



Python

Seminário referente a matéria de Linguagens de Programação

Alunos: Gabriel Samarane, Gabriel Araújo, Arthur Matos, João Madeira, Vitória Simil



PUC Minas

Sumário



1 Introdução

Linguagens
Relacionadas

5 



2 Histórico

Programas

6 



3 Paradigmas

Prática

7 



4 Características
Marcantes

Bibliografia

8 



01

INTRODUÇÃO

Introdução



- Python é uma linguagem fácil de aprender e poderosa.
- Possui uma abordagem simples mas efetiva de programação orientada a objetos.
- É uma linguagem de programação para propósitos diversos (GPL) que roda em basicamente qualquer arquitetura de sistema e pode ser usada para um leque enorme de aplicações em diferentes áreas.



- Grande demanda profissional, sendo uma linguagem muito valorizada.
- Usada em muitas áreas, como: análise de dados, aprendizado de máquina, desenvolvimento web, jogos, automação e computação científica.
- Possui muitas bibliotecas e frameworks para diferentes tipos de atuações.
- Usado por grandes empresas, como:



The Zen of Python

```
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import this
The Zen of Python, by Tim Peters

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!
>>>
```

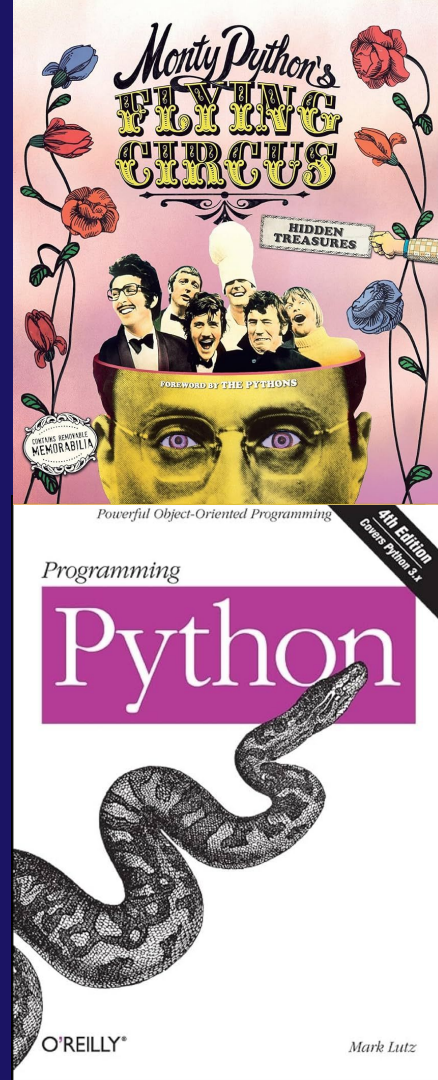


02

HISTÓRIA

História da Linguagem

- Centro de Pesquisa Holandês (CWI) - 1989.
- Devido à limitações de hardware com a linguagem ABC, **Guido Van Rossum** decidiu criar sua própria linguagem.
- Origem do nome Python: programa de TV **Monty Python's Flying Circus** .





Centrum Wiskunde
& Informatica (CWI),
Amsterdã
Guido Van Rossum

Cronologia

Versão 1.0.0

Classes com herança,
tratamento de exceções,
funcionamento modular

Versão 3.0.0

Correção das principais falhas no
design, quebra compatibilidade com
versões anteriores

Versão 3.12.3

1989

1991

1994

2000

2008

2018

Maio
2024

Liberação da primeira
versão pública do Python
(0.9.0)

Geração de listas,
novo sistema

Versão 2.0.0

Afastamento de
Guido Van Rossum



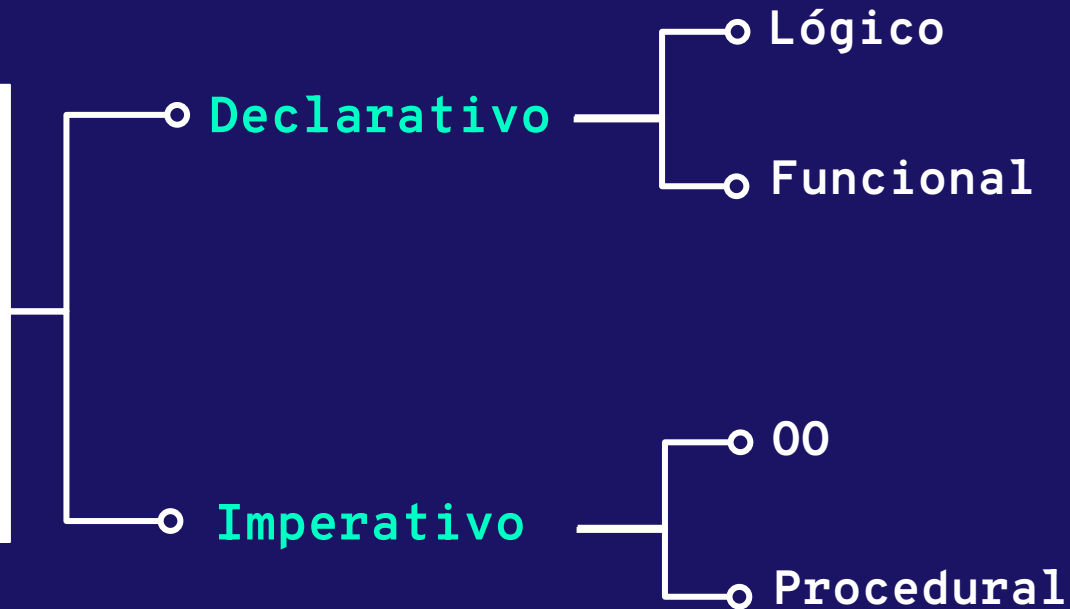
03

PARADIGMAS

Paradigmas

PARADIGMAS

Fornecem um conjunto de regras e diretrizes que orientam os desenvolvedores na resolução de problemas com soluções eficientes.



Paradigmas

PARADIGMAS

Fornecem um conjunto de regras e diretrizes que orientam os desenvolvedores na resolução de problemas com soluções eficientes.

- Python é uma **multiparadigma**
- Procedural
- Orientado a Objetos
- Funcional

Paradigmas: Boas Práticas

PARADIGMA PROCEDURAL

- **Divida** o código em funções e procedimentos

- **Utilize** nomes de variáveis descritivos e significativos

- **Faça** uso de comentários claros e explicativos

Paradigmas: Boas Práticas

PARADIGMA 00

- **Siga** os princípios de encapsulamento, herança e polimorfismo.

- **Evite** a criação de classes muito grandes e complexas

- **Utilize** nomes de classes, métodos e atributos que sejam descritivos

- **Utilize** exceções para lidar com erros

Paradigmas: Boas Práticas

PARADIGMA FUNCIONAL

- Utilize funções puras sempre que possível

- Faça uso de funções de ordem superior

Paradigmas: por que conhecer?

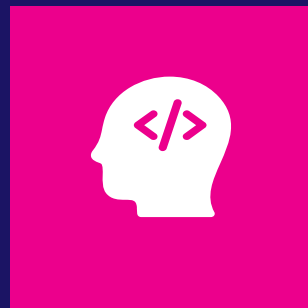
Melhoria na
qualidade do código



Flexibilidade



Reutilização de
Códigos





04

CARACTERÍSTICAS MARCANTES

Características Marcantes

- Pode interagir diretamente com a linguagem C durante a execução;
- Toda variável é um objeto;
- Tipagem dinâmica;
- Sintaxe simples e clara;
- Atribuição Múltipla;
- Blocos por indentação



Variáveis Como Objetos / Tipagem Dinâmica

```
x = 'teste'

print(x.upper())

x = {1,2,3}

print(x.pop())
```

```
TESTE
1
```

- Chamadas de métodos em cima de variáveis;
- Tipagem acontece durante a execução do programa.

Atribuição Múltipla

Exemplo :

```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3,random_state=100)
```

- A função retorna uma tupla, dicionário, lista, etc;
- São associados às variáveis os valores desta estrutura retornada, respectivamente;
- Extremamente útil para legibilidade e reuso de código

Blocos por Indentação

```
def printa_se_par(x):  
    for i in range(len(x)):  
        if x[i] % 2 == 0:  
            print(x[i])  
x = [1,2,3,4,5,6]  
printa_se_par(x)
```

```
2  
4  
6
```

- Exemplo de bloco de código que utiliza indentação;
- Indentação aninhada



05

LINGUAGENS
RELACIONADAS

Linguagens Relacionadas



LINGUAGENS BASE

- ABC
- C
- HASKELL
- ICON

LINGUAGENS DERIVADAS

- BOO
- COBRA
- GO

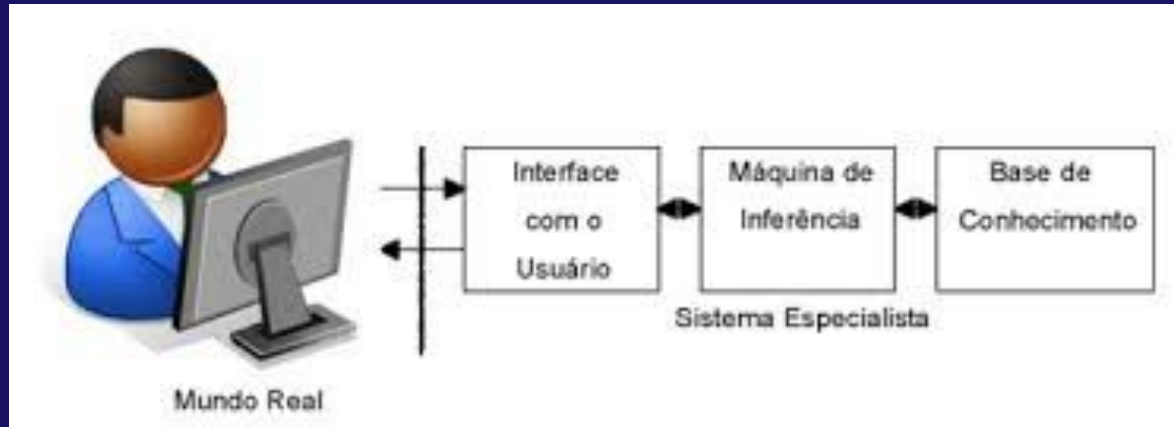


Linguagens Relacionadas

PYTHON E PROLOG

- Relação
- Bibliotecas

SISTEMAS ESPECIALISTAS





06

EXEMPLOS DE PROGRAMAS

Dicionários

```
alunos = {  
    'João': [85, 78, 92],  
    'Maria': [95, 88, 90],  
    'Pedro': [78, 74, 80]  
}  
  
# MÉDIA DE CADA ALUNO  
medias = {aluno: sum(notas)/len(notas) for aluno, notas in alunos.items()}  
  
print(medias)
```

✓ 0.0s

```
{'João': 85.0, 'Maria': 91.0, 'Pedro': 77.33333333333333}
```

Dicionários são estruturas de dados fundamentais que permitem manipulações rápidas e eficientes.

Explicação do código:
Cálculo da média das notas de cada aluno.

Biblioteca NumPy

```
import numpy as np

# CRIAR ARRAY NÚMEROS ALEATÓRIOS 5x5 DE 0 A 1
array = np.random.rand(5, 5)

#array = 10 + (20 - 10) * array

# MÉDIA E DESVIO PADRÃO
media = np.mean(array)
desvio_padrao = np.std(array)

print("Array:\n", array)
print("Média:", media)
print("Desvio Padrão:", desvio_padrao)
```

✓ 0.0s

Array:

```
[[0.44530079 0.37144366 0.50103779 0.42770421 0.38836316]
 [0.13511728 0.21808506 0.22611255 0.87097398 0.76100787]
 [0.00628488 0.14881733 0.60514727 0.29440742 0.26985219]
 [0.20939534 0.99613387 0.86272167 0.03551604 0.66139101]
 [0.36538306 0.2849278 0.64953252 0.16061433 0.3387881 ]]
```

Média: 0.4093623675788294

Desvio Padrão: 0.2630932489266776

Python se destaca pela vasta quantidade de bibliotecas externas disponíveis.

Exemplo: Biblioteca NumPy

Instalação:

```
PS C:\Users\User\Estudos\LP_SEMINARIO> pip install numpy
Collecting numpy
  Using cached numpy-1.26.4-cp312-cp312-win_amd64.whl.metadata (61 kB)
Using cached numpy-1.26.4-cp312-cp312-win_amd64.whl (15.5 MB)
Installing collected packages: numpy
Successfully installed numpy-1.26.4
PS C:\Users\User\Estudos\LP_SEMINARIO> █
```

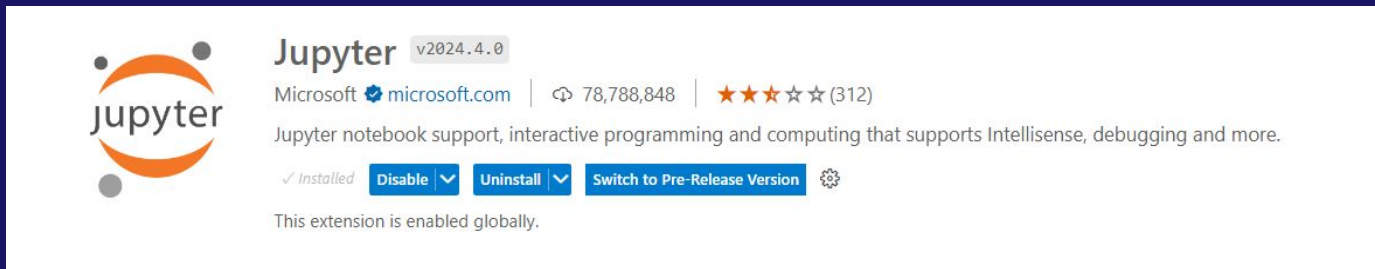


07

PRÁTICA

Instalação Jupyter VsCode

Instalando o Visual Studio Code: <https://code.visualstudio.com/>



```
PS C:\Users\User\Estudos\LP_SEMINARIO> pip install ipykernel
```



08

BIBLIOGRAFIA

Bibliografia

- History of Python. GeeksforGeeks, 13 de maio de 2024. Disponível em: <<https://www.geeksforgeeks.org/history-of-python/>>. Acesso em: 22 de maio de 2024.
- A história do Python. As versões de uma linguagem única. Tokio School, 07 de jul de 2021. Disponível em: <<https://tokioschool.pt/noticias/historia-python/#:~:text=A%20vers%C3%A3o%201.0%20foi%20lan%C3%A7ada,suas%20caracter%C3%ADsticas%20fundamentais%3A%20funcionamento%20modular.>>. Acesso em: 22 de maio de 2024.
- O Que é Python: Conheça Uma das Linguagens de Programação mais Populares do Mundo. Hostinger Tutoriais, 07 de mar. de 2024. Disponível em: <<https://www.hostinger.com.br/tutoriais/python-o-que-e>>. Acesso em: 22 de maio de 2024.
- Paradigmas De Linguagens De Programação Em Python: Descubra As Melhores Práticas. Disponível em: <<https://awari.com.br/paradigmas-de-linguagens-de-programacao-em-python-descubra-as-melhores-praticas/#:~:text=Os%20paradigmas%20de%20linguagens%20de%20programa%C3%A7%C3%A3o%20em%20Python%20s%C3%A3o%20abordagens.na%20cria%C3%A7%C3%A3o%20de%20solu%C3%A7%C3%B5es%20eficientes.>>. Acesso em: 27 de maio de 2024.



09

CONSIDERAÇÕES

Considerações

Arthur

Python sempre foi uma linguagem que me trouxe interesse, mas nunca tive contato direto antes. O seminário foi uma boa oportunidade de conhecer e aprender Python, e com certeza é uma linguagem que abrirá portas no mercado de trabalho.

Gabriel Samarane

Python é uma linguagem extremamente útil em tópicos de computação aplicada, algo que sempre me despertou muito interesse. Após o seminário, aprendi significativamente sobre os fundamentos da linguagem, o que me incentivou ainda mais a estudá-la.

Considerações

Gabriel Silva

Antes de realizar as pesquisas para o desenvolvimento deste trabalho, enxergava python como uma linguagem incompleta com o único propósito de ser didática. O seminário fez com que meu interesse aumentasse principalmente por entender melhor sua importância na área de inteligência artificial além de todas as outras aplicações citadas durante a apresentação.

Vitória Símil

Meu interesse pela linguagem Python começou depois que meu interesse por dados surgiu, apesar de eu não ter me aprofundado no aprendizado da linguagem o seminário me permitiu entender o básico da linguagem bem como observar na prática o conteúdo aprendido durante o semestre.

João Madeira

Python sempre me despertou interesse desde o início do curso, e comecei a aprendê-la de uma forma mais descontraída, mas nunca me aprofundei. O seminário foi um ótimo ponto de partida para pesquisar e aprender mais sobre Python.