Pró-reitora de Pós-Graduação, Pesquisa e Inovação

Especialização em Ciências de Dados e Analytics

Programação para Ciência de Dados

Parte 1 / Aula 3: Revisão de Programação Estruturada com Python

Agenda

- Overview de Programação com a Linguagem Python

✓ Tipos numéricos
✓ Strings
✓ Prints
✓ if, else, elif
✓ for, while
✓ range()
✓ Operadores Lógicos

✓ Listas
- Dicionários
- Tuplas e Sets
- Funções
- Expressões lambda
- Mapas e Filtros

Dicionários

- Assim como listas, os dicionários permitem elementos de tipos distintos

Set é diferente de List

```
In [67]: dic = {'UPE':1, 'UFPE':2, 'UNICAP':3}

In [68]: dic['UFPE']

Out[68]: 2

In [69]: dicProfs = {'UPE':{'Bruno','Byron'},'UFPE':{'Aluizio','Tereza'},
                     'UNICAP':{'Anthony','Madeiro'}}

In [70]: dicProfs['UPE']

Out[70]: {'Bruno', 'Byron'}

In [71]: type(dicProfs)

Out[71]: dict

In [72]: dicProfs['UPE'][0]

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-72-07499f6fba37> in <module>()
----> 1 dicProfs['UPE'][0]

TypeError: 'set' object does not support indexing

In [73]: type(dicProfs['UPE'])

Out[73]: set

In [74]: dicProfs = {'UPE':['Bruno','Byron'],'UFPE':['Aluizio','Tereza'],
                     'UNICAP':['Anthony','Madeiro']}

In [75]: dicProfs['UPE'][0]

Out[75]: 'Bruno'
```

Tuplas

```
In [84]: dicProfs = {'UPE':['Bruno','Byron'],'UFPE':['Aluizio','Tereza'],
                     'UNICAP':['Anthony','Madeiro']}

In [85]: dicProfs['UPE'][0]
Out[85]: 'Bruno'

In [86]: dicProfs['UPE'][0] = 'Carmelo'

In [87]: dicProfs['UPE']
Out[87]: ['Carmelo', 'Byron']

In [88]: dicProfs = {'UPE':('Bruno','Byron'),'UFPE':('Aluizio','Tereza'),
                     'UNICAP':('Anthony','Madeiro')}

In [89]: dicProfs['UPE'][0]
Out[89]: 'Bruno'

In [90]: dicProfs['UPE'][0] = 'Carmelo'
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-90-1b2efdb2fef7> in <module>()
----> 1 dicProfs['UPE'][0] = 'Carmelo'

TypeError: 'tuple' object does not support item assignment

In [91]: type(dicProfs['UPE'])
Out[91]: tuple
```

Tuplas são constantes!

Exercício

- A partir dos dicionários '*en*' e '*pt_br*' definidos abaixo, crie um terceiro dicionário denominado '*translator*' cujas chaves são, respectivamente, 'en' e 'pt_br' e os valores sejam os dicionários correspondentes. Exiba o dicionário criado.

```
pt_br = {1:'um', 2:'dois', 3:'três'}
en = {1:'one', 2:'two', 3:'three'}
```

Funções

```
def par(valor):
    if valor % 2 == 0:
        return True
    else:
        return False
```

```
par(3)
```

```
False
```

```
par(4)
```

```
True
```

Exercício:

- Construa uma função que calcula o fatorial de um número

```
def fat(n):
    if n == 0:
        return 1
    else:
        return n * fat(n-1)
```

```
fat(0)
```

```
1
```

```
fat(4)
```

```
24
```

Funções Lambda

- Úteis para um contexto específico do código

```
def par(valor):
    if valor % 2 == 0:
        return True
    else:
        return False
```

```
lambda valor: (valor % 2 == 0)
```

```
<function __main__.<lambda>>
```

Map

- Útil para aplicar uma função a uma lista

```
seq = range(1,10)
```

```
seq
```

```
range(1, 10)
```

```
pares = list(map(par,seq))
```

```
pares
```

```
[False, True, False, True, False, True, False, True, False]
```

Map com Função Lambda

- Útil para aplicar uma função a uma lista

```
pares = list(map(par,seq))
```

```
pares
```

```
[False, True, False, True, False, True, False, True, False]
```

```
pares = list(map(lambda valor : valor % 2 == 0,seq))
```

```
pares
```

```
[False, True, False, True, False, True, False, True, False]
```

Filter

- Útil para filtrar elementos de uma lista que atendem alguma condição

```
pares = list(filter(lambda valor : valor % 2 == 0,seq))
```

```
pares
```

```
[2, 4, 6, 8]
```

Exercício

- A partir da lista definida abaixo, filtre-a de forma que somente as palavras com menos de 3 caracteres sejam mantidas. Dica: utilize a função filter.

```
lista = ['tempor', 'erat,', 'in,', 'elit', 'Etiam', 'tincidunt.', 'rutrum', 'ut,', 'lacinia', 'Integer',
'Nam', 'turpis', 'Nulla', 'non.', 'vehicula', 'diam', 'porttitor', 'blandit', 'Sed', 'pharetra', 'erat',
'hendrerit', 'tristique', 'vulputate', 'faucibus.', 'augue.', 'potenti.', 'vel', 'eros', 'imperdiet,', 'a.',
'dolor.', 'pretium', 'Fusce', 'sit', 'ornare', 'Morbi', 'quis', 'fringilla', 'lobortis', 'tempus', 'mauris',
'ante,', 'lacus', 'porta.', 'faucibus,', 'quis.', 'vestibulum', 'primis', 'luctus,', 'ullamcorper.',
'augue', 'nec', 'mollis', 'lectus', 'dolor', 'sodales', 'ligula,', 'dignissim', 'sem', 'varius', 'mi', 'eu',
'elit.', 'semper', 'id,', 'tempus.', 'finibus.', 'neque', 'quam.', 'scelerisque', 'lorem', 'diam,', 'Cras',
'nisi', 'Lorem', 'leo,', 'Ut', 'ut', 'feugiat.', 'ante', 'venenatis', 'fermentum', 'congue', 'urna',
'Praesent', 'Donec', 'Vestibulum', 'purus.', 'Nullam', 'tincidunt', 'efficitur', 'velit', 'commodo.',
'iaculis', 'sed,', 'volutpat', 'amet', 'mauris.', 'odio', 'a', 'Interdum', 'neque.', 'risus', 'vitae',
'consectetur', 'adipiscing', 'at', 'Aliquam', 'molestie', 'euismod', 'odio.', 'sed', 'in', 'suscipit',
'augue,', 'sapien', 'posuere', 'euismod,', 'ipsum', 'et.', 'maximus.', 'risus,', 'Suspendisse', 'et',
'facilisis', 'elementum', 'efficitur,', 'ac', 'nulla.', 'quam', 'arcu', 'fames', 'Nunc', 'pharetra,',
'laoreet', 'ligula', 'fermentum,', 'auctor', 'tortor,', 'Curabitur', 'eget', 'finibus', 'ultrices',
'malesuada', 'purus', 'congue,', 'amet,', 'fermentum.', 'dui']
```