

Introdução à Linguagem Python

João Gabriel Rocha Silva

Programa de Pós-Graduação em Modelagem Computacional
Universidade Federal de Juiz de Fora

Conteúdo

- Contextualização
 - História
 - Características
- Sintaxe Básica
 - Declarações de variáveis, estruturas de repetição e funções
 - Orientação a objetos
- Bibliotecas Matemáticas
- Gráficos 2D
- Aplicações
- Conclusão e Referências



Conteúdo

- **Contextualização**
 - **História**
 - **Características**
- **Sintaxe Básica**
 - Declarações de variáveis, estruturas de repetição e funções
 - Orientação a objetos
- **Bibliotecas Matemáticas**
- **Gráficos 2D**
- **Aplicações**
- **Conclusão e Referências**



Contextualização

- **HISTÓRIA**

- 1989, Instituto Nacional de Pesquisa em Ciência da Computação e Matemática (CWI, Países Baixos).
- Tratamento de exceção e produção de interface para o Sistema “Amoeba” por scripts.
- Gerando produtividade do programador.
- Constantes evoluções.
- Orientação a objetos (1991) e bibliotecas (constante).

Contextualização

- **CARACTERÍSTICAS**

- Produzida com base na linguagem ABC, com derivações da sintaxe da linguagem C.
- Alto Nível
- Interpretada
- De Script
- Tipagem Dinâmica
- Legibilidade, facilidade e identificação
- Material na internet

- **DESVANTAGENS**

- Não se comporta bem com a utilização de threads
- Difícil debug (principalmente por conflitos de tipos).



Conteúdo

- Contextualização
 - História
 - Características
- **Sintaxe Básica**
 - **Declarações de variáveis, estruturas de repetição e funções**
 - **Orientação a objetos**
- Bibliotecas Matemáticas
- Gráficos 2D
- Aplicações
- Conclusão e Referências



Sintaxe Básica

- **DECLARAÇÃO DE VARIÁVEIS**

`a = 10`

`b = 'galo'`

`c = 5.433`

`>> print a`

`>> 10`

`>> type(a)`

`< type = 'int' >`

Sintaxe Básica

- **OPERADORES LÓGICOS**

C	Python
&&	and
 	or
!	not

- **LISTAS**

```
lista1 = []
```

```
lista2 = [1, 3, 5, 7, 9]
```


Sintaxe Básica

- **MATRIZES (Concatenação de listas)**

```
→ 1 A = []  
2 for i in range(5):  
3     A.append( [0] * 5 )  
4 A[1][1] = 2
```

Sintaxe Básica

- **ESTRUTURA DE REPETIÇÃO FOR**

Exemplo 1:

```
for i in range (10):
```

```
    print i
```

0 1 2 3 4 5 6 7 8 9

Exemplo 2:

```
lista = ['Tamara', 'Nielson', 'Guilherme']
```

```
for i in lista:
```

```
    print i
```

Tamara Nielson Guilherme

Sintaxe Básica

- **ORIENTAÇÃO A OBJETOS**
 - Classes, métodos e objetos

```
#calculadora.py
class Calculadora(object):

    def __init__(self, a, b):
        self.a = a
        self.b = b

    def soma(self):
        return self.a + self.b

    def subtrai(self):
        return self.a - self.b

    def multiplica(self):
        return self.a * self.b

    def divide(self):
        return self.a / self.b
```

Sintaxe Básica

- **ORIENTAÇÃO A OBJETOS**

```
$ python3
>>> from calculadora import Calculadora
>>> c = Calculadora(128,2)
>>> print('Soma:', c.soma())
>>> print('Subtração:', c.subtrai())
>>> print('Multiplicação:', c.multiplica())
>>> print('Divisão:', c.divide())
```

```
>>> c.a = 12
>>> c.b = 42
>>> print c.soma()
```

Conteúdo

- Contextualização
 - História
 - Características
- Sintaxe Básica
 - Declarações de variáveis, estruturas de repetição e funções
 - Orientação a objetos
- **Bibliotecas Matemáticas**
- Gráficos 2D
- Aplicações
- Conclusão e Referências



Bibliotecas Matemáticas

- **BIBLIOTECA MATH**
 - Responsável por facilitar operações matemáticas.

No início do código acionar a biblioteca:

```
import math
```

A partir deste momento qualquer função contida nesta biblioteca podem ser utilizadas e realizadas pelo programa.

Bibliotecas Matemáticas

- **BIBLIOTECA MATH**

- Exemplos de utilização:

math.sqrt(num): Retorna a raiz quadrada do número.

math.cos(num): Retorna o cosseno do número em radiano.

math.sin(num): Retorna o seno do número em radiano.

math.tan(num): Retorna a tangente do número em radiano.

math.radians(num): Converte o angulo 'numero' de graus para radiano.

math.pi: Constante com o número pi (3.1415926535897931).

math.hypot(x, y): Retorna a hipotenusa dos números (catetos) fornecido

Bibliotecas Matemáticas

- **BIBLIOTECA RANDOM**

- Responsável por gerar números aleatórios.

No início do código acionar a biblioteca:

```
import random
```

A partir deste momento qualquer função contida nesta biblioteca podem ser utilizadas e realizadas pelo programa.

Bibliotecas Matemáticas

- **BIBLIOTECA RANDOM**

- Exemplos de utilização:

random.randint(a, b): Retorna um número inteiro entre a e b.

`times = [Atlético Mineiro, Cruzeiro, Flamengo, Tupi]`

random.choice(times): Retorna um elemento da lista.

random.gauss(m, sigma): Retorna a distribuição de Gauss para estes parâmetros.

random.normalvariate(m, sigma): Retorna a distribuição Normal para estes parâmetros.

Conteúdo

- Contextualização
 - História
 - Características
- Sintaxe Básica
 - Declarações de variáveis, estruturas de repetição e funções
 - Orientação a objetos
- Bibliotecas Matemáticas
- **Gráficos 2D**
- Aplicações
- Conclusão e Referências



Gráficos 2D

- **BIBLIOTECA MATPLOTLIB**
 - Responsável por funções gráficas.

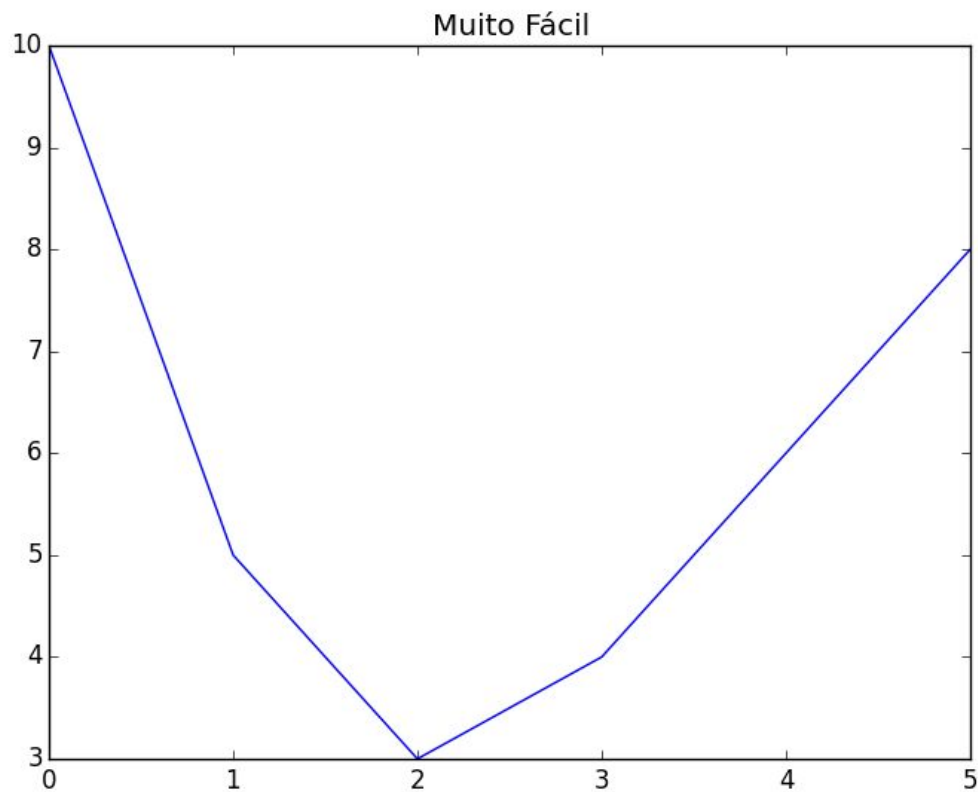
No início do código acionar a biblioteca:

```
import matplotlib.pyplot as plt
```

