

Nome: João Emanuel - Matricula: 162080263 - Data
13/agosto/2020

Listas

In [1]:

```
l = [ 'alcool em gel', 'algodão', 'máscara', 'antisséptico', 'alcool puro' ]  
l
```

Out[1]:

```
['alcool em gel', 'algodão', 'máscara', 'antisséptico', 'alcool pur  
o']
```

In [2]:

```
l2 = [ 'lenço de papel', 'pilha', 'pasta de dente' ]  
l2
```

Out[2]:

```
['lenço de papel', 'pilha', 'pasta de dente']
```

In [3]:

```
item1 = l[0]  
item2 = l[1]  
print(item1, ' - ', item2)
```

```
alcool em gel - algodão
```

In [4]:

```
# tamanho da lista  
len(l), len(l2)
```

Out[4]:

```
(5, 3)
```

In [5]:

```
l2[1] = 'pilha palito'  
l2
```

Out[5]:

```
['lenço de papel', 'pilha palito', 'pasta de dente']
```

In [6]:

```
del l2[1]  
l2
```

Out[6]:

```
['lenço de papel', 'pasta de dente']
```

In [7]:

```
lista = [ [1, 2, 3], [4, 5, 6], [7, 8, 9] ]  
print(lista)
```

```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

In [8]:

```
lista[1]
```

Out[8]:

```
[4, 5, 6]
```

In [9]:

```
lista[0]
```

Out[9]:

```
[1, 2, 3]
```

In [10]:

```
lista[1][0]
```

Out[10]:

```
4
```

In [11]:

```
a = lista[1][0]  
a
```

Out[11]:

```
4
```

In [12]:

```
b = lista[1]  
b
```

Out[12]:

```
[4, 5, 6]
```

In [13]:

```
type(b)
```

Out[13]:

list

In [14]:

```
c = lista[1][0] + 10  
c
```

Out[14]:

14

Concatenar listas

In [15]:

```
l, l2
```

Out[15]:

```
(['alcool em gel', 'algodão', 'máscara', 'antisséptico', 'alcool pur  
o'],  
 ['lenço de papel', 'pasta de dente'])
```

In [16]:

```
l3 = l + l2  
l3
```

Out[16]:

```
['alcool em gel',  
 'algodão',  
 'máscara',  
 'antisséptico',  
 'alcool puro',  
 'lenço de papel',  
 'pasta de dente']
```

Operandor in

In [17]:

```
ln = [ 1000, -2, -5, 3.14159]  
ln
```

Out[17]:

```
[1000, -2, -5, 3.14159]
```

In [18]:

```
10 in ln
```

Out[18]:

False

In [19]:

```
-2 in ln
```

Out[19]:

True

Funções

In [20]:

```
len(ln)
```

Out[20]:

4

In [21]:

```
max(ln)
```

Out[21]:

1000

In [22]:

```
min(ln)
```

Out[22]:

-5

In [23]:

```
ln.append(9)  
ln
```

Out[23]:

[1000, -2, -5, 3.14159, 9]

In [24]:

```
ln = ln + [ 33 ]  
ln
```

Out[24]:

[1000, -2, -5, 3.14159, 9, 33]

In [25]:

```
ln = ln + 33
```

```
-----  
-----  
TypeError                                Traceback (most recent call  
last)  
<ipython-input-25-32045588179c> in <module>  
----> 1 ln = ln + 33
```

TypeError: can only concatenate list (not "int") to list

In [26]:

```
ln = ln + [ 33 ]  
ln
```

Out[26]:

```
[1000, -2, -5, 3.14159, 9, 33, 33]
```

In [27]:

```
ln.count(33)
```

Out[27]:

```
2
```

In [28]:

```
a = []  
type(a)
```

Out[28]:

```
list
```

In [29]:

```
a.append(10)  
a
```

Out[29]:

```
[10]
```

In [30]:

```
a.append(20)
```

In [31]:

```
a.append(30)
```

In [32]:

```
a
```

Out[32]:

```
[10, 20, 30]
```

In [33]:

```
a
```

Out[33]:

```
[10, 20, 30]
```

In [34]:

```
b = a  
b
```

Out[34]:

```
[10, 20, 30]
```

In [35]:

```
b = []  
for item in a:  
    b.append(item)  
b
```

Out[35]:

```
[10, 20, 30]
```

In [36]:

```
list(range(10))
```

Out[36]:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [37]:

```
ln = []  
for i in range(10):  
    print(i+1)  
    ln.append(i)
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

In [38]:

```
ln
```

Out[38]:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [39]:

```
ln.sort()
```

