

Os erros nunca devem passar silenciosamente e não ser que sejam explicitamente silenciados.
Perante a ambiguidade, recusar a tentação de adivinhar.
Deve haver uma - e de preferência apenas uma - forma óbvia de fazer.
Embora essa forma possa não ser óbvia à primeira vista, a menos que se seja Holandês.
Agora é melhor do que nunca. Embora nunca seja frequentemente melhor do que agora.
Se a implementação for difícil de explicar, é uma má ideia. Se a implementação for fácil de explicar, pode ser uma boa ideia.
Namespaces são uma boa ideia.



Natal de 1989

Introdução a Python

Victor Luz

1989

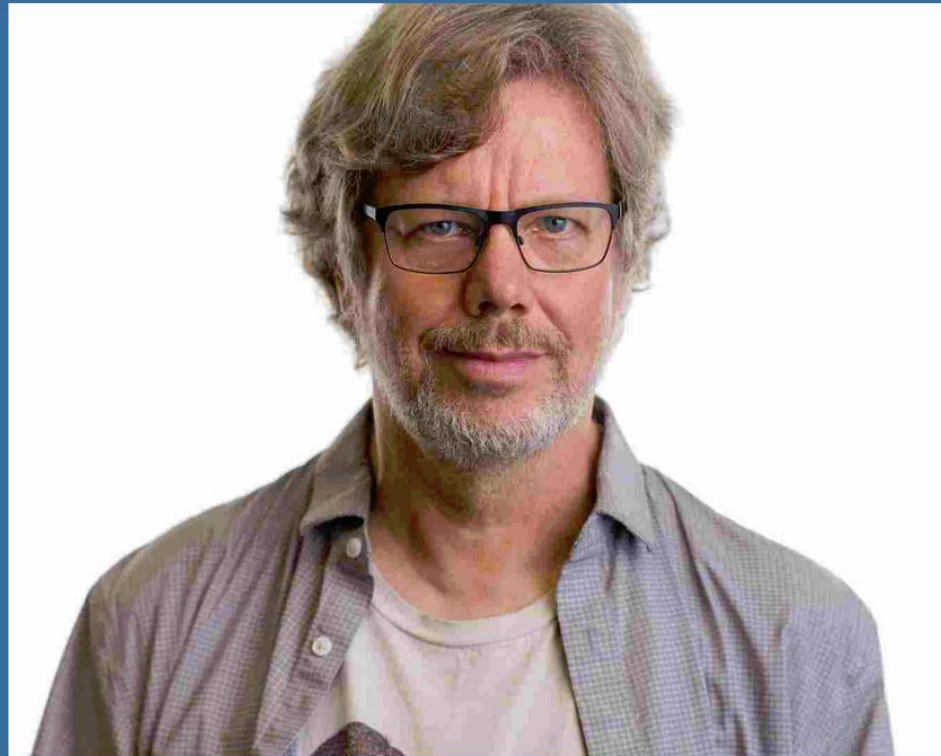
Pyt

- © Uma das linguagens mais utilizadas
- © Funciona desde microprocessadores
- © Fácil de aprender, fácil de ler, fácil de escrever
- © Milhares de bibliotecas disponíveis
- © Livre e de código aberto
- © Ferramentas de desenvolvimento

1989

- © Protestos na Praça Tiananmen
- © Lançamento da nave espacial Galileu pela NASA
- © A Guerra Fria termina na sequência da Conferência de Malta
- © O Muro de Berlim é derrubado
- © Exxon Valdez derrama 11 milhões de galões de petróleo
- © A Nintendo lança o Game Boy
- © Eleições livres na Polónia levam o Solidariedade ao poder

Natal de 1989



O ZEN de Python

Bonito é melhor do que feio. Explícito é melhor do que implícito.

Simples é melhor do que complexo. Complexo é melhor do que complicado.

Liso é melhor do que agrupado. Esperso é melhor do que denso.

A legibilidade é importante.

