



PROCESSAMENTO DE NUVEM DE PONTOS 3D

CGEO- 7028

Breve revisão de programação

- PYTHON
COLAB

UFPR – Departamento de Geomática
Prof. Jorge Centeno
2023
copyright@ centenet

PYTHON

-
- Vamos iniciar programando em Python, seja no PYZO ou no COLAB

Neste exercício usaremos a linguagem Python para criar e manipular imagens.



usando PYzo

Por favor, verifique se o Winpy está instalado no seu computador. Se não, obtenha uma copia do site oficial [PYZO](#)

abra o PYZO. Você terá uma tela com um prompt ">>" esperando comandos para interação. Você pode usar os ícones na barra superior para alterar o diretório de trabalho ou usar os comando no prompt.

```
a=5 b=4 c= a + b print(c)
```

Usando Colab

Usando Colab: Crie uma conta Google e entre no [COLAB](#)

Você tem duas opções: "código", ou "texto", use texto para comentários e código para partes do programa

No editor de textos, crie um programa com as seguintes linhas:
Salvar e rodar o programa

```
a=5  
b=4  
c= a + b  
print(c)
```

Para isto, mude o nome do arquivo. Verifique se ele foi criado (no Drive, se você está usando o COLAB). Veja a resposta na área interativa logo abaixo!

Agora, altere o valor de "a" ou de "b" e rode o programa de novo.

usando PYzo

Salve seu programa em seu diretório de trabalho, por exemplo com o nome "prog001.py". Para isto, mude o nome do arquivo. Verifique se ele foi criado na pasta selecionada

Para rodar o programa: pode teclar "run" ou control+Shift+E.

Usando Colab

Mude o nome "rename" de seu programa. Deve estar como Untitled. Renomeie para "prog001.py". Ele será armazenado em seu Google Drive

Acionar a tecla play (triângulo)

Um programa

a) Incluir bibliotecas

PYTHON permite processar operações simples. Para efetuar processamentos mais avançados é necessário adicionar bibliotecas com funções. Isto é feito usando o comando `import` e o nome da biblioteca

Ex:

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

Numpy: usado para executar uma ampla variedade de operações matemáticas em arrays.

matplotlib.pyplot:

Conjunto de funções que permitem criar gráficos, ler e escrever imagens. por exemplo, criar uma figura, criar uma área de plotagem em uma figura, plotar algumas linhas em uma área de plotagem, inserir rótulos, etc.

EXEMPLO

Criar dois vetores (com numpy que serve para manipular arrays) e plotar um gráfico, usando matplotlib (biblioteca de plotagem matemática).

Passos:

Carregar bibliotecas

NUMPY :

MATPLOTLIB.PY PLOT:

criar dois vetores x e y.

plotar o gráfico $y=f(x)$. Ele é criado, mas não mostrado.

mostrar o gráfico com `show()`

se quiser, pode adicionar uma legenda (2da opção).

```
import numpy as np
import matplotlib.pyplot as plt
# definir vetores x e y
x=[0, 50, 60, 80, 90, 100]
y=[0, 0, 1, 1, 0, 0]
# plotar o gráfico x vs y
plt.plot(x,y)
plt.show()

# ou com legenda...
plt.plot(x,y, label='X vs Y')
plt.legend()
plt.show()
```

Leitura

- Mas... Se quisermos ler um arquivo de pontos XYZ do disco rígido?

É muito prático ler dados de um arquivo ASCII. Vamos então escrever uma série de linhas que permitem ler dados de um arquivo no disco (pode ser seu diretório, ou no google drive).

```
import numpy as np
# ler uma tabela com 2 linhas contendo 9 numeros cada uma
Nlinhas=2 # aqui definimos nro de linhas e colunas da tabela
Ncolunas=9
T = np.zeros((Nlinhas, Ncolunas), dtype = float)
# criamos tabela vazia
f = open('tabela1.txt', 'r') # abrir arquivo
cl=0 # um contador de linhas
for line in f: # varrer linhas no arquivo
    cc=0; # um contador de colunas
    for n in line.split(','): #varrer elementos da linha lida
        a=float(n) # converter em float e armazenar em T(cl, cc)
        T[cl,cc]=a
        cc=cc+1
    cl=cl+1 f.close()
```

Opção 1, usando Pyzo.


```
import numpy as np
import matplotlib.pyplot as plt
from google.colab import drive # isto serve para usar o drive
drive.mount('/content/gdrive') # montar seu drive como um disco virtual
# ler uma tabela com 2 linhas contendo 9 numeros cada uma
Nlinhas=2 # aqui defino o nro de linhas e colunas da tabela
Ncolunas=9
T = np.zeros((Nlinhas, Ncolunas),dtype = float) # criamos tabela vazia
f = open('gdrive/My Drive/tabela1.csv', 'r')
cl=0 # um contador de linhas
for line in f: # varrer linhas no arquivo
    cc=0 # um contador de colunas
    for n in line.split(','): #varrer elementos da linha lida
        a=float(n) # converter em float e armazenar em T(cl, cc)
        T[cl,cc]=a
        cc=cc+1
    cl=cl+1
f.close()
plt.show()
```

Opção 1, usando Colab



```
print(T)
```

```
# separar X e Y, cada um em uma linha ...
```

```
x=T[0,:]
```

```
y=T[1,:]
```

```
plt.plot(x,y, color='red')
```



Laspy

Laspy

Laspy is a python library for reading, modifying and creating LAS LiDAR files.

Laspy is compatible with Python 3.7+.

Features

- LAS support.
- LAZ support via `lazrs` or `laszip` backend.
- LAS/LAZ streamed/chunked reading/writing.
- [COPC](#) support over files.
- [COPC](#) support over https with `requests` package.
- CRS support via `pyproj` package.

<https://pypi.org/project/laspy/>

Instalação

Laspy can be installed either with pip:

```
pip install laspy # without LAZ support
```

Or

```
pip install laspy[laszzip] # with LAZ support via LASzip
```

Or

```
pip install laspy[lazrs] # with LAZ support via lazrs
```



```
import laspy
```

```
# podemos ler informacoes basicas
```

```
with laspy.open('lake.laz') as f:
```

```
    print(f"Point format: {f.header.point_format}")
```

```
    print(f"Number of points: {f.header.point_count}")
```

```
    print(f"Number of vlrs: {len(f.header.vlrs)}")
```

```
# podemos ler os dados e visualizar
```

```
import numpy as np
```

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

```
las = laspy.read('lake.laz')
```

```
fig = plt.figure()
```

```
ax = plt.axes(projection='3d')
```

```
ax.scatter3D(las.x, las.y, las.z, cmap='Greens');
```

```
plt.show()
```

Agora, escreva um programa que crie uma grade regular com esses dados:

- leia os valores mínimos e máximos em X (x_{\min} , x_{\max}) e Y (y_{\min} , y_{\max}).
- calcule a faixa de variação em X e Y , isto define a área coberta pelo levantamento
- escolha uma resolução para a grade. Que tal dois metros?
- calcule o número de linhas e colunas
- para cada ponto:
 - calcule a distância do ponto à origem da grade
 - calcule a linha e coluna correspondentes a esta posição
 - se a célula estiver vazia ($==0$) copie o valor na célula
- mostre a grade

