FisicaGeral_python_1

March 13, 2019

1 Python

Python é uma linguagem de programação de propósito geral.

É uma das linguagens de programação mais usadas no mundo atualmente.

- É intuitiva e de rápida aprendizagem.
- Integra uma vasta biblioteca para computação científica e análise de dados: matplotlib, NumPy, pandas, SciPy, statsmodel, scikit-learn, etc.
- É "portátil": o código escrito em Python pode ser processado em qualquer sistema ou plataforma.
- Como Python é uma linguagem *interpretada*, código escrito em Python será consideravelmente mais lento em comparação com um linguagem *compilada*.
- Este problema pode ser solucionado ou reduzido utilizando-se bibliotecas e extensões que são por sua vez escritas em linguagens como C, C++ ou FORTRAN. Python oferece suporte para a integração destas bibliotecas.

Você também pode usar python no seu celular ou tablet através de aplicativos como: + QPython3 + Kivy

Nessa aula, aprenderemos a: + Definir e usar variáveis e objetos em Python. + Escrever um script simples em python fazendo o uso de funções, listas, dicionários, etc. + Usar ferramentas para visualizar um histograma (usando o pacote matplolib).

Depois de assistir a essa aula, você pode aprender sozinhx a: + Procurar por informação sobre Python + Escrever Módulos de python + Fazer tudo o que quiser com Python e não querer usar mais nada! (ou talvez algumas linguagens complementares)

1.1 O Criador

Guido van Rossum

1.2 O Zen do Python

```
In [2]: import this
The Zen of Python, by Tim Peters
Beautiful is better than ugly.
Explicit is better than implicit.
```



title

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one-- and preferably only one --obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than *right* now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea -- let's do more of those!

1.3 O que é Python?

Python: Linguagem de programação de alto nível.

Com regras de sintaxe para escrever o que é considerado um código de Python válido. Interpretador de Python: https://www.python.org/

Lê o código fonte (escrito em Python) e executa as instruções. $<\!$ ul>

Python para iniciantes sem experiência prévia em programação:



title

```
 https://automatetheboringstuff.com/
 http://openbookproject.net/thinkcs/python/english3e/
```

1.4 Como instalar Python?

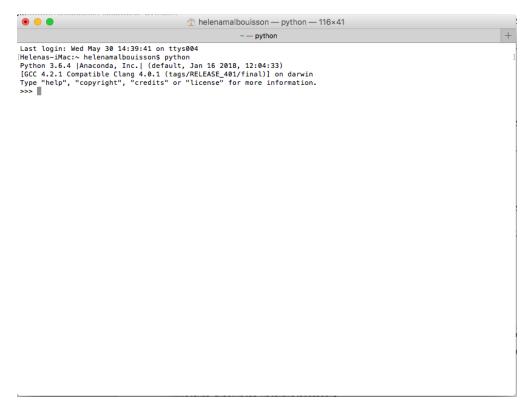
- Recomendação: Instalar Anaconda https://docs.anaconda.com/anaconda/install/
- Inclui IPython (https://ipython.org/index.html) e o Jupyter Notebook (https://jupyter.org/).
- Muitos outros ambientes de desenvolvimento (IDEs), de acordo com a preferência.

1.5 Interpretador de Python

• Modo *prompt* e modo *script*.

1.6 Interpretador de Python (Prompt)

- Abrir um terminal de Linux (com Python previamente instalado)
- python
- para sair: ctrl + D
- Para Windows, você pode usar um dos ambientes de desenvolvimento (IDE).



title

1.7 Python como uma calculadora

1.8 Tipo de dados de Python

valor: algo fundamental manipulado pelo programa

```
letra número
```

tipo de dados: classificação dos valores

```
letra: <i>string</i>número: <i>inteiro, float, bool</i>
```

1.9 Tipos de dados de Python

- tuplas
- listas
- dicionários
- tipos de dados built-in: https://docs.python.org/3/library/stdtypes.html
- tipos de dados especializados: https://docs.python.org/3/library/datatypes.html

```
In [9]: # Tipos numéricos: inteiros e float
       val = 12345
        val
Out[9]: 12345
In []:
In []:
In [6]: # A tupla é ordenada. Sintaxe: valores entre parênteses, separados por vírgulas.
        # A tupla é imutável.
       tupla = (1, 8, 10)
        print("isto é uma tupla: ", tupla)
isto é uma tupla: (1, 8, 10)
In [7]: # A lista É ORDENADA. Sintaxe: valores entre colchetes, separados por vírgulas.
        # A lista é mutável
        lista = [1, 8, 10]
        print("isto é uma lista: ", lista)
isto é uma lista: [1, 8, 10]
In []: # loop for: para cada item da lista, fazer o comando dentro do loop.
        # a indentação é a estrutura absolutamente fundamental em python!
        for item in lista:
            print("este é um item da lista: {item_lista}".format(item_lista=item))
In []: for number in range(10):
            print(number)
In []: soma = 0
        for numero in range(10):
            soma += numero
            print(soma)
```

