UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA - ITEC
CURSO DE ENGENHARIA ELÉTRICA

Apostila do Curso de introdução ao Python3

BELÉM - PA
2019

Sumário

1. Apresentação

Este material foi desenvolvido para apoiar o curso de introdução a Python3 ministrado na semana do ITEC. O material é um guia introdutório sobre a linguagem e suas aplicações para visualização de dados, manipulação de matrizes e vetores e métodos numéricos.

O material estará disponibilizado no Github, assim como os códigos que compuseram o curso. Dessa forma sendo disponibilizado para o acesso democrático de todos.

Para sanar duvidas relativas ao curso, ou ao material, ou em relação a linguagem Python e alguma aplicação, estará disponível para contato os e-mails dos ministrantes do curso.

Caso haja algum comentário, duvida ou opinião sobre o curso ou o material, mande por e-mail ! Nós agradeceríamos o feedback da comunidade do ITEC.

1.1. Contatos e links

- Contato: Rodrigo Gomes Dutra

  e-mail: dutra.rgdgd@gmail.com

  github: https://github.com/rodgdutra

- Contato: Antonio Adrian

  e-mail: dutra.rgdgd@gmail.com

  github: https://github.com/rodgdutra

- Link do material do curso e dos códigos:

https://github.com/rodgdutra/minicurso_python

2. Introdução

Este material foi desenvolvido para apoiar o curso de introdução a Python3 ministrado na semana do ITEC. O material é um guia introdutório sobre a linguagem e suas aplicações para visualização de dados, manipulação de matrizes e vetores e métodos numéricos.

Python é uma linguagem de propósito geral de altíssimo nível (VHLL – Very high Level Language), criada pelo holandês Guido Van Rossum seguindo o idel de “Programação de Computadores para todos”. Esta linguagem é multiparadigma, suporta o paradigma orientado a objetos, imperativo, funcional e procedural. Possui tipagem dinâmica e uma de suas principais características é permitir a fácil leitura do código e exigir poucas linhas de código se comparado ao mesmo programa em outras linguagens. Devido às suas características, ela é principalmente utilizada para processamento de textos, dados científicos e criação de CGIs para páginas dinâmicas para a web. Foi considerada pelo público a 3ª linguagem "mais amada", de acordo com uma pesquisa conduzida pelo site Stack Overflow em 2018,e está entre as 5 linguagens mais populares, de acordo com uma pesquisa conduzida pela RedMonk.

A linguagem foi projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional. Prioriza a legibilidade do código sobre a velocidade ou expressividade. Combina uma sintaxe concisa e clara com os recursos poderosos de sua biblioteca padrão e por módulos e *frameworks* desenvolvidos por terceiros.

As Aplicações de python vão desde o backend até ciência de dados(data science) e machine learning. A Imagem abaixo, segundo o site https://data-flair.training/blogs/python-applications/, ilustra as principais aplicações do python3 até então:

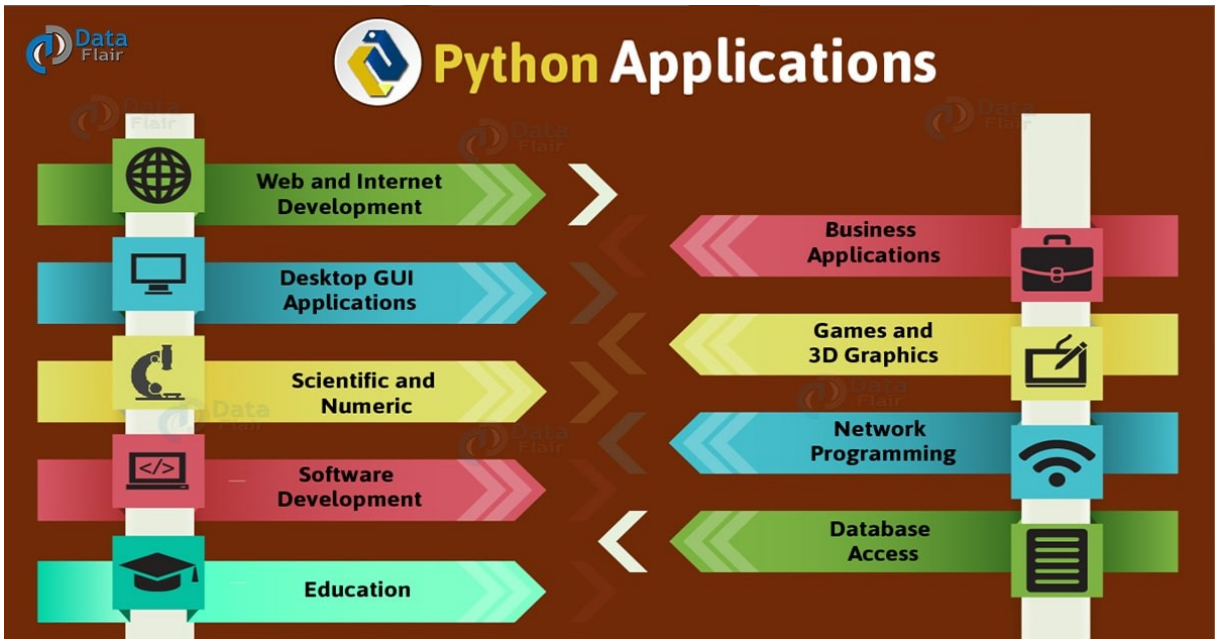

Figura. 1. Real World Applications of Python Programming.

Além dessas aplicações, ainda há aplicações em IOT, como na programação de scripts para raspberry Pi.

Baseado no indice de TIOBE, o python já consta como a terceira linguagem de programação mais utilizada.

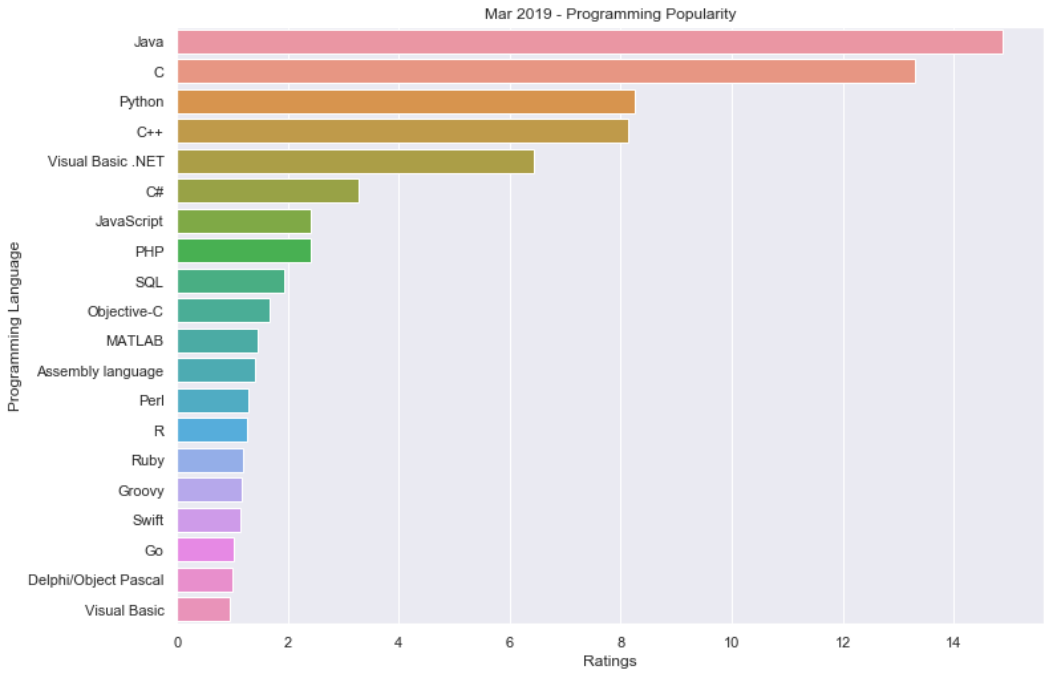

Figura. 2. Linguagens de programação mais utilizadas.

