



Introdução à Linguagem Python

Paradigmas de Linguagens de Programação

Rômulo Souza Fernandes
Ausberto S. Castro Vera

6 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, Maio 2019



Sumário

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



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.

1.2.2 Orientação a objetos

Programação orientada a objetos é a manipulação de dados, onde os dados são guardados em objetos. O Python é uma linguagem multiparadigma, a orientação a objetos sendo um dos paradigmas da linguagem, pois 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.

Podemos resumir a programação orientada a objetos no Python em 3 pontos, são eles:

- Encapsulamento: Proteção dos métodos e atributos de uma classe, na linguagem Python existe apenas o `private` e o `public`, sendo definidos no próprio nome do método ou atributo.
- Polimorfismo: Uma subclasse pode ter métodos com o mesmo nome que a superclasse, o programa deve saber especificamente qual método chamar. Resumindo, o objeto possui o poder de assumir diversas formas.
- Herança: Estabelece que uma classe pode herdar métodos e atributos de outra classe, reduzindo linhas de códigos repetidos.

1.2.3 Games



Referências Bibliográficas

- [Bor14] Luiz Eduardo Borges. *Python para desenvolvedores: aborda Python 3.3*. Novatec Editora, São Paulo, SP, Brasil, 2014. Citado na página 5.
- [Lut07] M. Lutz. *Aprendendo Python*. Bookman, Porto Alegre, RS, 2007. Citado na página 6.
- [Per16] Ljubomir Perkovic. *Introdução à computação usando Python: um foco no desenvolvimento de aplicações*. Rio de Janeiro, RJ, 2016. Citado na página 5.
- [SH12] Maureen Sprankle and Jim Hubbard. *Problem solving and programming concepts*. Pearson Education, Inc., 9 edition, 2012. Nenhuma citação no texto.

Disciplina: *Paradigmas de Linguagens de Programação 1970*

Linguagem: *LinguagemXYZabcd*

Aluno: *Rômulo Souza Fernandes*

Ficha de avaliação:

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

Observação: Requisitos mínimos significa a metade dos pontos