- atom

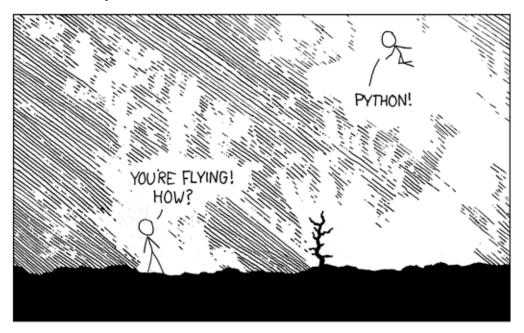
```
import statistics, math #bibliotecas usadas para o calculo de medias e desvio padrao por
# bibliotecas para fazer histograma
import matplotlib.pyplot as plt # matplotlib é uma biblioteca de python para produzir fi
import numpy as np
from matplotlib import colors
def main(arguments):
```

```
# Nome do arquivo de entrada
input_file = '/Users/helenamalbouisson/cernbox/Work/UERJ/Aulas/PYTHON/FisicaGeral_20
# Abre o arquivo de entrada no modo de leitura
ifile = open(input_file, 'r')
# Percorre o arquivo de entrada linha por linha e
# salva cada coluna do arquivo em listas
```

```
idade = []
    altura = []
    massa = []
    for line in ifile:
        columns = line.split()
        idade.append(eval(columns[0]))
        altura.append(eval(columns[1]))
        massa.append(eval(columns[2]))
    # calcula a média de cada uma das colunas do arquivo de entrada:
    # idade, altura e massa:
    print ('media idade = ', media(idade), ' mean(idade): ', statistics.mean(idade), '
    print ('desvio padrao idade: ', sstddev(idade), ' stats.stdev(idade): ', statistics
    print ('media altura = ', media(altura), ' mean(altura): ', statistics.mean(altura)
   print ('media massa = ', media(massa), ' mean(massa): ', statistics.mean(massa), '
    # histogramas
    fig, axs = plt.subplots(1, 3) # cria uma figura com três histogramas
    # define a classe de frequência, título dos eixos e do histograma na posição O
    axs[0].hist(idade, bins=30, range=(16,46), color='red')
    axs[0].set_title('idade')
    axs[0].set_xlabel('idade (anos)')
    axs[0].set_ylabel('frequencia/ano')
    # define a classe de frequência, título dos eixos e do histograma na posição 1
    axs[1].hist(altura, bins=8, range=(1.6,2.0), color='green')
    axs[1].set_title('altura')
    axs[1].set_xlabel('altura (m)')
    axs[1].set_ylabel('frequencia/0.05 m')
    # define a classe de frequência, título dos eixos e do histograma na posição 2
    axs[2].hist(massa, bins=8, range=(40,120), color='grey')
    axs[2].set_title('massa')
    axs[2].set_xlabel('massa (Kg)')
    axs[2].set_ylabel('frequencia/10 Kg')
    plt.show()
    #plot_hist(idade, 30, (16,46), 'red')
    #plot_hist(altura, 40, (1.6,2.0), 'green')
    #plot_hist(massa, 80, (40, 120), 'qrey')
def media(input_list):
    # essa funcao calcula a media dos itens da lista input_list
    \#\#sum\_num = 0
    ##for i in input_list:
        \#\#sum\_num = sum\_num + i
```

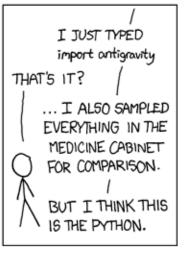
```
#return sum_num/len(input_list)
    return sum(input_list)/len(input_list)
def sstddev(input_list):
    # cacula o desvio padrão amostral
    # usa a biblioteca statistics do python
    # https://docs.python.org/3/library/statistics.html
    N = len(input_list)
    m = statistics.mean(input_list)
    variance = 0
    for i in input_list:
        variance = variance + pow((i - m), 2)
    variance = variance/N
    sample_stddev = math.sqrt((N/(N - 1))*variance)
    return sample_stddev
def plot_hist(input_list, n_bins, hist_range, cor):
    # essa função pega como input uma lista
    # e faz um histograma usando bibliotecas do
    # matplotlib: https://matplotlib.org/api/_as_gen/matplotlib.pyplot.hist.html
    # outras opções de módulos que to fazem histogramas são:
    # plotly, rootpy.
    fig, ax = plt.subplots()
    ax.hist(input_list, bins=n_bins, range=hist_range, color=cor)
    plt.show()
    ##n_bins = [16 + i for i in range(30)]
    ##fiq, ax = plt.subplots()
    ##ax.hist(idade, bins=n_bins, color='red')
    ##plt.show()
if __name__ == '__main__':
    sys.exit(main(sys.argv[1:]))
```

1.11 Feliz Python!









https://xkcd.com/353/

In []: