



# O GUIA DE ESTILOS DE PROGRAMAÇÃO PEP8 DO PYTHON

Vamos aprender nesse ebook sobre o Guia de Estilo de Programação PEP8 do Python e seus principais pontos.

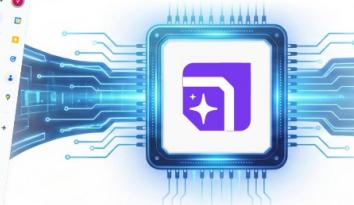
Gere ebooks como este com



Ebookr

em <https://ebookr.ai>

# Crie ebooks profissionais incríveis em minutos com IA



Chega de formatar texto no Google Docs, Word ou ferramentas que só te fazem perder tempo...

E deixe que nossa IA faça o trabalho pesado!



Capas gerados por IA



Infográficos feitos por IA



Edite em Markdown em Tempo Real



Adicione Banners Promocionais

TESTE AGORA



PRIMEIRO CAPÍTULO 100% GRÁTIS

Salve salve Pythonista!

Na programação, além de termos a preocupação em criar um código funcional e eficiente, também é importante manter um padrão de escrita que facilite a leitura e a manutenção do código ao longo do tempo.

Dessa forma, o código se torna mais fácil de entender e mais amigável tanto para o próprio desenvolvedor quanto para outros colaboradores que possam trabalhar no mesmo projeto.

O Python é uma linguagem conhecida por ter uma sintaxe clara e legível.

Isso se deve em parte ao PEP8, que é o guia de estilos de programação recomendado para Python.

Neste artigo, vamos explorar os principais pontos do PEP8 e entender como utilizá-lo em nossos projetos.

## Por que seguir o PEP8?

Seguir o PEP8 traz diversas vantagens para um desenvolvedor Python.

Primeiramente, como mencionado anteriormente, o código se torna mais fácil de entender e de dar manutenção.

Isso porque o PEP8 define regras claras para nomes de variáveis, indentação e organização de código.

Outra vantagem é a padronização do código: quando todos os colaboradores de um projeto seguem o mesmo guia de estilos, a leitura e o entendimento do código se tornam **mais rápidos e fluidos**.

Além disso, a consistência do código facilita a identificação de erros e a realização de revisões.

Seguir o PEP8 é uma prática **muito valorizada** pela comunidade Python.

Ao compartilhar seu código com outros desenvolvedores ou ao contribuir para projetos open-source, seguir o guia de estilos de programação do Python demonstra profissionalismo e respeito pela comunidade.

## Os principais pontos do PEP8

### Indentação e espaçamento

Um dos pilares do PEP8 é a padronização da indentação e espaçamento do código.

O uso correto de espaços em branco ajuda a tornar o código **mais legível**.

As principais recomendações estão relacionadas à quantidade de espaços para fazer a indentação.

O PEP8 recomenda o uso de 4 espaços para cada nível de indentação.

**Não é recomendado o uso de tabulações** para realizar a indentação.

Outro ponto importante é a separação de operadores e argumentos por espaços.

Por exemplo, em uma chamada de função, deve-se deixar um espaço após cada vírgula que separa os argumentos.

Portanto, o seguinte código não segue o PEP8:

```
def soma(num_1: int, num_2: int) -> int:  
    return num_1 + num_2  
  
resultado = soma(1, 2)  
  
# O correto seria  
resultado = soma(1, 2)  
#  
#  
#           ^ espaço
```

## Nomes de variáveis

A escolha dos nomes de variáveis também é uma parte importante do PEP8.

O uso de nomes descritivos ajuda a entender a função e propósito de cada variável no código.

De acordo com o guia de estilos, é recomendado utilizar letras minúsculas e separar palavras por sublinhados (snake\_case) para nomes de variáveis e funções.

Por exemplo:

```
nome_completo = "John Doe"  
idade = 30
```

Para nomes de Classes, por convenção, utiliza-se a notação **CamelCase**, ou seja, as palavras são iniciadas com letras maiúsculas e não há sublinhados.

Por exemplo:

```
class Pessoa:  
    def __init__(self, nome, idade):  
        self.nome = nome  
        self.idade = idade
```

Portanto, não faça:

```
class Pessoa_Desenvolvedora:  
    def __init__(self, nome, idade):  
        self.nome = nome  
        self.idade = idade
```

 Estou desenvolvendo o **Ebookr.ai**, uma plataforma que transforma suas ideias em ebooks profissionais usando IA – com geração de capa, infográficos e exportação em PDF. Te convido a conhecer!

## Crie Ebooks profissionais incríveis em minutos com IA

Chega de formatar texto no Google Docs, Word ou ferramentas que só te fazem perder tempo...

... e deixe que nossa IA faça o trabalho pesado!

