Instituto Nacional de Pesquisas Espaciais

Curso de Pós-Graduação em Sensoriamento Remoto

SER-347 – Introdução à Programação para Sensoriamento Remoto
Lista de Exercícios 01

Dr. Gilberto Ribeiro de Queiroz (gilberto.queiroz@inpe.br)
Dr. Thales Sehn Körting (thales.korting@inpe.br)
Dr. Fabiano Morelli (fabiano.morelli@inpe.br)

22 de março de 2018

Introdução à Programação com a Linguagem Python - Lista 01

Exercícios

Atenção:

1. Os exercícios práticos devem ser desenvolvidos em Python. Escreva a documentação que achar pertinente dentro do próprio código fonte, que deverá utilizar a codificação de caracteres UTF-8.

2. A solução de cada exercício deverá ser entregue em um único arquivo de código fonte na linguagem Python. Use arquivos com a extensão `.py` com a seguinte nomenclatura: exercicio-{numero}.py. Ex: `exercicio-01.py`.

3. Envie por e-mail **um único** arquivo no **formato zip**, chamado `lista01.zip`, contendo todos os arquivos de código fonte dos exercícios.

4. O título do e-mail deve seguir o seguinte padrão[1]:
   `[ser347-2018][lista-01] nome-completo-aluno`.

5. **Prazo para entrega:** 27/03/2018 - 22:00

[1] Não use acentos ou caracteres especiais no nome do arquivo.

Exercício 01. Usando o terminal interativo do Python, use o comando `type(objeto)` e descubra o tipo de dados dos seguintes objetos:

- `1`
- `True`
- `5 / 2`
- `2.5`
- `False`
- `5 + 3j`

Obs.: Você deverá entregar um arquivo chamado `exercicio-01.py` com a sequência de comandos digitada no terminal interativo. Para cada comando, inclua na linha seguinte o seu resultado como um texto comentado.

Exercício 02. Qual o resultado das seguintes expressões:

- `5 / 2`
- `7 * 4 + 2`
- `(7 * 4) + 2`
- `7 * (4 + 2)`
- `2 ** 3`
- `2 ** 3 ** 4`
- `2 ** -3 ** 4`
- `5 % 2`
- `6 % 2`
- `7 % 2`
- `8 % 2`
- `5 + 1`
- `5.0 + 1`
- `5 * math.log10(100) - 8 ** 2`
- `math.pi`

- `math.sin(math.pi / 2 )`
- `math.cos(math.pi / 4 )`

Obs.: Você deverá entregar um arquivo chamado `exercicio-02.py` com a sequência de comandos digitada no terminal interativo. Para cada comando, inclua na linha seguinte o seu resultado como um texto comentado.

Exercício 03. Faça um programa que pergunte ao usuário valores hipotéticos de reflectância do vermelho (*red*) e infravermelho próximo (*nir*), e compute o *NDVI*, escrevendo na tela o valor computado.

Exercício 04. *The McFeeters' `NDWI` is computed as follows:*

$$NDWI = \frac{(Xgreen - Xnir)}{(Xgreen + Xnir)}$$

where `Xgreen` refers to the green band (MODIS band 4) and `Xnir` refers to the nir band (MODIS band 2).
The `NDVI` is calculated according to the following formula:

$$NDVI = \frac{(Xnir - Xred)}{(Xnir + Xred)}$$

where `Xnir` refers to the nearinfrared band (MODIS band 2) and `Xred` refers to the red band (MODIS band 1)

Fonte: `https://deltas.usgs.gov/fm/default.aspx`.

De acordo com as equações acima, crie um programa que leia os valores de `Xgreen`, `Xnir` e `Xred` e imprima como resultados os valores de `NDWI` e `NDVI`.

Exercício 05. Escreva um programa que pergunte ao usuário as coordenadas de latitude e longitude, em grau-decimal, de dois pontos quaisquer na esfera terrestre e que apresente a distância entre eles. Essa distância deverá ser calculada de acordo com a fórmula de Haversine[2]:

$$d(p, q) = 2r \arcsin \sqrt{sin^2(\frac{\phi_2 - \phi_1}{2}) + \cos\phi_1 \cos\phi_2 \sin^2(\frac{\lambda_2 - \lambda_1}{2})}$$

onde:

- r: é o raio da esfera ($\sim$ 6371km)
- ϕ_1 e ϕ_2: latitude dos pontos em radianos.
- λ_1 e λ_2: longitude dos pontos em radianos.

[2]Veja: `https://en.wikipedia.org/wiki/Haversine_formula`.

Obs.: Lembrem-se que todas as funções trigonométricas e suas inversas, bem como funções para conversão entre graus e radianos encontram-se na biblioteca matemática do Python.

Exercício 06. Escreva uma função que calcule a menor distância entre um ponto e uma reta. Faça um programa que utilize esta função, possibilitando que o usuário entre com as informações de dois pontos pertencentes à reta, bem como o ponto para o qual deva ser avaliada a distância.

Dica: A menor distância entre um ponto P e uma reta r corresponde ao segmento de reta perpendicular à r que parte de P e chega a r (Figura 1).

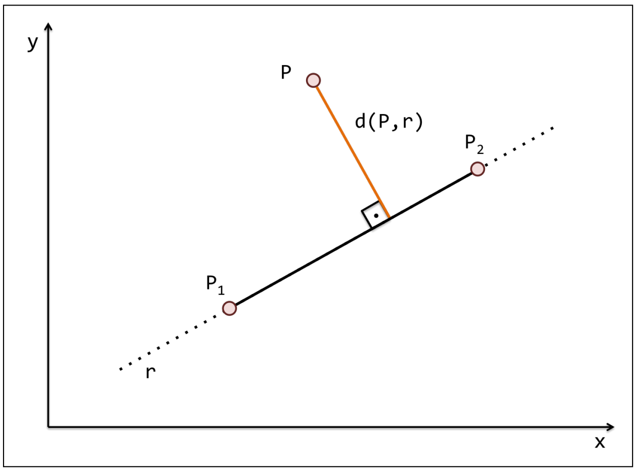

Figura 1: Distância mínima entre um ponto e uma reta.

Uma forma de computar esta distância consiste na utilização da forma normal de Hessean para retas[3]. Para uma reta r que passa pelos ponto $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$, temos a seguinte fórmula:

$$h(x,y) = \frac{(y_2 - y_1) \times (x - x_1) - (x_2 - x_1) \times (y - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} = 0$$

com: $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} > 0$.
A distância de $P = (x, y)$ a r é dada por $|h(x, y)|$.

[3]Jurg Nievergelt e Klaus H. Hinrichs. Algorithms & Data Structures with Applications to Graphics and Geometry. Prentice Hall, 1993. 350 pp.