A partir deste momento qualquer função contida nesta biblioteca podem ser utilizadas e realizadas pelo programa.

Gráficos 2D

```
import matplotlib.pyplot as plt
plt.plot( [10,5,3,4,6,8] )
plt.title("Muito Fácil")
plt.show()
```



Gráficos 2D

```
import numpy as np
import matplotlib.pyplot as plt

data1 = [10,5,2,4,6,8]
data2 = [ 1,2,4,8,7,4]
x = 10*np.array(range(len(data1)))

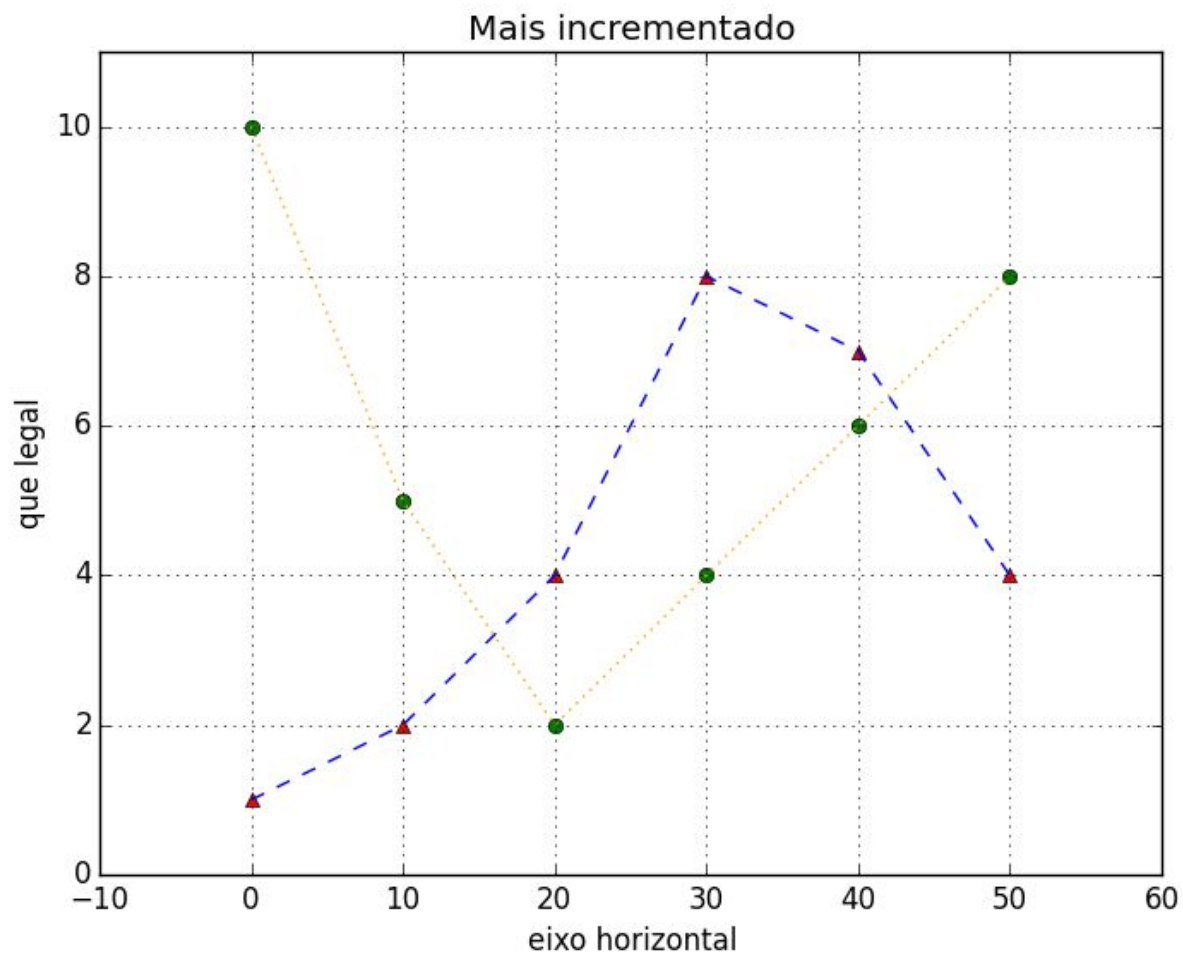
plt.plot( x, data1, 'go') # green bolinha
plt.plot( x, data1, 'k:', color='orange') # linha pontilha orange

plt.plot( x, data2, 'r^') # red triangulo
plt.plot( x, data2, 'k--', color='blue') # linha tracejada azul

plt.axis([-10, 60, 0, 11])
plt.title("Mais incrementado")

plt.grid(True)
plt.xlabel("eixo horizontal")
plt.ylabel("que legal")
plt.show()
```