In [40]:

```
ln.append(-1)  
ln
```

Out[40]:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -1]
```

In [41]:

```
ln.sort()  
print(ln)
```

```
[-1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Tuplas

In [42]:

```
t1 = ( "UEPB", 2020, 'cct' )  
t1
```

Out[42]:

```
('UEPB', 2020, 'cct')
```

In [43]:

```
t1.append('DC')
```

```
-----  
-----  
AttributeError                                Traceback (most recent call  
l last)  
<ipython-input-43-01542a73f7bd> in <module>  
----> 1 t1.append('DC')
```

AttributeError: 'tuple' object has no attribute 'append'

In [44]:

```
t1[0]
```

Out[44]:

```
'UEPB'
```

In [45]:

```
len(t1)
```

Out[45]:

```
3
```

In [46]:

```
t1[1:]
```

Out[46]:

```
(2020, 'cct')
```

In [47]:

```
t1[0] = 'uepb-cct'
```

```
-----  
-----  
TypeError                                Traceback (most recent call  
l last)  
<ipython-input-47-c83e5810b298> in <module>  
----> 1 t1[0] = 'uepb-cct'
```

TypeError: 'tuple' object does not support item assignment

In [48]:

```
#del t1
```

In [49]:

```
l1 = list(t1)  
type(l1)
```

Out[49]:

```
list
```

In [50]:

```
l1
```

Out[50]:

```
['UEPB', 2020, 'cct']
```


In [51]:

```
l1[0] = 'uepb-cct'  
print(l1)
```

```
['uepb-cct', 2020, 'cct']
```

Dicionário

In [52]:

```
d1 = { 'Mateus':24, 'Fernanda':22, 'Tamires':26, 'Cristiano':25 }  
type(d1)
```

Out[52]:

```
dict
```

In [53]:

```
d1
```

Out[53]:

```
{'Mateus': 24, 'Fernanda': 22, 'Tamires': 26, 'Cristiano': 25}
```

In [54]:

```
d1.values()
```

Out[54]:

```
dict_values([24, 22, 26, 25])
```

In [55]:

```
d1.keys()
```

Out[55]:

```
dict_keys(['Mateus', 'Fernanda', 'Tamires', 'Cristiano'])
```

In [56]:

```
d1.items()
```

Out[56]:

```
dict_items([('Mateus', 24), ('Fernanda', 22), ('Tamires', 26), ('Cristiano', 25)])
```

In [57]:

```
d1['Fernanda']
```

Out[57]:

```
22
```

In [58]:

```
d1['Pedro'] = 27  
d1
```

Out[58]:

```
{'Mateus': 24, 'Fernanda': 22, 'Tamires': 26, 'Cristiano': 25, 'Pedro': 27}
```

In [59]:

```
d2 = {}
```

In [60]:

```
d2
```

Out[60]:

```
{}
```

In [61]:

```
d2['Mariana'] = 28  
d2['Fabio'] = 27  
d2
```

Out[61]:

```
{'Mariana': 28, 'Fabio': 27}
```

In [62]:

```
# junção  
d1.update(d2)  
d1
```

Out[62]:

```
{'Mateus': 24,  
 'Fernanda': 22,  
 'Tamires': 26,  
 'Cristiano': 25,  
 'Pedro': 27,  
 'Mariana': 28,  
 'Fabio': 27}
```

In [63]:

```
d1[10] = 340  
d1
```

Out[63]:

```
{'Mateus': 24,  
 'Fernanda': 22,  
 'Tamires': 26,  
 'Cristiano': 25,  
 'Pedro': 27,  
 'Mariana': 28,  
 'Fabio': 27,  
 10: 340}
```

In [64]:

```
d1['Mateus']
```

Out[64]:

24

In [65]:

```
d1['mateus']
```

```
-----  
-----  
KeyError                                Traceback (most recent call  
l last)  
<ipython-input-65-72467dc174bd> in <module>  
----> 1 d1['mateus']
```

KeyError: 'mateus'

In [66]:

```
d1[10]
```

Out[66]:

340

In [67]:

```
d1[0]
```

```
-----  
-----  
KeyError                                Traceback (most recent call  
l last)  
<ipython-input-67-f64ff61913e8> in <module>  
----> 1 d1[0]
```

KeyError: 0

In [68]:

```
d1.keys()
```

Out[68]:

```
dict_keys(['Mateus', 'Fernanda', 'Tamires', 'Cristiano', 'Pedro', 'Mariana', 'Fabio', 10])
```

