

In [22]: *# 01 - Crie um FUNÇÃO que mprema a sequenca de números PARES entre 1 a 30,  
# e depois faça a chamada a função para listar apenas os numeros pares*

```
def listaPar():  
    for i in range(2,30,2):  
        print(i)
```

```
listaPar()
```

2  
4  
6  
8  
10  
12  
14  
16  
18  
20  
22  
24  
26  
28

In [35]: *# 02 - Crie um FUNÇÃO que mprema a sequenca de números IMPARES entre 1 a 30,  
# e depois faça a chamada a função para listar apenas os numeros pares*

```
def listaImpar():  
    for i in range(1,30,2):  
        print (i)
```

```
listaImpar()
```

1  
3  
5  
7  
9  
11  
13  
15  
17  
19  
21  
23  
25  
27  
29

```
In [31]: # FUNÇÃO
def nome ( ):
    print('Edison Neves Soares')
nome()
```

Edison Neves Soares

```
In [21]: # 03 - Crie uma FUNÇÃO que receba uma string como argumento e retorne a mesma string
# faça uma chamada a função, passando como parâmetro uma string

def listaString (feira):
    print(feira.upper())
    return
listaString ('Laranja, banana, limão, uva, cenoura')
```

LARANJA, BANANA, LIMÃO, UVA, CENOURA

```
In [31]: # 04 - Criar uma função que recebe como parâmetro uma lista de cinco elementos, e
# fazer duas chamadas a função, uma com um elemento e outra com 4 elementos

def listaFeira (feira):
    print(feira.append('banana'))
    print(feira.append('cenoura'))

feira1= ['larana', 'limao', 'tomate', 'uva', 'maça']
listaFeira (feira1)
print(feira1)
```

None

None

['larana', 'limao', 'tomate', 'uva', 'maça', 'banana', 'cenoura']

```
In [36]: # 5 - Criar uma função que receba um argumento formal e uma possível lista de elementos
# Fazer duas chamadas a função com apenas um elemento e na segunda com 4 elementos
# Fazendo a chamada

def printNum(arg1, *lista):
    print (arg1)
    for i in lista:
        print(i)
    return;
printNum(150)
printNum('a','b','c','d')
```

150

a

b

c

d

```
In [40]: # 6 - Crie uma função e atribua ser retorno a uma variável chamada soma. a expressão  
# expressões anonimas, tb lambda sem o uso da def  
  
soma = lambda arg, arg1: arg + arg1  
print ( 'A soma entre os argumentos é ', soma (556546, 7848993))
```

A soma entre os argumentos é 8405539

```
In [50]: # 7 - Execute o código e certifique se que compreende a diferença entre a variável  
  
total = 0  
  
def soma( arg, arg1):  
    total = arg + arg1;  
    print('Resultado dentro da Função é: ', total)  
    return total  
  
soma (155245, 6746836)  
print ('Resultado fora da Função é:', total)
```

Resultado dentro da Função é: 6902081

Resultado fora da Função é: 0

```
In [51]: # 8 - Crie uma função anônima para conversão de temperaturas em graus Celsius.  
Celsius = [38.7, 35.6, 37.2, 38.8]  
faherntheit = map(lambda x: (float(9)/5)*x + 32, Celsius)  
print(list(faherntheit))
```

[101.66000000000001, 96.08, 98.96000000000001, 101.84]

In [52]: *# 9 - Criar um dicionário e listar os metodos e atributos deste dicionário:*

```
dic = {'k1': 'São Paulo', 'k2': 'Ceara'}  
dir(dic)
```

Out[52]:

```
['__class__',  
 '__contains__',  
 '__delattr__',  
 '__delitem__',  
 '__dir__',  
 '__doc__',  
 '__eq__',  
 '__format__',  
 '__ge__',  
 '__getattribute__',  
 '__getitem__',  
 '__gt__',  
 '__hash__',  
 '__init__',  
 '__init_subclass__',  
 '__iter__',  
 '__le__',  
 '__len__',  
 '__lt__',  
 '__ne__',  
 '__new__',  
 '__reduce__',  
 '__reduce_ex__',  
 '__repr__',  
 '__setattr__',  
 '__setitem__',  
 '__sizeof__',  
 '__str__',  
 '__subclasshook__',  
 'clear',  
 'copy',  
 'fromkeys',  
 'get',  
 'items',  
 'keys',  
 'pop',  
 'popitem',  
 'setdefault',  
 'update',  
 'values']
```

In [55]: *# 10 - import pandas as pd*

```
import pandas as pd  
pd.__version__
```

Out[55]: '0.25.1'