**TESTE AGORA! PRIMEIRO CAPÍTULO 100% GRÁTIS** 

 Capas gerados por IA

 Adicione Banners Promocionais

 Edite em Markdown em Tempo Real

 Infográficos feitos por IA

GERE EBOOKS COMPLETOS COM O  **Ebookr**  
ACESSE EBOOKR.AI  PRIMEIRO CAPÍTULO 100% GRÁTIS

4

## Comprimento das linhas

Embora não haja um limite rígido para o comprimento das linhas, o PEP8 recomenda que as linhas não ultrapassem 79 caracteres.

Para linhas mais longas, é indicado quebrá-las em várias linhas, respeitando a indentação adequada.

Veja um exemplo:

```
resultado_muito_longo = variavel_a + variavel_b + variavel_c + \
                        variavel_d + variavel_e
```

## Comentários

Os comentários são uma parte importante do código, pois ajudam a explicar a lógica por trás de cada trecho.

No entanto, é importante utilizar os comentários de forma consciente e evitar excessos.

Segundo o PEP8, os comentários devem ser claros e explicativos, sendo escritos em linhas separadas e alinhados à mesma indentação do código que estão comentando.

## Outros pontos importantes

Além dos pontos mencionados acima, o PEP8 também aborda outras questões relevantes, como:

- Não utilizar linhas em branco no final do arquivo.
- Importar módulos em linhas separadas.

- Evitar o uso de parênteses adjacentes sem a devida necessidade.

## Conclusão

Em resumo, seguir o guia de estilos de programação do Python, conhecido como PEP8, traz diversas vantagens para um desenvolvedor Python.

Além de tornar o código mais legível e fácil de dar manutenção, o PEP8 padroniza a escrita do código, facilitando a colaboração em projetos e demonstrando profissionalismo para a comunidade.

Ao utilizar nomes de variáveis descritivos, indentação correta, espaçamento adequado e seguir as demais recomendações do PEP8, seu código Python se tornará mais coerente, legível e de fácil manutenção.

É importante mencionar que existe uma ferramenta chamada **Pylint**, que verifica automaticamente se o código está seguindo as diretrizes do PEP8.

Utilizar essa ferramenta em seus projetos é uma ótima maneira de garantir a aderência ao guia de estilos.

Seguir o PEP8 é uma prática que todo desenvolvedor Python deveria adotar, pois contribui diretamente para a qualidade e a legibilidade do código fonte.

Mantenha-se atualizado sobre as últimas diretrizes do PEP8 e torne-se um programador Python mais eficiente e profissional.

Espero que tenha curtido esse Post!

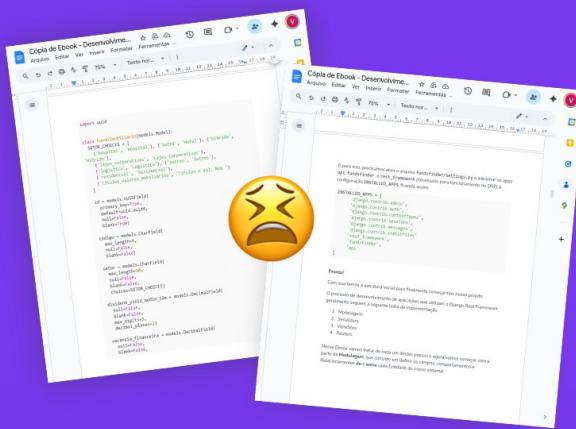
Nos vemos no próximo 

Não se esqueça de conferir!



Ebookr

# Crie Ebooks profissionais em minutos com IA



Chega de formatar código no Google Docs ou Word



**Arquitetura de Software Moderna**

```
import python
import python

class Arquitetura_de_Software_Moderna:
    ...
    def share(self):
        pass
    ...
    return "Arquitetura de Net", "arquitetura"
}

def __init__(self):
    if not self.share():
        raise ValueError("A classe deve implementar o método share()")
    else:
        self.__share()

# Exemplo de uso da classe
arquitetura = Arquitetura_de_Software_Moderna()
print(arquitetura.share())

```

**AI-generated system**

A arquitetura com propósito é a mesma software armazena de fusões modernas. Seus sistemas integrados consistem na interface entre os sistemas externos. Chama-se a arquitetura de gerenciamento de sistemas gerados.

```
graph TD
    Motor[Motor gerador] --> Agente[Agente Especializado]
    Agente --> Sistema[Sistema Gerador]
    Sistema --> Orchestrador[Orchestrador]
    Orchestrador --> Motor

```

**Clean layout**

Garantimos que o layout é limpo e organizado para facilitar a leitura e compreensão dos dados.



Capas gerados por IA

Infográficos feitos por IA

Adicione Banners Promocionais

Deixe que nossa IA faça o trabalho pesado

Edite em Markdown em Tempo Real

**TESTE AGORA**



PRIMEIRO CAPÍTULO 100% GRÁTIS