Dessa forma, além do Python ser uma linguagem de programação multiparadigma, é uma linguagem estremamente utilizada no mundo todo com diversas aplicações do “ Mundo real “, assim não faltam motivos para estudar essa linguagem e aplica-la de diversas maneiras.

3. Instalação de um ambiente de desenvolvimento Python

Para instalar o Python3 é recomendado baixar o pacote de instalação direto do link: https://www.python.org/downloads/

Após a instalação do Python3 é recomendado um conjunto de ferramentas para o desenvolvimento com a linguagem. A primeira dessas ferramentas é o Spyder, o qual possui uma interface que lembra a do matlab e possibilita a utilização do console python juntamente com a possibilidade de rodar e escrever scripts, realizar o processo de debug e visualizar variáveis.

Outra ferramenta interessante é o Jupyter notebook, que possibilita ao usuário escrever o codigo, roda-lo e visualizar gráficos na mesma página, como um caderno de anotações. Além disso os notebooks com os gráficos já gerados podem ser salvos, dessa forma para visualização posterior o gráfico já estará plotado na pagina do notebook.

Ambas as ferramentas são instaladas no windows pelo pacote Anaconda, no link: https://www.anaconda.com/distribution/.

Com o Anaconda instalado o usuário tem acesso as ferramentas listadas e conjuntos de bibliotecas que serão abordados nesse curso para manipulação de dados e visualização de dados, as quais são o numpy e a matplotlib.

4. Primeiros passos com o Python

Após a instalação do ambiente de desenvolvimento, é necessário a visualização do terminal python como um primeiro passo do usuário que aspira o desenvolvimento no mesmo. Para isso vá no Iniciar e procure por python3, e abra o terminal do python3. Com isso você já pode realizar pequenas operações com o terminal e testar a sintaxe.

5. Primeiros passos com o Numpy

O módulo numpy é uma das bibliotecas preferidas para trabalhar com matrizes, vetores contendo várias funções para análises numéricas de forma computacionalmente otimizada.

5.1. Primeiro código: Alocando vetores

```
# Importando o numpy

import numpy as np

# Alocando um vetor de 0 até 45 com intervalos de 5 em 5
x1 = np.arange(0,50,5)
print("x1          : {0}".format(x1))

# Criando um array com valores randomicos
x2 = np.random.rand(10)
print("x2          : {0}".format(x2))

# Note que o ultimo valor é o final - intervalo

# Realizando operações com o vetor
# Somando 2 a todos os campos do vetor
soma = x1 + 2
print("soma        : {0}".format(soma))

# Multiplicando cada campo por 0.5
mult = x1 * 0.5
print("multiplicação : {0}".format(mult))

# Somando um array com outro array
x2   = np.arange(0,10)
soma = x1 + x2
print("x1 + x2      : {0}".format(soma))

# Divindo um array por outro
mult = x2 * x1
print("x2 * x1      : {0}".format(mult))

# Acessando valores individuais de vetores
print("x1[0] : {0}".format(x1[0]))

# Nota-se que pode-se alocar valores para os valores acessados
x1[0] = 10
print("x1[0] : {0}".format(x1[0]))
```

