

LIGUAGEM DE PROGRAMAÇÃO PYTHON

SEMINÁRIO



PUC Minas

AUTORES: Bruno Santiago de Oliveira; Fábio Freire Kochem; Fernando Campos da Silva Dal Maria

INSTITUIÇÃO: Pontifícia Universidade Católica de Minas Gerais

Belo Horizonte – 29/05/2022

Sumário

Histórico	(1)
Características Marcantes	(5)
Paradigmas	(7)
Estrutura	(8)
Aplicação	(17)
Prática	(20)
Referências	(27)

Histórico – Criador

Guido van Rossum

1

Na década de 80 trabalhou na Centrum Wiskunde & Informatica (**CWI**) em um sistema operacional chamado Amoeba e na linguagem **ABC**.

Em dezembro de 1989 começou a trabalhar em uma **linguagem semelhante a ABC** que poderia se comunicar com o sistema operacional e seria adequada para o desenvolvimento rápido de utilitários.



Observação: O "distinto engenheiro" trabalha atualmente na divisão de desenvolvimento da Microsoft.

Histórico - O nome da linguagem?

2

Python

Teve seu nome inspirado no programa de televisão Monty Python's Flying Circus

Inspirada em:

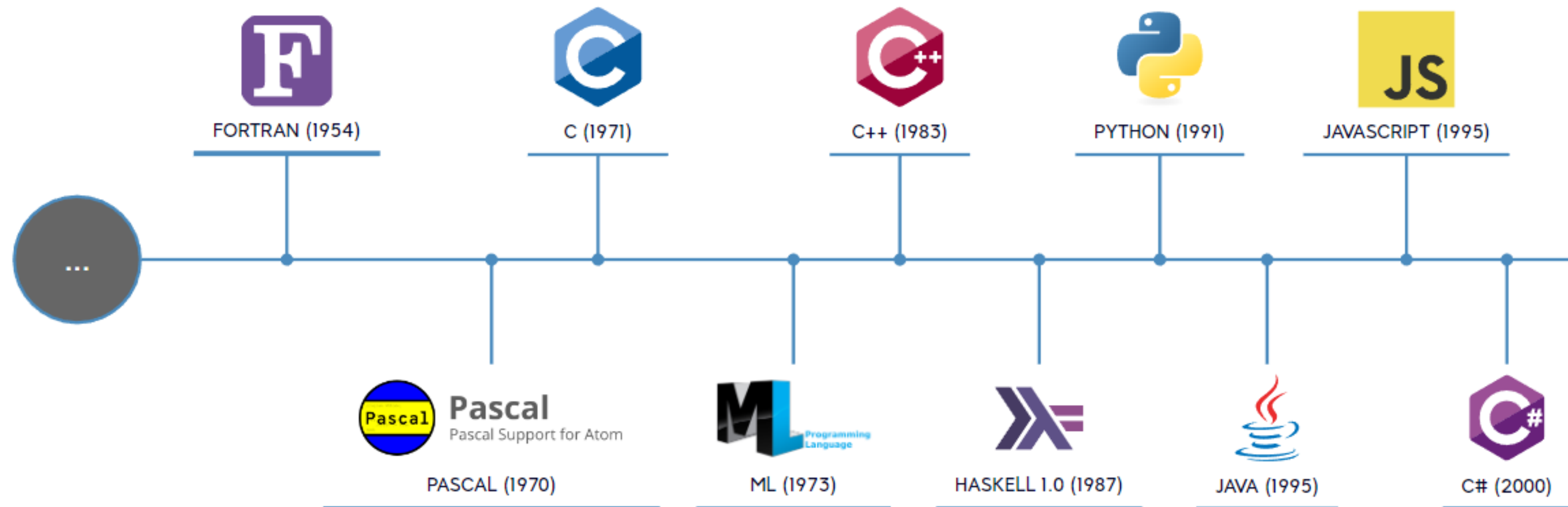


ABC



Histórico - Linha do Tempo

3



Histórico – Versões

Python 1.0  Python 2.0  Python 3.0

Fevereiro de 1991

Primeiro workshop de python em novembro de 1994.

Saída de Guido da CWI para a Corporation for National Research Initiatives (**CNRI**).

CNRI quer mudar o modelo de distribuição de **Python** para gerar lucro

Outubro de 2000

Conflito de interesses:
Guido sai da **CNRI** para uma startup e depois para uma empresa chamada Zope.

