

☰ 10

Manipulação de diretórios e arquivos com Python

No último vídeo, aprendemos algumas técnicas de manipulação de diretórios e arquivos utilizando Python. O método `walk(top)` da biblioteca nos retorna os nomes dos arquivos e diretórios que pertençam ao endereço passado como argumento (`top`), possibilitando iterar entre eles em mais de uma direção.

Para responder a questão abaixo, suponha que você criou um novo notebook na pasta que estamos trabalhando atualmente (**05.Informações Censitárias**), digitou os códigos abaixo em três células diferentes e obteve os seguintes resultados:

In [1]:

```
import os
for root, dirs, files in os.walk('../01.Dados/Transporte'):
    print(root)
```

```
../01.Dados/Transporte
../01.Dados/Transporte\BRT
../01.Dados/Transporte\Metrô
../01.Dados/Transporte\Trem
```

In [2]:

```
for root, dirs, files in os.walk('../01.Dados/Transporte'):
    print(dirs)
```

```
['BRT', 'Metrô', 'Trem']
[]
[]
[]
```

In [3]:

```
for root, dirs, files in os.walk('../01.Dados/Transporte'):
    print(files)
```

```
[]
['Estações_BRT.geojson']
['Estações_Metrô.geojson']
['Estações_Trem.geojson']
```

Observação: Neste exercício, estamos assumindo que você manteve a organização dos arquivos e diretório proposto no curso.

Assumindo que os códigos estejam corretos, avalie as resposta de cada item:

Seleciona uma alternativa

A Somente **In [1]** está correta.

B **In [2]** e **In [3]** estão incompletas.

C Somente **In [2]** está correta.

D Todas estão corretas.