5.2. Interações com vetores

```
import numpy as np

# Alocando um vetor de 0 até 50 com passo de 1 em 1
x = np.arange(0,50)
print("x  : {0}".format(x))

# Partindo o Array
# A notação do python para o particionamento de arrays é [inicio:final:passo]
# Nota-se que não necessáriamente é necessário utilizar todos os ':'

# partindo o array em 2 metades
x1 = x[:25]
x2 = x[25:50]
print("x1  : {0}".format(x1))
print("x2  : {0}".format(x2))

# Realizando o 'Downsampling' do array
x3 = x[::2]
print("x3  : {0}".format(x3))

# Juntando 2 arrays
x4 = np.concatenate((x1,x2))
print("x4  : {0}".format(x4))

# Comparando 2 arrays
comp = (x4 == x)
print("comp: {0}".format(comp))

# Transformando um array do tipo numpy para lista do python
x1_list = x1.tolist()
print("classe antes: ",type(x1))
print("classe depois: ",type(x1_list))

# Adicionando termos em uma lista python
x1_list.append(255)
print("x1_list: {0}".format(x1_list))

# Acessando informações do array
len_x1 = len(x1)
dim_x1 = x1.shape
print("tamanho x1: {0}, dimensões x1: {1}".format(len_x1,dim_x1))
```

5.3. Alocando matrizes e primeiras operações

```
import numpy as np
from pprint import pprint
# Nota :  a biblioteca pprint serve para mostrar na tela de
# forma mais agradável matrizes e outras estruturas

# Declarando uma matriz e visualizando-a
A = np.array([[2, 4], [5, -6]])
print("A:")
pprint(A)

# Visualizando somente a primeira linha
print("A[0]:")
pprint(A[0])

# Visualizando somente a primeira coluna
print("A j1:")
pprint(A[:,0])

# Operações basicas com matrizes
B = np.array([[3,4],[6,9]])
print("B :")
pprint(B)

# Soma
print("A + B :")
pprint(A + B)

# Multiplicação
print("A * B :")
pprint( A * B)

# Transposição
print("A_t:")
pprint(np.transpose(A))

# Transformando matriz em vetor unidimensional
tran_a = A.flatten()
print("A unidimensional :")
pprint(tran_a)

# Manipulando as dimensões de um array
C = np.arange(0,4)
C = C.reshape(2,2)
print("C :")
pprint(C)
```

5.4. Exercício envolvendo as operações vistas

Exercício: Escreva um código que execute as instruções abaixo, utilizando somente o que foi exposto nos códigos anteriores:

- Aloque um vetor x com 10 elementos, começando do 0
- Aloque um vetor y com 10 valores randômicos
- Aloque uma matriz com 2 colunas e 10 linhas
- Passe os valores de x e y para a matriz, de forma que os valores de x fiquem na primeira coluna e de y fiquem na segunda coluna.
- Mostre na tela a matriz com os valores de x e y
- Adicione um valor escalar para cada valor da coluna y da matriz
- Mostre a matriz novamente

Resolução:

```
import numpy as np
from pprint import pprint
# declarando um vetor de x
x = np.arange(0,10)

# declarando um vetor y de valores randomicos
y = np.random.rand(10)

# Alocando um array de 0
z = np.zeros(20)
z = z.reshape(10,2)
# Colocar a primeira coluna como x
z[:,0] = x
# Colocar a segunda coluna como y
z[:,1] = y
# Mostrando na tela a matriz
print("Matrix xy : ")
pprint(z)
# Adicionando valores para a coluna y
z[:,1] = z[:,1] + 4
# Mostrando na tela a matriz
print("Matrix xy : ")
pprint(z)
```

6. Conclusão

O laboratório já empenha papel importante na formação de profissionais da área de alta e extra altas tensões, realizando várias pesquisas na área. Os ensaios realizados são de extrema importância para essa função. Além disso, após o laboratório conquistar a certificação necessária será possível empenhar pepel ainda maior na área, atingindo os polos industriais locais, contribuindo ainda mais para o desenvolvimento do Pará e da região Norte na área de altas e extra-altas tensões.

REFERENCIAS BIBLIOGRÁFICAS

NAIDU, M. S.; KAMARAJU, V. – High Voltage Engineering – McGrawhill, 1996

WADHWA, C.L. – High Voltage Engineering – New Age International, 2007

AE **Fitzgerald**, JR Kingsley, C Umans, SD **Máquinas** - 2006 - Bookman, Porto Alegre

KUFFEL, E.; ZARNGL, W.S.; KUFFEL, J. – High Voltage Engineering Fundamentals – Newnes, 2000