

Reading Python Programming

PEP-8 e Zen of Python

9 de julho de 2024

Agenda da Sessão

- 01. PEP-8
- 02. Zen of Python
- 03. Live Demo
- 04. Exercícios de Algoritmia

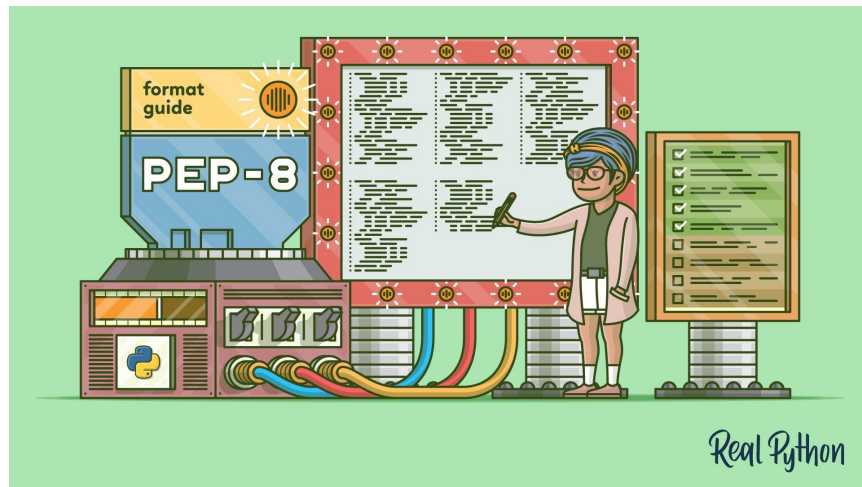
01

PEP-8

PEP-8

Python Enhancement Proposal

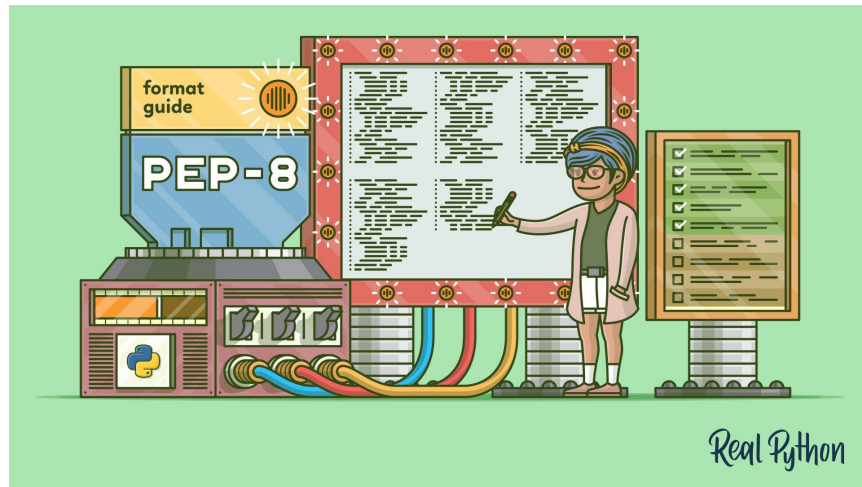
- PEP-8 é o guia de estilo oficial para código Python
- Criado em 2001, continua a definir o standard da programação em Python
- Fornece diretrizes para escrever código Python legível e consistente, o que é essencial para facilitar a colaboração e a manutenção de projetos de código



PEP-8

Python Enhancement Proposal

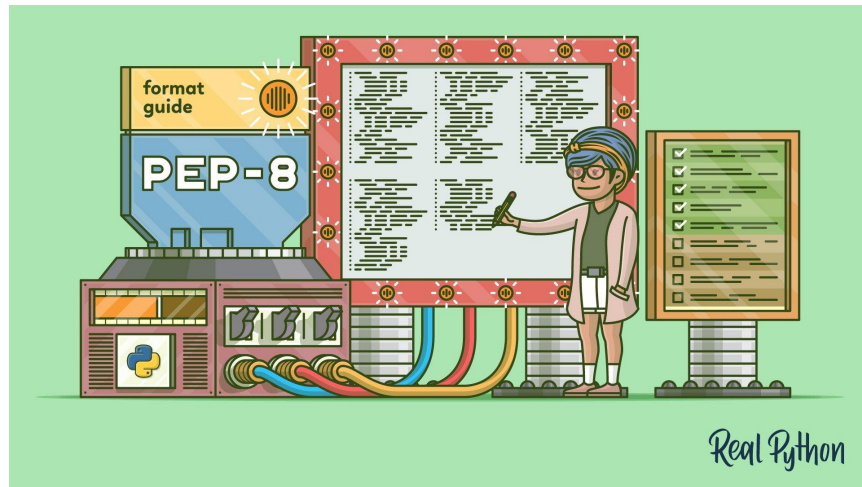
- **Indentação** O PEP-8 recomenda usar quatro espaços por nível de indentação. Não deve ser usado tabulação, exceto se estiver num projeto que já use essa convenção
- **Espaços em Branco** Usar espaços em branco de forma consistente. Isso inclui espaços ao redor de operadores e após vírgulas
- **Comprimento das Linhas** O comprimento máximo recomendado para uma linha de código é 79 caracteres. Linhas longas devem ser quebradas para facilitar a leitura



PEP-8

Python Enhancement Proposal

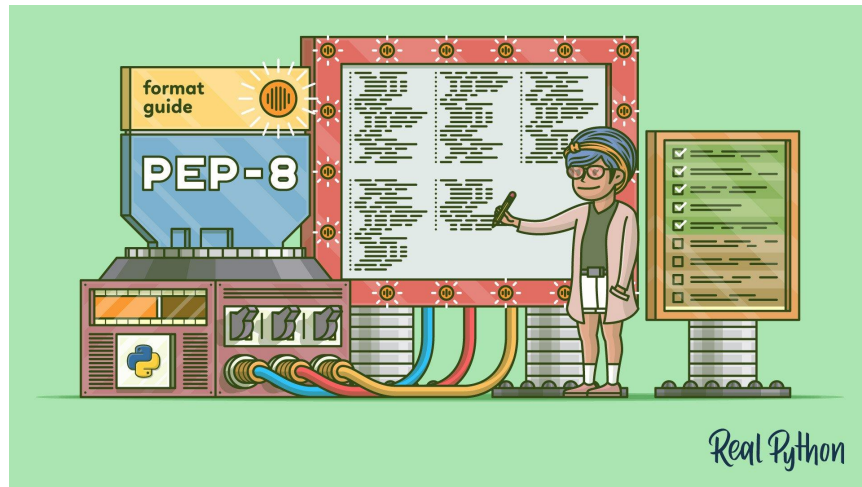
- **Imports** Os imports devem ser organizados em grupos separados e devem seguir uma ordem específica (ex: primeiro os imports padrão, depois os de terceiros e, por fim, os imports locais)
- **Nomes de Variáveis e Funções** Deve se usar nomes descritivos e significativos para variáveis, funções e classes.
Nomeadamente o estilo snake_case para nomes de variáveis e funções, o estilo UPPERCASE para variáveis constantes, e o estilo CamelCase para nomes de classes



PEP-8

Python Enhancement Proposal

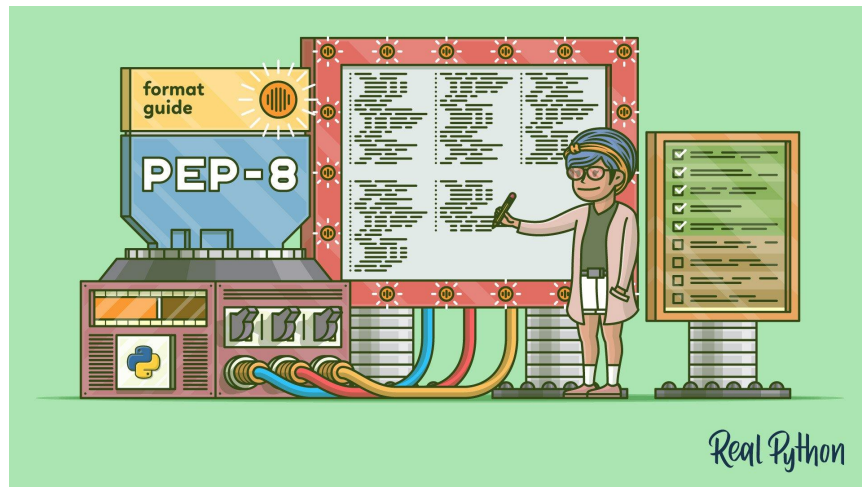
- **Docstrings** Usar docstrings para documentar funções, classes e módulos. Isso ajuda outros programadores a entenderem a finalidade e como o código funciona
- **Comentários** Usar comentários para explicar partes complexas do código ou para fornecer contexto quando necessário. Evitar comentários óbvios que não acrescentam valor



PEP-8

Python Enhancement Proposal

- **Consistência** A consistência é o mais importante. Não se deve misturar padrões, mesmo que coincidentes com o guia de estilos. É importante manter a formatação do projeto



02

Zen of Python

Zen Of Python

- Manifesto elaborado em 2004
- Pretende definir o mindset ao programador de Python, mas é útil e relevante para um programador de qualquer linguagem
- Glorifica a simplificação e leitura do código, realça a importância de tornarmos o código perceptível para qualquer programador – e que os standards são importantíssimos para atingir isso



Zen of Python

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one-- and preferably only one --obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than **right** now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea -- let's do more of those!

03

Live Demo

Pedra-Papel-Tesoura!

vs Computer

Objetivo

- 1 - O programa solicita ao utilizador que escolha uma das opções: Pedra, Papel ou Tesoura.
- 2 - O computador escolherá aleatoriamente uma das mesmas três opções.
- 3 - O programa determina o vencedor com base nas regras do jogo:
 - Pedra vence Tesoura.
 - Tesoura vence Papel.
 - Papel vence Pedra.

Se ambos escolherem a mesma opção, o resultado é um empate.

- 4 - Repetir até Utilizador ou Computador vencerem 3 vezes.



03

Exercícios de Algoritmia

miscellaneous.py

Proposta

Executar os 15 exercícios propostos no ficheiro miscellaneous.py.

Objetivo

Praticar os conceitos dados anteriormente, consoante o tempo disponível.



Outras plataformas de exercícios/apoio

Exercícios

<https://codingbat.com/>

<https://www.codewars.com/>

Material Apoio

<https://www.w3schools.com/>

<https://realpython.com/>

Outros

ChatGPT ou similar para pedir exercícios ou explicações – exemplo
pedir um exercício por dia



Formação e
Certificação
Técnica que
potenciam
Profissionais e
Organizações

Lisboa

Edifício Mirage – Entrecampos
Rua Dr. Eduardo Neves, 3
1050-077 Lisboa

[ver google maps](#) ↗

Tel +351 217 824 100
Email info@training.rumos.pt

Siga-nos



Porto

Edifício Mirage – Entrecampos
Rua Dr. Eduardo Neves, 3
1050-077 Lisboa

[ver google maps](#) ↗

Tel +351 222 006 551
Email info@training.rumos.pt