In [69]:

```
idade1, idade2 = d1['Mateus'], d1['Fabio']  
idade1, idade2
```

Out[69]:

```
(24, 27)
```

In [70]:

```
d3 = { 'k1': 1256,  
        'k2': [23, 27, 'UEPB', 29],  
        'k3': ['alcool em gel', 'algodão', 'máscara', 'antisséptico', 'alcool puro'] }  
d3
```

Out[70]:

```
{'k1': 1256,  
 'k2': [23, 27, 'UEPB', 29],  
 'k3': ['alcool em gel', 'algodão', 'máscara', 'antisséptico', 'alcohol puro']}
```

In [71]:

```
d3['k2']
```

Out[71]:

```
[23, 27, 'UEPB', 29]
```

In [72]:

```
d3['k2'][2]
```

Out[72]:

```
'UEPB'
```

In [73]:

```
produto = d3['k3'][4]  
produto
```

Out[73]:

```
'alcool puro'
```

In [74]:

```
produto = d3['k3'][5]
produto
```

```
-----
-----
IndexError                                Traceback (most recent call
last)
<ipython-input-74-bb0cc8ba3b1a> in <module>
----> 1 produto = d3['k3'][5]
      2 produto
```

IndexError: list index out of range

In [75]:

```
produto = d3['k3'][4].upper()
produto
```

Out[75]:

'ALCOOL PURO'

In [76]:

```
v1 = d3['k1'] - 1000
v1
```

Out[76]:

256

In [77]:

```
d3
```

Out[77]:

```
{'k1': 1256,
 'k2': [23, 27, 'UEPB', 29],
 'k3': ['alcool em gel', 'algodão', 'máscara', 'antisséptico', 'alco
ol puro']}
```

In [78]:

```
d3['k1'] = d3['k1'] - 1000
d3
```

Out[78]:

```
{'k1': 256,
 'k2': [23, 27, 'UEPB', 29],
 'k3': ['alcool em gel', 'algodão', 'máscara', 'antisséptico', 'alco
ol puro']}
```

In [79]:

```
d3['k1'] -= 100 # d3['k1'] - 1000  
d3
```

Out[79]:

```
{'k1': 156,  
 'k2': [23, 27, 'UEPB', 29],  
 'k3': ['alcool em gel', 'algodão', 'máscara', 'antisséptico', 'alco  
ol puro']}
```

In [80]:

```
d4 = { 'k1': {'k11': 1},  
       'k2': -2 }  
d4
```

Out[80]:

```
{'k1': {'k11': 1}, 'k2': -2}
```

In [81]:

```
d4['k2']
```

Out[81]:

```
-2
```

In [82]:

```
d4['k1']
```

Out[82]:

```
{'k11': 1}
```

In [83]:

```
d4['k1']['k11']
```

Out[83]:

```
1
```

In [84]:

```
d3.keys()
```

Out[84]:

```
dict_keys(['k1', 'k2', 'k3'])
```

In [85]:

```
'k1' in d3.keys()
```

Out[85]:

```
True
```

In [86]:

```
if 'k1' in d3.keys():  
    print('ok')  
    x = d3['k1']  
    print(x)
```

ok
156

Aula-03-Diccionarios-Dataframes-Python-13-Agosto-2020

Tweeter Json

In [87]:

```
d5 = {  
    "attributes": {},  
    "bounding_box":  
        {  
            "coordinates":  
                [[  
                    [-77.119759, 38.791645],  
                    [-76.909393, 38.791645],  
                    [-76.909393, 38.995548],  
                    [-77.119759, 38.995548]  
                ]],  
            "type": "Polygon"  
        },  
    "country": "United States",  
    "country_code": "US",  
    "full_name": "Washington, DC",  
    "id": "01fbe706f872cb32",  
    "name": "Washington",  
    "place_type": "city",  
    "url": "http://api.twitter.com/1/geo/id/0172cb32.json"  
}
```

In [88]:

```
d5
```

Out[88]:

```
{'attributes': {},  
 'bounding_box': {'coordinates': [[[-77.119759, 38.791645],  
   [-76.909393, 38.791645],  
   [-76.909393, 38.995548],  
   [-77.119759, 38.995548]]],  
 'type': 'Polygon'},  
 'country': 'United States',  
 'country_code': 'US',  
 'full_name': 'Washington, DC',  
 'id': '01fbe706f872cb32',  
 'name': 'Washington',  
 'place_type': 'city',  
 'url': 'http://api.twitter.com/1/geo/id/0172cb32.json'}
```

