



Oceano do Python

por Edinei Cavalcanti

Vamos conversar?



Edinei Cavalcanti

neiesc

Programador há mais de 15 anos, graduado em ciência da computação na UNIP de Rio Preto em 2015, focado em tecnologia e autoconhecimento, professor pelo Estado de São Paulo para o curso técnico em informática do novo ensino médio, um eterno aprendiz...

<https://edinei.social>

<https://edinei.dev>

Vamos jogar?



<https://partici.fi/43963606>



Breve História do Python



Início

- Criado por **Guido van Rossum** em 1989 
- Objetivo: linguagem simples, legível e poderosa 
- Inspirado em linguagens como **ABC**, **Modula-3** e **C** 

⚙️ Evolução

- Python 1.0 lançado em 1991 🎉
- Python 2.0 em 2000: list comprehensions e melhor coleta de lixo 💚
- Python 3.0 em 2008: mudanças incompatíveis para modernização 🔪

🌟 Popularidade

- Simplicidade e legibilidade atraem iniciantes e profissionais 💁‍♀️ 💁‍♂️
- Usado em web, ciência de dados, IA, automação, e muito mais 🌐 🤖 📊
- Comunidade ativa e extensa biblioteca padrão 📚

🔥 Python Hoje

- Versão estável: 3.x 
- Muito utilizado globalmente 
- Crescimento contínuo em ciência, educação e indústria 



O Zen do Python

por Tim Peters

- ✨ Bonito é melhor que feio
- 🔍 Explícito é melhor que implícito
- 🧠 Simples é melhor que complexo
- ✖ Complexo é melhor que complicado



Raso é melhor que profundo



Esparsos é melhor que denso



Legibilidade conta

- 🚫 Casos especiais não são especiais o bastante para quebrar as regras
- ⚖️ Embora a praticidade supere a pureza

✗ Erros nunca devem passar silenciosamente

 **A menos que sejam explicitamente silenciados**

? Diante da ambiguidade, recuse a tentação de adivinhar

 Deve haver uma – e preferencialmente apenas uma – maneira óbvia de fazer isso

 Embora essa maneira possa não ser óbvia à primeira vista, a menos que você seja holandês



Agora é melhor que nunca

💡 *Embora nunca frequentemente seja melhor que agora mesmo*

- ⚠️ Se a implementação é difícil de explicar, é uma má ideia
- 💡 Se a implementação é fácil de explicar, pode ser uma boa ideia



Namespaces são uma grande ideia – vamos fazer mais desses!



Zen concluído

Escreva código bonito, legível e Pythonic.



"Readability counts."



Características do Python

- ✓ **Sintaxe simples:** fácil de ler e escrever
- ✳️ **Multiparadigma:** suporta OO, funcional e procedural
- ⚡ **Interpretada e dinâmica:** sem compilação, tipagem flexível
- 🌐 **Portável:** roda em Windows, Linux, macOS
- 📦 **Biblioteca padrão rica:** pronta para diversas tarefas
- 🚀 **Ecossistema vasto:** web, dados, IA, automação e mais
- 🔗 **Extensível:** integração com C, C++, Java, APIs
- ❤️ **Grande comunidade:** ativa, colaborativa e com muito material



Aplicações do Python



Ciência de Dados



Analisar, tratar e visualizar dados

- pandas , numpy , matplotlib
- Ex: Gráficos de vendas, análise de CSVs



Inteligência Artificial

🧠 Criar modelos que aprendem com dados

- scikit-learn , tensorflow , keras
- Ex: Prever preços, reconhecer imagens



Desenvolvimento Web



Criar sites, APIs e sistemas web

- Flask , Django , FastAPI
- Ex: Blog pessoal, API de login

⚙️ Automação de Tarefas

🤖 Automatizar processos repetitivos

- `os` , `pyautogui` , `schedule`
- Ex: Renomear arquivos, enviar e-mails



Computação Científica



Resolver problemas técnicos e matemáticos

- `scipy` , `sympy` , `matplotlib`
- Ex: Simulações, equações diferenciais

Apps com Interface Gráfica

Criar aplicativos desktop ou mobile

- `tkinter` , `PyQt` , `kivy`
- Ex: Calculadora, app de tarefas



Desenvolvimento de Jogos



Criar jogos 2D simples

- pygame
- Ex: Pong, Flappy Bird, jogo da cobrinha



Web Scraping



Coletar dados automaticamente da internet

- `requests` , `BeautifulSoup` , `selenium`
- Ex: Preços de produtos, notícias, vagas



Cibersegurança



Testar segurança de sistemas

- `scapy` , `nmap` , `socket`
- Ex: Scan de portas, sniffers de rede



Finanças e Economia



Análise de investimentos e mercados

- `yfinance` , `pandas` , `quantlib`
- Ex: Gráficos de ações, backtests



Obrigado por assistirem

OBRIGADO