Gráficos 2D



Conteúdo

- Contextualização
 - História
 - Características
- Sintaxe Básica
 - Declarações de variáveis, estruturas de repetição e funções
 - Orientação a objetos
- Bibliotecas Matemáticas
- Gráficos 2D
- **Aplicações**
- Conclusão e Referências



Aplicações

- **Scripts** para programas que dependem de várias execuções para garantir a confiabilidade.
- **Sites** e até mesmo **sistemas**, principalmente com a conexão com **Ruby on Rails**.
- Aplicações para pesquisas **matemáticas**.
- Desenvolvimento de jogos (**Pygames**).

Aplicações

- JOGOS



Aplicações

- **EMPRESAS QUE POSSUEM APLICAÇÕES REALIZADAS EM PYTHON:**



YAHOO!



Google

Conteúdo

- Contextualização
 - História
 - Características
- Sintaxe Básica
 - Declarações de variáveis, estruturas de repetição e funções
 - Orientação a objetos
- Bibliotecas Matemáticas
- Gráficos 2D
- Aplicações
- Conclusão e Referências



Conclusão

- Python é uma linguagem com vasta área de aplicação.
- Fácil programação e legibilidade.
- Linguagem dinâmica e de fácil migração.



Referências

- Griffiths, Dawn, and David Griffiths. *Use a Cabeça! Python*. Alta Books Editora, 2013.
- Menezes, Nilo Ney Coutinho. "Introdução à Programação com Python." *Editora Novatec, São Paulo, SP* (2010).
- Lutz, Mark. *Learning python*. " O'Reilly Media, Inc.", 2013.
- Python Brasil <Disponível em <http://wiki.python.org.br/>. Acesso em 06/04/2016>
- Python Club <Disponível em <http://pythonclub.com.br/>. Acesso em 06/04/2016>

Obrigado pela atenção!

Dúvidas ou comentários?



Introdução à Linguagem Python

João Gabriel Rocha Silva

joaogabriel.comp@gmail.com