Os casos especiais não são suficientemente especiais para quebrar as regras, embora a praticidade vença a pureza.

Os erros nunca devem passar silenciosamente a não ser que sejam explicitamente silenciados.

Perante a ambiguidade, recusar a tentação de adivinhar.

Deve haver uma - e de preferência apenas uma - forma óbvia de fazer.

Embora essa forma possa não ser óbvia à primeira vista, a menos que se seja Holandês.

Agora é melhor do que nunca. Embora nunca seja frequentemente melhor do que agora.

Se a implementação for difícil de explicar, é uma má ideia. Se a implementação for fácil de explicar, pode ser uma boa ideia.

Namespaces são uma boa ideia.

Janeiro de 2024

503 888 projectos

5 227 010 versões

9 946 348 ficheiros

770 845 utilizadores

18 600 gigabytes de dados

Python 3

- © Uma das linguagens mais utilizadas no mundo
- © Funciona desde microprocessadores a centros de dados
- © Fácil de aprender, fácil de ler, fácil de programar
- © Milhares de bibliotecas disponíveis
- © Livre e de código aberto
- © Ferramentas de desenvolvimento livres

Projetos conhecidos a correr Python

YouTube

Netflix

Instagram

Spotify

Facebook

PayPal

Amazon

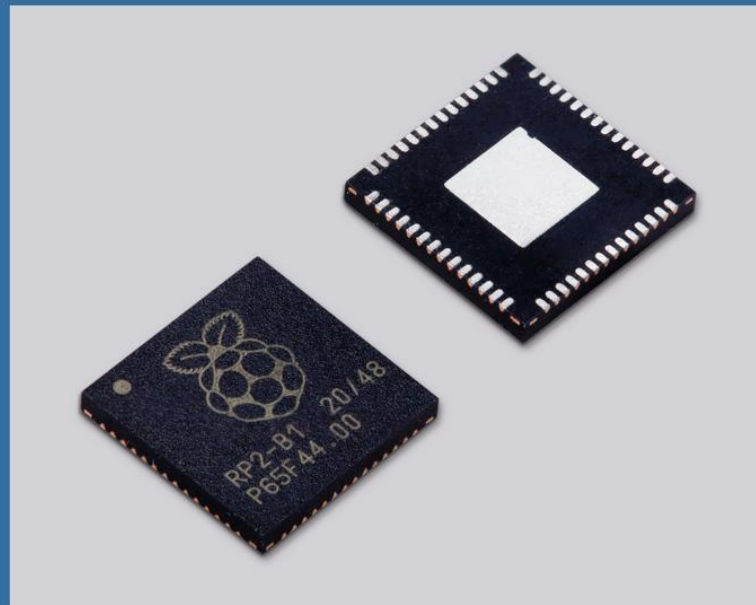
Uber

NASA

Industrial Light and Magic

Entre outros...

RP2040



Centro de dados do Facebook



Instalar Python

<https://www.python.org/downloads>

V3.11.7

Instalar GIT e LFS

<https://git-scm.com/downloads>

Instalar LFS

Instalar VSCodium

<https://vscodium.com>

Instalar Extensões Relevantes

Instalar Ungoogled Chromium

<https://ungoogled-software.github.io>

~V120

PIP, Jupyter e pipreqs

<https://jupyter.org>

<https://pypi.org/project/pipreqs>

Sites Relevantes

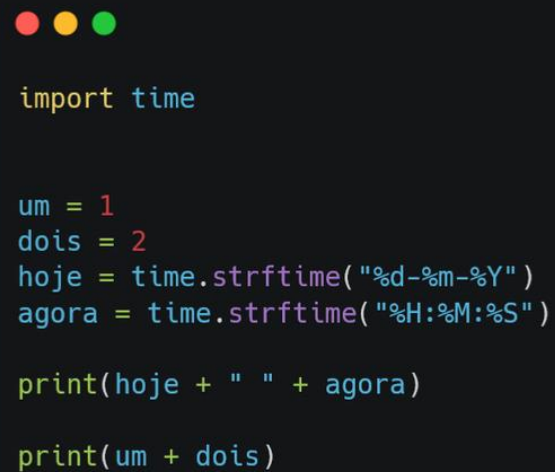
- © Github - <https://github.com/>
- © Pypi - <https://pypi.org/>
- © Google Collab - <https://colab.research.google.com/>
- © Infinity Free - <https://www.infinityfree.com/>
- © Render - <https://render.com/>
- © Hugging Face - <https://huggingface.co/>
- © Carbon - <https://carbon.now.sh/>
- © Publii - <https://getpublii.com/>

Clonar um repositório

<https://github.com/vluz/MiniCursoPy>

```
git clone https://github.com/vluz/MiniCursoPy.git
```

Primeiro Programa



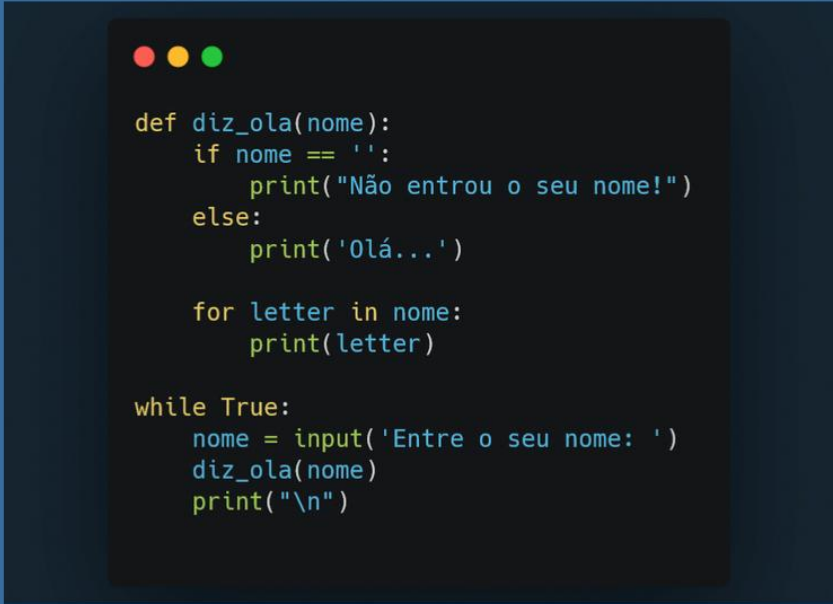
```
import time

um = 1
dois = 2
hoje = time.strftime("%d-%m-%Y")
agora = time.strftime("%H:%M:%S")

print(hoje + " " + agora)

print(um + dois)
```

Segundo Programa



```
def diz_ola(nome):  
    if nome == '':  
        print("Não entrou o seu nome!")  
    else:  
        print('Olá...')  
  
    for letter in nome:  
        print(letter)  
  
while True:  
    nome = input('Entre o seu nome: ')  
    diz_ola(nome)  
    print("\n")
```

Segundo com GUI

```
import streamlit as st

def diz_ola(nome):
    st.write('Olá...')
    st.write(nome)

st.title("Olá GUI")
nome = st.text_input("Entre o seu nome e pressione o botão:")
if st.button("Diz"):
    if nome:
        diz_ola(nome)
    else:
        st.write("Não entrou o seu nome!")
```

Recursos Adicionais

Think Python:

<https://greenteapress.com/thinkpython2/thinkpython2.pdf>

Cursos livres no YouTube:

https://www.youtube.com/watch?v=_uQrJ0TkZlc

<https://www.youtube.com/watch?v=eWRfhZUzrAc>

IOS e Android

<https://apps.apple.com/us/app/carnets-jupyter/id1450994949>

<https://apps.apple.com/us/app/libterm/id1380911705>

<https://termux.dev/en/> + pkg install python

<https://dev.to/codeledger/how-to-get-visual-studio-code-to-run-in-termux-on-android-405j>

Obrigado!

Dedicado à Lana por Suporte e Paciência