Com o aumento das aplicações Python a versão 2 foi revisada.

A terceira versão de é desenvolvida: **Python 3000**

Dezembro de 2008

Referências – (SEVERANCE, 2015a); (SEVERANCE, 2015b); (ROSSUM, 1996)

Características Marcantes

A linguagem é:

- Interpretada
- Multi-paradigma
- Estrutura de Bloco Aninhada
- Dinâmica e Fortemente Tipada - *Duck Type*
- Amarrada Dinamicamente
- *Garbage Collected* - Utiliza um coletor de lixo
- **Simples e Legível**

Interpretadores

Existem implementações em diferentes linguagens:

CPython



Jython

PyPy



IronPthon

Referências – (PYTHON SOFTWARE FOUNDATION, 2023);

Belo Horizonte – Linguagem de Programação Python

Paradigmas da Linguagem

7

- **Programação funcional (PF):** se baseia na declaração do que se deseja que um programa faça, em vez de como ele deve ser executado.
- **Programação Imperativa (PI):** código deve descrever explicitamente o processo de execução de determinado algoritmo.
 - **Programação Orientada à Objetos (POO):** fornece abstrações utilizando classes e objetos.
 - **Programação Procedural (PP):** consiste em organizar funcionalidades de um programa em trechos de código reutilizáveis chamados procedimentos.

Estrutura - Duck Type

Inicialmente, Python foi desenvolvida para apresentar **equivalência de tipos por nomes**. Posteriormente uma **equivalência de tipo estrutural** foi adicionada a linguagem:

"If it walks like a duck, and it quacks like a duck, then it must be a duck."

Dessa forma a linguagem não está interessada em identificar se aquele objeto é de fato um "Pato", apenas precisa identificar que aquele tipo age como um "Pato". Para isso basta apenas verificar as assinaturas dos métodos.

Estrutura - Duck Type

9

Equivalência de tipos por nome

Duck Type

```
class Bucket(Sized, Iterable[int]):  
    ...  
    def __len__(self) -> int: ...  
    def __iter__(self) -> Iterator[int]: ...
```

```
class Bucket:  
    ...  
    def __len__(self) -> int: ...  
    def __iter__(self) -> Iterator[int]: ...
```

```
def collect(items: Iterable[int]) -> int: ...  
result = collect(Bucket())
```

Estrutura - Tipos

Tipos primitivos

- int¹
- float
- complex
- bool
- str
- bytes
- NoneType

Tipos Compostos e Enum

- list
- tuple
- set
- frozenset
- dict
- bytearray
- NamedTuple
- Enum
- Queue
- Stack
- Heap
- Deque
- OrderedDict
- Counter
- ChainMap
- DefaultDict

Estrutura - Operadores Básicos

11

```
+ # adição  
- # subtração  
* # multiplicação  
// # divisão de inteiros  
/ # divisão de pontos flutuantes  
** # potenciação  
@ # multiplicação de matrizes  
% # resto da divisão inteira  
= # atribuição  
:= # atribuí valor à um identificador quando em uma expressão condição
```


Estrutura - Operadores Bit a Bit e Lógicos

12

Bit a Bit

& ^ | ~ << >>

Lógicos

and or not <= < >= > == !=

Escapes

`break` `continue` `pass` `return`

Exceções

BaseException

- |—— BaseExceptionGroup
- |—— GeneratorExit
- |—— KeyboardInterrupt
- |—— SystemExit
- |—— Exception

Estrutura – Iteração e Condicionais

14

Iteráveis

```
while expressao: ...  
else: ...
```

```
for identificador in interavel: ...  
else: ...
```

Condicional

```
if expressao: ...  
elif expressao: ... # opcional  
else: ... # opcional
```

Referências – (PYTHON SOFTWARE FOUNDATION, 2023)

Belo Horizonte – Linguagem de Programação Python

Estrutura – List Comprehension & Operador Ternário

15

List Comprehension

1

```
identificador for identificador in interavel if expressao
```

"Operador Ternário"

1

```
... if expressao else: ...
```

Estrutura - POO & Encapsulamento

16

Recursos da Linguagem:

Encapsulamento
Herança Múltipla
Polimorfismo¹
Dataclasses

Métodos Especiais
Atributos Especiais

Observação:

