



# Introdução à Linguagem Python

Paradigmas de Linguagens de Programação

**Rômulo Souza Fernandes**  
**Ausberto S. Castro Vera**

8 de setembro de 2022



Copyright © 2022 Rômulo Souza Fernandes e Ausberto S. Castro Vera

UENF - UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE DARCY RIBEIRO

CCT - CENTRO DE CIÊNCIA E TECNOLOGIA

LCMAT - LABORATÓRIO DE MATEMÁTICAS

CC - CURSO DE CIÊNCIA DA COMPUTAÇÃO

*Primeira edição, Setembro 2022*



## Sumário

<b>1</b>	<b>Introdução</b>	<b>5</b>
<b>1.1</b>	<b>História da linguagem Python</b>	<b>5</b>
<b>1.2</b>	<b>Áreas de Aplicação da Linguagem</b>	<b>6</b>
1.2.1	Big Data	6
1.2.2	Orientação a objetos	7
1.2.3	Pentest	7
	<b>Bibliografia</b>	<b>9</b>
	<b>Index</b>	<b>11</b>





# 1. Introdução

O Python é uma linguagem orientada a objetos de alto nível, que possui uma sintaxe simples e objetiva, assim colaborando para a fácil compreensão do código-fonte e permitindo que a linguagem seja produtiva. O Python contém várias estruturas de alto nível, como hora, data, dicionários, listas, complexos, entre outras estruturas, contém um amplo conjunto de módulos disponíveis para utilização, frameworks que podem ser acrescentados, possui ferramentas de outras linguagens atuais, como persistência, unidades de teste, geradores, introspecção e metaclasses, além de ter disponíveis diversas bibliotecas, como IPython, Matplotlib, mIPy, NumPy, Pandas, SciPy, ScraPy, entre outras bibliotecas conhecidas.

O Python é uma linguagem multiparadigma, suportando a programação orientada a objetos, modular e funcional. A linguagem Python foi criada na Holanda, no ano de 1990, por Guido van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação. [Bor14]

A linguagem Python é de código aberto, porém o criador Guido van Rossum possui a função central de decidir a evolução da linguagem. O Python se popularizou e se tornou a linguagem de desenvolvimento de aplicações mais indicada para iniciantes, assim sendo aconselhada como primeira linguagem de programação. [Per16]

## 1.1 História da linguagem Python

O intuito de Guido van Rossum era criar uma linguagem que pudesse suprir suas exigências, assim criando o Python, com base na linguagem ABC, mas solucionando as incoerências encontradas por ele na linguagem. O Python tinha como usuários principais os engenheiros e físicos.

A seguir um pouco da história da linguagem Python, baseados em [Per16] e [Bor14] :

- O Holandês Guido van Rossum foi o autor principal da linguagem Python. O autor trabalhava no CWI (Centrum Wiskunde & Informatica), localizada em Amsterdã na Holanda.
- O nome Python não veio da espécie de serpente e sim do seriado de comédia preferido do autor da linguagem, chamado Monty Python's Flying Circus.
- A versão 0.9.0 do Python foi lançado em 1991, incluindo manipulação de exceções, classes, listas e strings. Incluía também alguns aspectos de programação funcional como lambda, maps, filter e reduce.

- No ano de 1995, o autor da linguagem continuou seu trabalho sobre Python na Corporation for National Research Initiatives (CNRI) em Reston, Virginia, USA.
- Em Maio de 2000, Guido van Rossum e o grupo de desenvolvimento do Python se mudaram para BeOpen.com, assim formando a equipe BeOpen PythonLabs.
- A versão 1.6 do Python foi lançada em 5 de setembro de 2000.
- A versão 2.0 do Python foi lançada em 16 de outubro de 2000.
- A versão 3.0 do Python foi lançada em 3 dezembro de 2008.

## 1.2 Áreas de Aplicação da Linguagem

O Python está entre as linguagens de programação mais utilizadas no mundo, é utilizado por usuários individuais, mas sua aplicação se estende para empresas reais. A natureza do Python é de propósito geral, assim tornando a linguagem aplicável em quase todas as áreas. A IBM, Seagate e Hewlett-Packard, que utilizam o Python para testes de hardware. O Yahoo! e Google usam a linguagem em serviços de Internet. Já a empresa Industrial Light and Magic e outras empresas de filmes utilizam o Python na produção de animações. Entre todas as aplicações para o Python atualmente, o ponto em comum é que a linguagem é usada em todo o espectro, em questão de domínios de aplicação. [Lut07]

### 1.2.1 Big Data

Atualmente o Python é uma das melhores linguagens de programação para trabalhar com Big Data. Um dos motivos dessa preferência de uso é, o suporte avançado de inúmeras bibliotecas e frameworks, muitas das bibliotecas são voltadas para lidar com Big Data, dando suporte e auxiliando na implementação de algoritmos de Machine Learning e Data Analytics. Abaixo algumas das bibliotecas de software livre:

- SciPy: Utilizada para computação técnica e computação científica, possibilita a interpolação, otimização, integração e modificação de dados utilizando funções especiais, álgebra linear, etc.
- NumPy: Utilizada para computação numérica para dados com formas de grandes matrizes multidimensionais e arrays. A biblioteca também oferece diversas funções matemáticas de alto nível, para manipular os dados com transformada de Fourier, álgebra linear, processamento de números aleatórios, etc.
- Scikit-learn: Utilizada para Machine Learning, relacionado a vários algoritmos de regressão, clustering e classificação. Pode ser utilizado também em conjunto com outras bibliotecas, como a NumPy e SciPy.
- Pandas: Utilizada para análise e manipulação de dados, oferece diversas estruturas de dados e operações para manipulação de dados, no formato de séries temporais e tabelas numéricas. A biblioteca também dispõe diferentes ferramentas para gravar e ler dados, entre estruturas de dados na memória e diferentes formatos de arquivo.

O Python possui uma sintaxe simples, possibilitando uma fácil leitura do código, assim tanto os estudantes iniciantes quanto os desenvolvedores experientes, podem se concentrar melhor no objetivo ao invés de se desgastar se concentrando nas nuances técnicas da linguagem que está utilizando. Sendo assim, a linguagem preferida dos Cientistas de dados e Engenheiros de Big Data.

A linguagem Python é extremamente flexível, permitindo finalizar mais trabalhos com menor número de linhas de código. O Python também é escalável na manipulação de dados em grandes quantidades, sendo um ponto muito importante quando se trata de Big Data. Comparando o Python com outras linguagens de programação utilizadas em Big Data Analytics, como R e Java, não são tão escaláveis e flexíveis como o Python, onde se o volume de dados aumentar, o Python sem dificuldades pode aumentar a velocidade de processamento dos dados, sendo uma tarefa complicada

para fazer em R ou Java. [McK19]

### 1.2.2 Orientação a objetos

O Python é uma linguagem multiparadigma, suportando diferentes abordagens de programação, um jeito de solucionar problemas de programação é criar objetos, esse método é conhecido como Programação Orientada a Objetos(POO) e a orientação a objetos é um dos paradigmas da linguagem Python, com isso, a criação de objetos e classes é mais simples.

No Python os dados são guardados em objetos, em outras linguagens, determinados tipos são guardados na memória, não em entidades abstratas como objetos. A programação orientada a objetos é um importante método de organizar e desenvolver códigos, focando em criar códigos reutilizáveis. [Lut07]

No Python, a programação orientada a objetos possui alguns pontos importantes, baseados no mesmo autor do parágrafo anterior:

- Encapsulamento: Restrição ao acesso de métodos e atributos de uma classe, evitando que os dados sejam alterados diretamente, na linguagem Python existe apenas o `private` e `public`.
- Métodos: São funções definidas no corpo da classe. Utilizados para definir o comportamento do objeto.
- Objeto: É a instância de uma classe. Somente a definição do objeto é definida no momento que a classe é definida. Consequentemente nenhum espaço de memória é alocado.
- Classe: Juntam as funcionalidades de um certo objeto, servindo como um modelo.
- Polimorfismo: É a capacidade de utilizar uma interface para vários tipos de dados. Possibilitando que o objeto possua o poder de assumir diversas formas.
- Herança: Cria uma classe nova que será descendente e irá utilizar particularidades de outra classe já existente, sem realizar modificações na mesma.

### 1.2.3 Pentest

O Python está entre as melhores linguagens para pentest e segurança da informação, isso é devido a grande quantidade de bibliotecas, ferramentas, frameworks, versatilidade, ser multiplataforma, entre outros pontos que tornam o Python uma das melhores linguagens para essa aplicação, como código de fácil leitura e sintaxe simples. A linguagem é usada para diversas finalidades dentro do processo, possibilitando que as soluções sejam vinculadas e automatizadas sem dificuldades, como decodificar e enviar pacotes, varrer redes e portas, analisar malwares, acessar servidores, com base em [Mor18] e [Sei15].

Como citado, a linguagem Python possui uma grande diversidade de ferramentas, de acordo com os mesmos autores do parágrafo anterior, essas são algumas das ferramentas disponíveis:

- Análise de malware:
  - Exefilter: Utilizada para filtrar tipos de arquivos para páginas web, e-mails e arquivos, podendo detectar diversos tipos de arquivos e apagar o conteúdo ativo.
  - PyClamAV: Acrescenta a detecção de vírus maliciosos nas ferramentas do Python.
  - Pyew: Utilizado geralmente para analisar malwares, o pyew desmonta e edita hexadecimais de linha de comando.
- Utilitários de rede:
  - Dpkt: Utilizado para gerar e analisar pacotes de dados utilizando definições do protocolo TCP/IP.
  - Spoodle: Usado para verificar subdomínios e vulnerabilidade.
  - Knock Subdomain Scan: utilizado para retornar a lista de subdomínios do domínio de destino através da técnica de lista de palavras.
- Explorar bibliotecas:
  - Scapy: Biblioteca e ferramenta de processamento de pacotes, podendo decodificar

diversos protocolos, enviar pela rede, capturar e combinar respostas e solicitações. Oferece funções como as funções oferecidas pelo Tcpdump, Wireshark e Nmap, também oferece acesso programático.

- Python Nmap: Usado para analisar os resultado da varredura do Nmap e lançar ataques particularizados contra hosts específicos.
- Monda: O Monda é um depurador de imunidade, que auxilia no desenvolvimento de programas de exploração.
- Forense
  - Rekall: O Rekall é um framework criado pelo Google para realizar varreduras e análises de memória.
  - LibForensics: É uma biblioteca para desenvolvimento de aplicativos Forenses digitais
  - Aft: É um pacote de ferramentas forenses voltado para Android.





## Referências Bibliográficas

- [Bor14] Luiz Eduardo Borges. *Python para desenvolvedores: aborda Python 3.3*. Novatec Editora, Sao Paulo, SP, Brasil, 2014. Citado na página 5.
- [Lut07] M. Lutz. *Aprendendo Python*. Bookman, Porto Alegre, RS, 2007. Citado 2 vezes nas páginas 6 e 7.
- [McK19] Wes McKinney. *Python para análise de dados*. Novatec Editora, Sao Paulo, SP, May 2019. Citado na página 7.
- [Mor18] Daniel Moreno. *Python para pentest*. Novatec Editora, Sao Paulo, SP, 2018. Citado na página 7.
- [Per16] Ljubomir Perkovic. *Introdução a computação usando Python: um foco no desenvolvimento de aplicações*. Rio de Janeiro, RJ, 2016. Citado na página 5.
- [Sei15] Justin Seitz. *Black Hat Python: Python Programming for Hackers and Pentesters*. No Starch Press, Sao Paulo, SP, 2015. Citado na página 7.



**Disciplina:** *Paradigmas de Linguagens de Programação 1970*

**Linguagem:** *LinguagemXYZabcd*

**Aluno:** *Rômulo Souza Fernandes*

### Ficha de avaliação:

Aspectos de avaliação (requisitos mínimos)	Pontos
<b>Introdução (Máximo: 01 pontos)</b> <ul style="list-style-type: none"> <li>• Aspectos históricos</li> <li>• Áreas de Aplicação da linguagem</li> </ul>	
<b>Elementos básicos da linguagem (Máximo: 01 pontos)</b> <ul style="list-style-type: none"> <li>• Sintaxe (variáveis, constantes, comandos, operações, etc.)</li> <li>• Cada elemento com exemplos (código e execução)</li> </ul>	
<b>Aspectos Avançados da linguagem (Máximo: 2,0 pontos)</b> <ul style="list-style-type: none"> <li>• Sintaxe (variáveis, constantes, comandos, operações, etc.)</li> <li>• Cada elemento com exemplos (código e execução)</li> <li>• Exemplos com fonte diferenciada (listing)</li> </ul>	
<b>Mínimo 5 Aplicações completas - Aplicações (Máximo : 2,0 pontos)</b> <ul style="list-style-type: none"> <li>• Uso de rotinas-funções-procedimentos, E/S formatadas</li> <li>• Uma Calculadora</li> <li>• Gráficos</li> <li>• Algoritmo QuickSort</li> <li>• Outra aplicação</li> <li>• Outras aplicações ...</li> </ul>	
<b>Ferramentas (compiladores, interpretadores, etc.) (Máximo : 1,0 pontos)</b> <ul style="list-style-type: none"> <li>• Ferramentas utilizadas nos exemplos: pelo menos DUAS</li> <li>• Descrição de Ferramentas existentes: máximo 5</li> <li>• Mostrar as telas dos exemplos junto ao compilador-interpretador</li> <li>• Mostrar as telas dos resultados com o uso das ferramentas</li> <li>• Descrição das ferramentas (autor, versão, homepage, tipo, etc.)</li> </ul>	
<b>Organização do trabalho (Máximo: 01 ponto)</b> <ul style="list-style-type: none"> <li>• Conteúdo, Historia, Seções, gráficos, exemplos, conclusões, bibliografia</li> <li>• Cada elemento com exemplos (código e execução, ferramenta, nome do aluno)</li> </ul>	
<b>Uso de Bibliografia (Máximo: 01 ponto)</b> <ul style="list-style-type: none"> <li>• Livros: pelo menos 3</li> <li>• Artigos científicos: pelo menos 3 (IEEE Xplore, ACM Library)</li> <li>• Todas as Referências dentro do texto, tipo [ABC 04]</li> <li>• Evite Referências da Internet</li> </ul>	
<b>Conceito do Professor (Opcional: 01 ponto)</b>	
<p style="text-align: right;"><b>Nota Final do trabalho:</b></p>	

*Observação:* Requisitos mínimos significa a metade dos pontos