In [89]:

```
d5['country']
```

Out[89]:

```
'United States'
```

In [90]:

```
d5['bounding_box']
```

Out[90]:

```
{'coordinates': [[[-77.119759, 38.791645],  
   [-76.909393, 38.791645],  
   [-76.909393, 38.995548],  
   [-77.119759, 38.995548]]],  
 'type': 'Polygon'}
```

In [91]:

```
d5['bounding_box']['coordinates']
```

Out[91]:

```
[[-77.119759, 38.791645],  
 [-76.909393, 38.791645],  
 [-76.909393, 38.995548],  
 [-77.119759, 38.995548]]
```

In [92]:

```
type(d5['bounding_box']['coordinates'])
```

Out[92]:

```
list
```


In [93]:

```
d5['bounding_box']['coordinates'][0]
```

Out[93]:

```
[[-77.119759, 38.791645],  
 [-76.909393, 38.791645],  
 [-76.909393, 38.995548],  
 [-77.119759, 38.995548]]
```

In [94]:

```
d5['bounding_box']['coordinates'][0][0]
```

Out[94]:

```
[-77.119759, 38.791645]
```

In [95]:

```
d5['bounding_box']['coordinates'][0][1]
```

Out[95]:

```
[-76.909393, 38.791645]
```

In [96]:

```
d5['bounding_box']['coordinates'][0][3]
```

Out[96]:

```
[-77.119759, 38.995548]
```

In [97]:

```
latitude1 = d5['bounding_box']['coordinates'][0][0][0]  
longitude1 = d5['bounding_box']['coordinates'][0][0][1]  
print(latitude1, longitude1)
```

```
-77.119759 38.791645
```

In [98]:

```
latitude2 = d5['bounding_box']['coordinates'][0][1][0]  
longitude2 = d5['bounding_box']['coordinates'][0][1][1]  
print(latitude2, longitude2)
```

```
-76.909393 38.791645
```

Exercício: Acessar o twitter e exibir a seguinte saída:

Link: <https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/overview/tweet-object>
(<https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/overview/tweet-object>)

País:

Coordenadas:

latitude 1 = ... , longitude 1 = ...

latitude 2 = ... , longitude 2 = ...

latitude 3 = ... , longitude 3 = ...

latitude 4 = ... , longitude 4 = ...

In [99]:

```
objectJsonTwitter= {
    "bounding_box":
        {
            "coordinates":
                [[
                    [-77.119759,38.791645],
                    [-76.909393,38.791645],
                    [-76.909393,38.995548],
                    [-77.119759,38.995548]
                ]],
            "type": "Polygon"
        },
    "country": "United States",
    "country_code": "US",
    "full_name": "Washington, DC",
    "id": "01fbe706f872cb32",
    "name": "Washington",
    "place_type": "city",
    "url": "http://api.twitter.com/1/geo/id/0172cb32.json"
}
```

In [101]:

```
pais = objectJsonTwitter['country']
```

In [107]:

```
objectJsonTwitter['bounding_box']['coordinates'][0][0]
```

Out[107]:

```
[-77.119759, 38.791645]
```

In [109]:

```
latitude1, longitude1 = objectJsonTwitter['bounding_box']['coordinates'][0][0][0], objectJsonTwitter['bounding_box']['coordinates'][0][0][1]
latitude2, longitude2 = objectJsonTwitter['bounding_box']['coordinates'][0][1][0], objectJsonTwitter['bounding_box']['coordinates'][0][1][1]
latitude3, longitude3 = objectJsonTwitter['bounding_box']['coordinates'][0][2][0], objectJsonTwitter['bounding_box']['coordinates'][0][2][1]
latitude4, longitude4 = objectJsonTwitter['bounding_box']['coordinates'][0][3][0], objectJsonTwitter['bounding_box']['coordinates'][0][3][1]
```

In [114]:

```
print('Saida:')
print('\tPais:', pais)
print('\tCoordenadas: ')
print('\t\tLatitude 1 e Longitude 1: ', latitude1, longitude1)
print('\t\tLatitude 2 e Longitude 2: ', latitude2, longitude2)
print('\t\tLatitude 3 e Longitude 3: ', latitude3, longitude3)
print('\t\tLatitude 4 e Longitude 4: ', latitude4, longitude4)
```

Saida:

Pais: United States

Coordenadas:

Latitude 1 e Longitude 1: -77.119759 38.791645

Latitude 2 e Longitude 2: -76.909393 38.791645

Latitude 3 e Longitude 3: -76.909393 38.995548

Latitude 4 e Longitude 4: -77.119759 38.995548

In []: