



Introdução à Linguagem Python

Paradigmas de Linguagens de Programação

Aluno Fulano de tal
Ausberto S. Castro Vera

9 de março de 2020



Copyright © 2020 Aluno Fulano de Tal 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	Aspectos históricos da linguagem Python	5
1.2	Áreas de Aplicação da Linguagem	7
1.2.1	Big Data	7
1.2.2	Orientação a objetos	7
1.2.3	outras	7
2	Conceitos básicos da Linguagem Python	9
2.1	Variáveis e constantes	9
2.2	Tipos de Dados Básicos	9
2.2.1	String	9
2.3	Tipos de Dados de Coleção	10
2.3.1	Tipos Sequenciais	10
2.3.2	Tipos Conjunto	10
2.3.3	Tipos Mapeamento	10
2.4	Estrutura de Controle e Funções	10
2.4.1	O comando IF	10
2.4.2	Laço FOR	10
2.4.3	Laço WHILE	10
2.5	Módulos e pacotes	10
2.5.1	Módulos	10
2.5.2	Pacotes	10
3	Programação Orientada a Objetos com Python	13
3.1	Classes e Objetos	13

3.2	Operadores ou Métodos	13
3.3	Herança	13
3.4	Estudo de Caso:	13
4	Aplicações da Linguagem Python	15
4.1	Operações básicas	15
4.2	Programas gráficos	15
4.3	Programas com Objetos	15
4.4	O algoritmo Quicksort - Implementação	15
4.5	Aplicações com Banco de Dados	15
5	Ferramentas existentes e utilizadas	17
5.1	Editor MNOP	17
5.2	Compilador XYZ	17
5.3	Interpretador UVW	17
5.4	Ambientes de Programação IDE MNP	17
6	Conclusões	19
	Bibliografia	21
	Index	23



1. Introdução

Python é uma poderosa linguagem de programação de alto nível e orientada a objetos, originalmente conceitualizada por Guido van Rossum, no final dos anos 1980, no National Research Institute of Mathematics and Computer Science, Holanda. Python foi a sucessora da linguagem ABC. Python é uma linguagem de uso geral, orientada a objetos, com código bastante legível, e com muitas bibliotecas disponíveis e amplamente conhecidas (NumPy, SciPy, Pandas, IPython, Matplotlib, mIPy, ScraPy, etc.)

Python, uma linguagem de script de código aberto, se tornou a linguagem de ensino introdutória mais popular nas principais universidades americanas - entre elas, Georgia Tech - segundo uma pesquisa recente de Philip Guo, professor assistente de ciência da computação na Universidade de Rochester. Guo decidiu conduzir a pesquisa depois de notar, nos últimos anos, que o Python estava substituindo linguagens como Java como a introdução de fato à classe de programação em mais e mais aulas de ciência da computação em universidades de todo o país. [She15]

1.1 Aspectos históricos da linguagem Python

A história da maioria de linguagens de programação não tem uma data fixa, nem um autor único. A sua evolução inclui muitos personagens, muitas instituições e muitas versões.

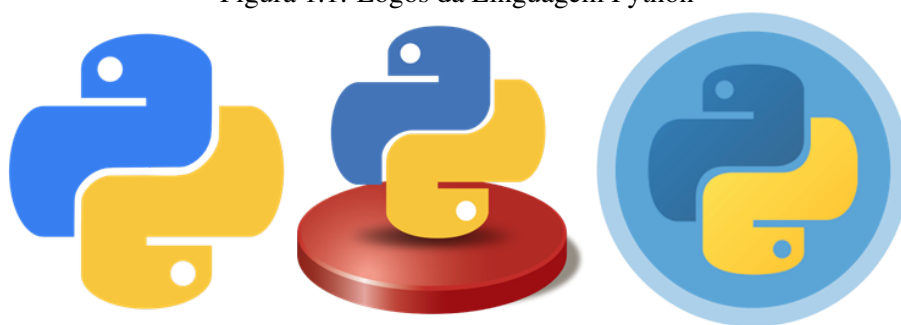
A seguir, menciona-se alguns aspectos históricos da linguagem Python, baseados em [Per16], [Sev15] :

- O autor principal foi o holandês Guido van Rossum, quando trabalhava no CWI (Centrum Wiskunde & Informatica), em Amsterdã, Holanda.
- A linguagem não recebeu esse nome por causada espécie de serpente, mas sim do seriado de comédia da BBC *Monty Python's Flying Circus* da qual Rossum é um fã.
- Python 0.9.0 foi lançado em 1991. Esta versão incluía manipulação de exceções, classes, listas e strings. Também foram incluídos alguns aspectos de programação funcional: lambda, maps, filtros e reduce.
-

- Em 1995, Guido continuou seu trabalho sobre Python na Corporation for National Research Initiatives (CNRI) em Reston, Virginia, USA.
- Python 1.6 foi lançado no CNRI em
- No ano 2000, Guido e a equipe de desenvolvimento principal do Python foram para BeOpen.com para formar a equipe BeOpen PythonLabs.
- Python 2.0 foi lançado no ano 2000
- Python 3.0 foi lançado em dezembro de 2008
-
-

Algumas linguagens são consideradas tradicionais (como mostrado na Fig.1.1) e outras são consideradas modernas (ver Fig.6.1). Devemos observar aqui, que a inclusão de qualquer figura, significa que ela deve ser referenciada em algum lugar do texto

Figura 1.1: Logos da Linguagem Python



Fonte: O autor deste trabalho

Algumas figuras são criadas e elaboradas pelo mesmo autor, neste caso, deve-se escrever como fonte "O autor", "Os autores", etc. Figuras que incluam imagens de outras fontes deve-se especificar claramente, indicando o link ou referência correspondente, por exemplo, uma imagem que aparece em [SH12, p. 93], é mostrada na Fig.1.2 e a fonte deve ser indicada na parte inferior da figura.

Figura 1.2: Algoritmo, Diagrama de fluxo, e Pseudo-código

Algorithm	Flowchart	Pseudocode
<p><i>NameAge</i></p> <ol style="list-style-type: none"> 1. <i>Enter Name, Age</i> 2. <i>Print Name, Age</i> 3. <i>End</i> 	<pre> graph TD Start([NameAge]) --> Input[/Enter Name, Age/] Input --> Output[/Print Name, Age/] Output --> End([End]) </pre>	<p><i>Enter Name, Age</i> <i>Print Name, Age</i> <i>End</i></p>

Fonte: [SH12, p. 93]

1.2 Áreas de Aplicação da Linguagem

Esta linguagem é utilizada e aplicada nas seguintes áreas: !!!!! As aqui mostradas são exemplos!!!

1.2.1 Big Data

Fazer uma breve descrição. Pelo menos 3 parágrafos mencionando exemplos

1.2.2 Orientação a objetos

Fazer uma breve descrição. Pelo menos 3 parágrafos mencionando exemplos

1.2.3 outras



2. Conceitos básicos da Linguagem Python

Os livros básicos para o estudo da Linguagem Python são: [Sum13], [Gut15], [Per16]

Neste capítulo é apresentado

Segundo [Seb11], a linguagem Python, . . .

De acordo com [Seb11] e [RH04], a linguagem Python . . .

[Seb11] afirma que a linguagem Python . . .

Considerando que a linguagem Python ([Seb11], [Wat90]) é considerada como

2.1 Variáveis e constantes

2.2 Tipos de Dados Básicos

2.2.1 String

Um string é uma sequência de caracteres considerado como um item de dado simples. Para Python, um string é um array de caracteres ou qualquer grupo de caracteres escritos entre dobre aspas ou aspas simples. Por exemplo,

```
>>> #usando aspas simples
>>> pyStr1 = 'Brasil'
>>> print (pyStr1)  Brasil
>>> #usando aspas duplas
>>> pyStr2 = "Oi, tudo bem?"
>>> print (pyStr2)
Oi, tudo bem?
```

- *Concatenação de strings*

Strings podem ser concatenadas utilizando o operador +, e o seu comprimento pode ser calculado utilizando o operador `len(string)`

```
>>> # concatenando 2 strings
>>> pyStr = "Brasil" + " verde amarelo"
>>> print (pyStr)
```

```
Brasil verde amarelo
>>> print (len(pyStr))
20
```

- *Operador de indexação*

Qualquer caracter de um string ou sequência de caracteres pode ser obtido utilizando o operador de indexação []. Existem duas formas de indexar em Python, os caracteres de um string:

Index com inteiros positivos indexando a partir da esquerda começando com 0 e onde 0 é o index do primeiro caracter da sequência

Index com inteiros negativos indexando a partir da direita começando com -1, e onde -1 é o último elemento da sequência, -2 é o penúltimo elemento da sequência, e assim sucessivamente.

```
>>> # Indexando strings
>>> pyStr = "Programando"
>>> print (len(pyStr))
11
>>> print (pyStr)
Brasil verde amarelo
```

- *Operador de Fatias*

O operador de acesso a itens (caracteres individuais) também pode ser utilizado como operador de fatias, para extrair uma fatia inteira (subsequência) de caracteres de um string. O operador de Fatias possui três sintaxes:

```
seq[ inicio ]
seq[ inicio : fim ]
seq[ início : fim : step ]
onde início, fim e step são números inteiros.
```

```
>>> # Indexando strings
>>> pyStr = "Programando Python"
>>> print (len(pyStr))
11
>>> print (pyStr)
Brasil verde amarelo
```

2.3 Tipos de Dados de Coleção

2.3.1 Tipos Sequenciais

2.3.2 Tipos Conjunto

2.3.3 Tipos Mapeamento

2.4 Estrutura de Controle e Funções

2.4.1 O comando IF

2.4.2 Laço FOR

2.4.3 Laço WHILE

2.5 Módulos e pacotes

2.5.1 Módulos

2.5.2 Pacotes

Código fonte para a linguagem Python:

```
number_1 = int(input('Ingresse o primeiro numero: '))
number_2 = int(input('Ingresse o segundo numero: '))

# Soma
print('{} + {} = '.format(number_1, number_2))
print(number_1 + number_2)

# Subtra\c{c}\~{a}o
print('{} - {} = '.format(number_1, number_2))
print(number_1 - number_2)

# Multiplica\c{c}\~{a}o
print('{} * {} = '.format(number_1, number_2))
print(number_1 * number_2)

# Divis\~{a}o
print('{} / {} = '.format(number_1, number_2))
print(number_1 / number_2)
```




3. Programação Orientada a Objetos com Python

3.1 Classes e Objetos

```
class NomeClasse:  
  
    def metodo1:  
  
    def Metodo2:
```

3.2 Operadores ou Métodos

3.3 Herança

3.4 Estudo de Caso:



4. Aplicações da Linguagem Python

Devem ser mostradas pelo menos CINCO aplicações completas da linguagem, e em cada caso deve ser apresentado:

- Uma breve descrição da aplicação
- O código completo da aplicação,
- Imagens do código fonte no compilador-interpretador,
- Imagens dos resultados após a compilação-interpretação do código fonte
- Links e referencias bibliográficas de onde foi obtido a aplicação

4.1 Operações básicas

4.2 Programas gráficos

4.3 Programas com Objetos

4.4 O algoritmo Quicksort - Implementação

4.5 Aplicações com Banco de Dados



5. Ferramentas existentes e utilizadas

Neste capítulo devem ser apresentadas pelo menos DUAS (e no máximo 5) ferramentas consultadas e utilizadas para realizar o trabalho, e usar nas aplicações. Considere em cada caso:

- Nome da ferramenta (compilador-interpretador)
- Endereço na Internet
- Versão atual e utilizada
- Descrição simples (máx 2 parágrafos)
- Telas capturadas da ferramenta
- Outras informações

5.1 Editor MNOP

5.2 Compilador XYZ

5.3 Interpretador UVW

5.4 Ambientes de Programação IDE MNP



6. Conclusões

Os problemas enfrentados neste trabalho ...

O trabalho que foi desenvolvido em forma resumida ...

Aspectos não considerados que poderiam ser estudados ou úteis para ...

Figura 6.1: Linguagens de programação modernas



Fonte: O autor



Referências Bibliográficas

- [Gut15] John V. Guttag. *Introdução à Computação e Programação Usando Python*. Infopress Nova Mídia, São Paulo, edição revista e ampliada edition, 2015. Citado na página 9.
- [Per16] Ljubomir Perkovic. *Introdução à Computação usando Python: um foco no desenvolvimento de Aplicações*. LTC Livros Técnicos e Científicos Editora Ltda, Rio de Janeiro, 2016. Citado 2 vezes nas páginas 5 e 9.
- [RH04] Peter Van Roy and Seif Haridi. *Concepts, Techniques and Models of Computer Programming*. The MIT Press, Cambridge, 2004. Citado na página 9.
- [Seb11] Robert W. Sebesta. *Conceitos de Linguagens de Programação*. Bookman, Porto Alegre, 2011. Citado na página 9.
- [Sev15] C. Severance. Guido van Rossum: The Early Years of Python. *Computer*, 48(2):7–9, Feb 2015. Citado na página 5.
- [SH12] Maureen Sprankle and Jim Hubbard. *Problem solving and programming concepts*. Pearson Education, Inc., 9 edition, 2012. Citado na página 6.
- [She15] Esther Shein. Python for beginners. *Commun. ACM*, 58(3):19–21, February 2015. Citado na página 5.
- [Sum13] Mark Summerfield. *Programação em Python 3 - Uma Introdução Completa à Linguagem Python*. Biblioteca do Programador. Alta Books Editora, Rio de Janeiro, 2013. Citado na página 9.
- [Wat90] David Anthony Watt. *Programming Language Concepts and Paradigms*. Prentice Hall International, London, 1990. Citado na página 9.

Disciplina: Paradigmas de Linguagens de Programação 2019

Linguagem: Linguagem Python

Aluno: Nome Completo do aluno

Ficha de avaliação:

Aspectos de avaliação (requisitos mínimos)	Pontos
Elementos básicos da linguagem (Máximo: 01 pontos) <ul style="list-style-type: none"> • Sintaxe (variáveis, constantes, comandos, operações, etc.) • Usos e áreas de Aplicação da Linguagem 	
Cada elemento da linguagem (definição) com exemplos (Máximo: 02 pontos) <ul style="list-style-type: none"> • Exemplos com fonte diferenciada (Courier , 10 pts, azul) 	
Mínimo 5 exemplos completos - Aplicações (Máximo : 2 pontos) <ul style="list-style-type: none"> • Uso de rotinas-funções-procedimentos, E/S formatadas • Menu de operações, programas gráficos, matrizes, aplicações 	
Ferramentas (compiladores, interpretadores, etc.) (Máximo : 2 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 obtidos nas 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 	
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)	
Nota Final do trabalho:	

Observação: Requisitos mínimos significa a *metade* dos pontos