Visibilidade __<attr>

```
class Student(Human, Programmer):
    def __init__(self, name: str, age: int, *languages):
        Human.__init__(self, name, age)
        Programmer.__init__(self, [x for x in languages])

    def __str__(self):
        return f'Student {self.name} work with {self.languages}'

    def __eq__(self, other):
        return self.name == other.name and self.age == other.age

fernando = Student('Bruno', 20, 'C++', 'Python')
fabio = Student('Bruno', 20, 'Java', 'Python', 'C')
print(fabio)
print(fernando == fabio)
```


Aplicações

Aplicações Web

Aplicações Científicas

Aplicações Educacionais

Aplicações Empresariais

Criação de Desktop GUI



PUC Minas



Stanford



2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)

UNO: A Web Application using Django

A Low-Effort Analytics Platform for Visualizing
Evolving Flask-Based Python Web Services



19th Brazilian Congress of Thermal Sciences and Engineering
November 6th–10th, 2022, Bento Gonçalves - RS - Brazil

**ENCIT2022-0493 Development of a Solar Radiation Map Using Artificial
Intelligence Techniques**

Referências – (AMAUURI, et al., 2022); (BAIRAGI, I. A., 2021); (NAGPAL, A, 2019)

Aplicações - Correlações



```
print('Hello World!')
```



```
#include <iostream>
int main() {
    std::cout << "Hello World!\n";
    return 0;
}
```

```
console.log('Hello World!');
```



```
print "Hello World!\n";
```



```
public class Hello {
    public static void main (String[] args) {
        System.out.println("Hello, world!");
    }
}
```

Prática - Instalação

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



Prática – Instalação

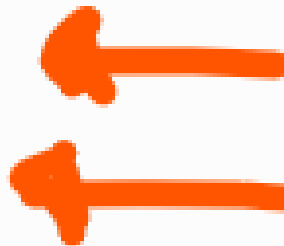
Windows

Stable Releases

- [Python 3.10.11 - April 5, 2023](#)

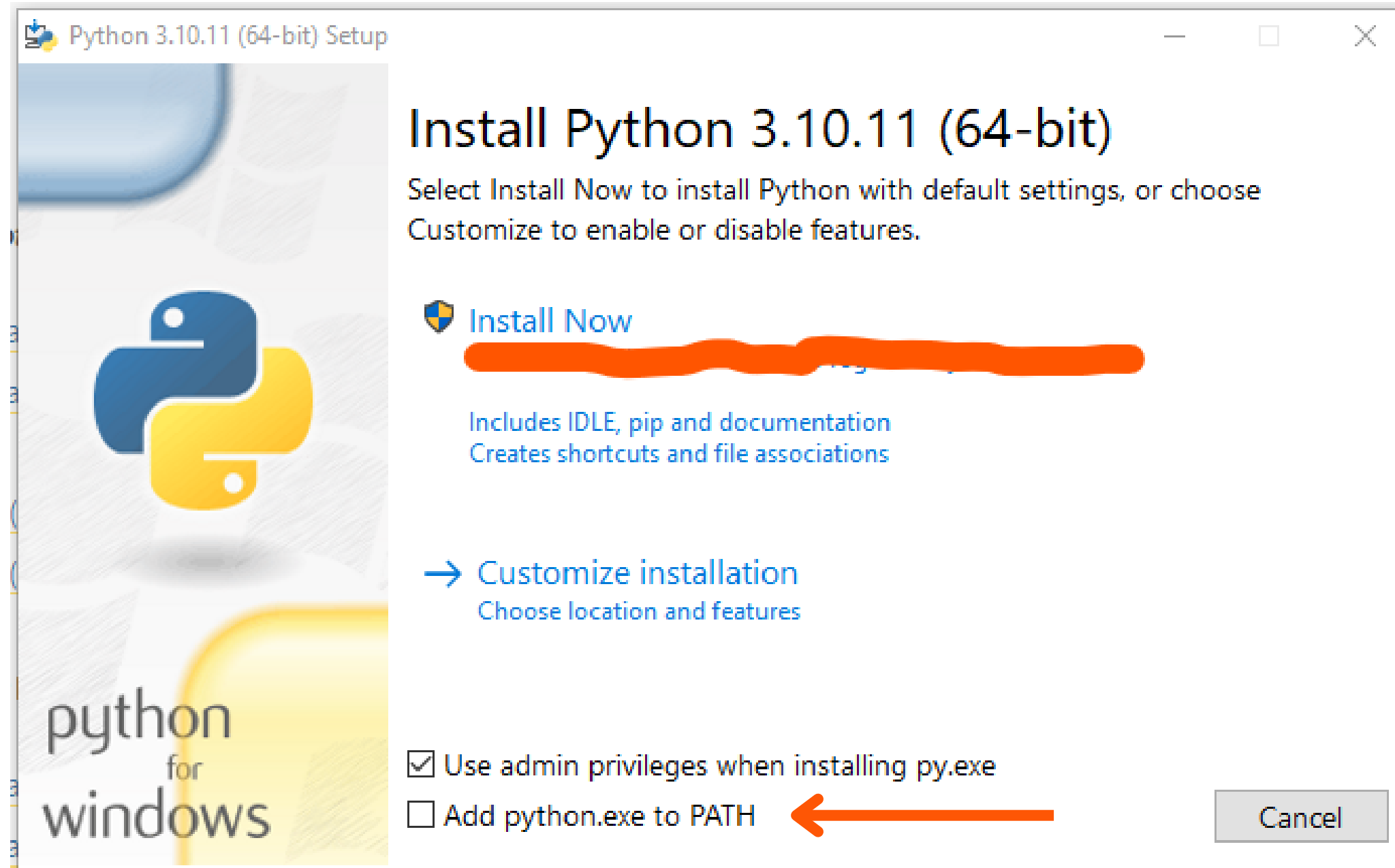
Note that Python 3.10.11 cannot be used on Windows 7 or earlier.

- Download [Windows embeddable package \(32-bit\)](#)
- Download [Windows embeddable package \(64-bit\)](#)
- Download [Windows help file](#)
- Download [Windows installer \(32-bit\)](#)
- Download [Windows installer \(64-bit\)](#)
- [Python 3.11.3 - April 5, 2023](#)



Prática – Instalação Windows

22



Prática – Instalação

Linux/Unix & MacOS

23

Verifique se já tem o Python instalado, se você usa GNU/Linux ou macOS (10.2 ou superior), provavelmente já possui alguma versão do Python instalada. Para conferir, digite em um terminal:

```
$ which python
```

ou

```
$ which python3
```

Prática – Instalação

Linux/Unix

24

Caso não tenha instalado, deve fazer a instalação por gerenciador de pacotes, sendo os mais famosos: apt-get(Debian, Ubuntu) e Yum(RedHat, CentOS):

Apt-get

```
$ sudo apt-get install python3
```

Yum

```
$ sudo yum install python3
```

Prática – Instalação

MacOS

Caso não tenha instalado, é preciso fazer a instalação do XCode, que pode ser baixado na App Store, command line tools e dos gerenciadores de pacotes pip e homebrew. Siga os passos:

```
$ xcode-select --install
```

```
$ sudo easy_install pip
```

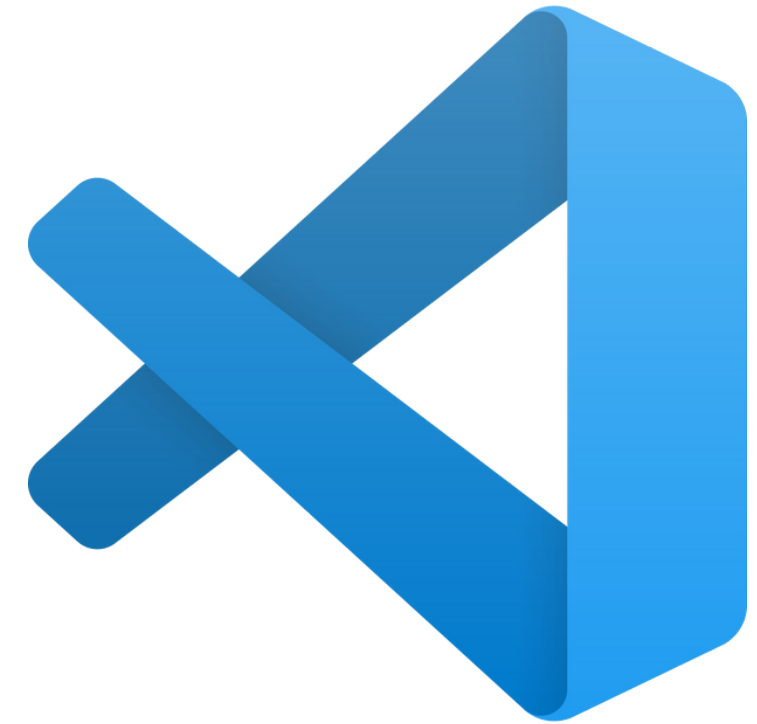
```
$ sudo pip install --upgrade pip
```

```
$ ruby -e "$(curl -fsSL https://raw.githubusercontent.com/mxcl/homebrew/go)"
```

```
$ brew install python3
```

Prática - IDEs

IDE's mais utilizadas para programar em Python



Considerações Finais

A linguagem Python é utilizada em múltiplas aplicações, sejam elas aplicações web, científicas ou empresariais. Sua estrutura sintática e suas abstrações são poderosas e se destacam como um pilar para a popularidade e para o uso da linguagem na comunidade científica e por diversos desenvolvedores. Além disso suas bibliotecas também são muito bem trabalhadas e pleiteadas por diversas empresas e instituições que deseja sistemas robustos e escaláveis. Para mais

Informações acesse:



Obrigado a todos!

Referências

28

AMAUURI, R. et al. Development of a Solar Radiation Map Using Artificial Intelligence Techniques. In: 2022 19th Brazilian Congress of Thermal Sciences and Engineering, **Associação Brasileira de Engenharia e Ciências Mecânicas (ABCM)**. [S.l.: s.n.], 2022. 8 p.

BAIRAGI, I. A. et al. Uno: A web application using django. In: 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N). [S.l.: s.n.], 2021. p. 1371–1374.

CHALLAPALLI, S. S. N. et al. Web development and performance comparison of web development technologies in node.js and python. In: 2021 International Conference on Technological Advancements and Innovations (ICTAI). [S.l.: s.n.], 2021. p. 303–307.

DYER, R.; CHAUHAN, J. An exploratory study on the predominant programming paradigms in python code. In: . New York, NY, USA: **Association for Computing Machinery**, 2022. (ESEC/FSE 2022), p. 684–695. ISBN 9781450394130. Disponível em: <<https://doi.org/10.1145/3540250.3549158>>.

LUTZ, M. **Learning Python**. 5. ed. [S.l.]: O'Reilly, 2013. 1540 p.

MILOJKOVIC, N.; GHAFARI, M.; NIERSTRASZ, O. It's duck (typing) season! In: 2017 **IEEE/ACM 25th International Conference on Program Comprehension (ICPC)**. [S.l.: s.n.], 2017. p. 312–315.

Referências

29

NAGPAL, A.; GABRANI, G. Python for data analytics, scientific and technical applications. In: 2019 **Amity International Conference on Artificial Intelligence (AICAI)**. [S.l.: s.n.], 2019. p. 140–145.

PYTHON SOFTWARE FOUNDATION. Applications for Python. 2023. Acesso em: 23 Abril 2023. Disponível em: <<https://www.python.org/>>.

PYTHON WIKI. The Python Wiki. 2023. Acesso em: 20 Maio 2023. Disponível em: <<https://wiki.python.org/moin/>>.

ROSSUM, G. van. **Foreword for "Programming Python"**(1st ed.). Reston, Virginia: [s.n.], 1996. Acesso em: 23 Abril 2023. Disponível em: <<https://legacy.python.org/doc/essays/foreword/>>.

ROSSUM, G. van. **Guido van Rossum – Resume**. 2023. Acesso em: 23 Abril 2023. Disponível em: <<https://gvanrossum.github.io//Resume.html>>.

SEBESTA, R. W. **Concepts of programming languages**. 12. ed. [S.l.]: Pearson, 2019. 752 p.

SEVERANCE, C. Guido van rossum: The early years of python. **Computer**, v. 48, n. 2, p. 7–9, 2015.

Referências

30

SEVERANCE, C. Guido van rossum: The modern era of python. **Computer**, v. 48, n. 3, p. 8–10, 2015.

STACKOVERFLOW. **Developer Servey 2022**. 2022. Acesso em: 18 Maio 2023. Disponível em: <<https://survey.stackoverflow.co/2022/#most-popular-technologies-language>>.