivos mantiene el repositorio organizado y hace que sea fácil navegar por el log. El mensaje del commit debe ser corto, en tiempo presente, y la razón de cambio debe estar explícita.

USO DE ESTILO IMPERATIVO EN LA LÍNEA DE ASUNTO

Usar el modo imperativo en lugar del estilo de tiempo pasado. Escribir en tiempo presente le dice a alguien lo que hará el commit, en lugar de lo que hiciste.

USAR EDITOR DE CÓDIGO PARA MENSAJES LARGOS DE COMMIT

Es mucho más fácil usar un editor de código cuando tienes que escribir un cuerpo. La línea de asunto y el cuerpo deben estar separados por una línea en blanco.

APRENDER A USAR GIT CON LA LÍNEA DE COMANDOS O TERMINAL PRIMERO

A la mayoría de los desarrolladores les gusta la interfaz de línea de comando y sus herramientas, y si no sabes cómo usar Git dentro de la línea de comando, no te sentirás profesional.

LEER EL LIBRO "PRO GIT"

Puedes usar este libro como un recurso para aprender diversos comandos y herramientas que pueden mejorar tu flujo de trabajo.

Estándares de Python

Equipo 5

Introducción a Python

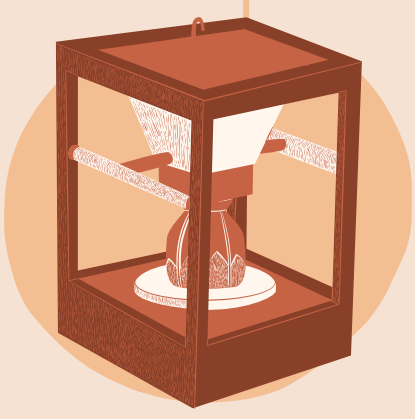
Python es un lenguaje de programación de propósito general de alto nivel interpretado. Su filosofía de diseño enfatiza la legibilidad del código con el uso de una sangría significativa. Sus construcciones de lenguaje, así como su enfoque orientado a objetos, tienen como objetivo ayudar a los programadores a escribir código claro y lógico para proyectos de pequeña y gran escala.

Sangría de 4 espacios

Las líneas de continuación deben alinear los elementos envueltos ya sea verticalmente usando la línea implícita de Python que se une entre paréntesis, corchetes y llaves, o usando una sangría francesa.

¿Espacios o TAB?

Los espacios son el método de sangría preferido. Los tabs deben usarse únicamente para mantener la coherencia con el código que ya está sangrado con tabs. Python no permite mezclar tabulaciones y espacios para la sangría.

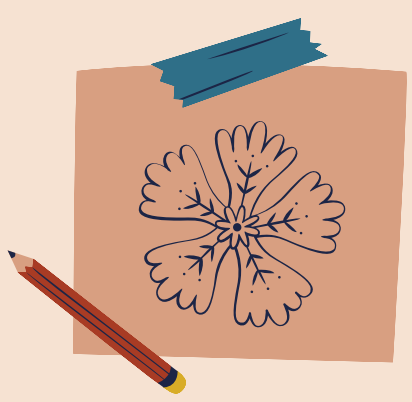


Longitud de línea

La longitud es un máximo de 79 caracteres. Para bloques de texto largos fluidos con menos restricciones estructurales (docstrings o comentarios), la longitud de la línea debe limitarse a 72 caracteres.

String quotes

En Python, las cadenas entre comillas simples y las cadenas entre comillas dobles son iguales. Sin embargo, cuando una cadena contiene caracteres de comillas simples o dobles, hay que utilizar la otra para evitar barras invertidas en la cadena.

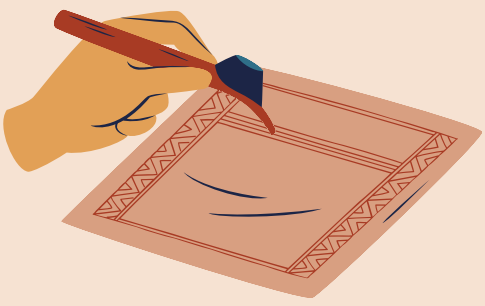


Comentarios

Hay que mantener los comentarios actualizados cuando el código cambie. Deben ser oraciones completas. Los comentarios en bloque generalmente consisten en uno o más párrafos construidos a partir de oraciones completas, y cada oración termina en un punto.

Convención de nombramiento

Ejemplos:
b, B, lowercase, lower_case_underscores, UPPERCASE, UPPER_CASE_UNDERSCORES, CamelCase, mixedCase, Capitalized_Words_Underscores



Nombre de clases

Hay que mantener los comentarios actualizados cuando el código cambie. Deben ser oraciones completas. Los comentarios en bloque generalmente consisten en uno o más párrafos construidos a partir de oraciones completas, y cada oración termina en un punto.

Variables y funciones

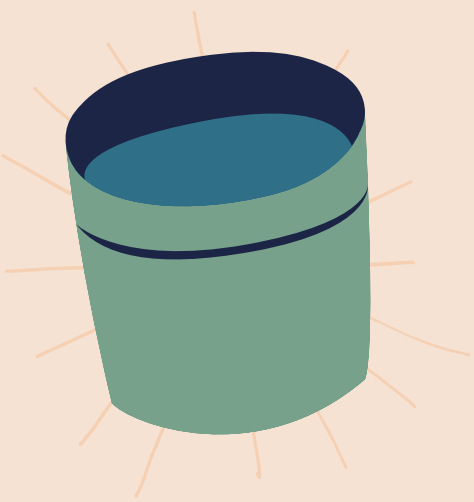
Los nombres de las funciones deben estar en minúsculas, con palabras separadas por guiones bajos según sea necesario para mejorar la legibilidad. Los nombres de las variables siguen la misma convención que los nombres de las funciones. MixedCase está permitido solo en contextos donde ese ya es el estilo predominante (por ejemplo, threading.py), para mantener la compatibilidad con versiones anteriores.

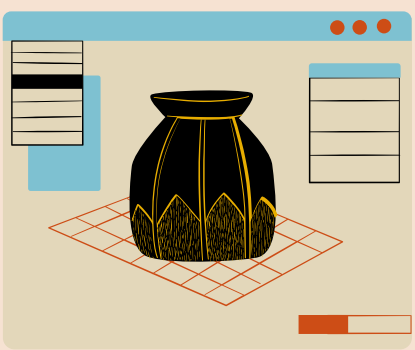
Funciones y argumentos

Usar siempre "self" para el primer argumento de los métodos de instancia. Usar siempre "cls" para el primer argumento de los métodos de clase.

Métodos y variables de instancia

Nomenclatura de funciones: minúsculas con palabras separadas por guiones bajos según sea necesario para mejorar la legibilidad. Usar un guión bajo inicial solo para métodos no públicos y variables de instancia. Para evitar conflictos de nombres con subclases, usar dos guiones bajos iniciales para invocar las reglas de modificación de nombres de Python.





Recomendaciones

El código debe escribirse de una manera que no perjudique a otras implementaciones de Python (PyPy, Jython, IronPython, Cython, Psyco, etc.).