

Data Science Academy - Python Fundamentos - Capítulo 3

Download: <http://github.com/dsacademybr>

```
In [ ]: # Versão da Linguagem Python
        from platform import python_version
        print('Versão da Linguagem Python Usada Neste Jupyter Notebook:', python_version())
```

Exercícios - Métodos e Funções

```
In [ ]: # Exercício 1 - Crie uma função que imprima a sequência de números pares entre 1 e 2
        # depois faça uma chamada à função para listar os números
```

```
In [1]: #Tentativa 1:
        def par():
            return list(range(2,21,2)) # Tipo lista
        par()
```

```
Out[1]: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

```
In [127... #Resposta do DSA
def par():
    for i in range(2, 21, 2): # Sequência
        print(i)
par()
```

2
4
6
8
10
12
14
16
18
20

```
In [2]: # Exercício 2 - Crie uma função que receba uma string como argumento e retorne a mes
        # Faça uma chamada à função, passando como parâmetro uma string
```

```
In [128... #Tentativa 1
def caracteres(frase):
    print(frase.upper())

fr="Onde está a sua felicidade?"
caracteres(fr)
```

ONDE ESTÁ A SUA FELICIDADE?

```
In [6]: #Tentativa 2: Utilizando input
```

```
Nome=input("Escreva o seu nome: ")
def nome(frase):
    print(frase.upper())
nome(Nome)
```

Escreva o seu nome: Carla Caroline Carvalho Silva
CARLA CAROLINE CARVALHO SILVA

```
In [4]: #Resposta DSA
def listaString(texto):
    print(texto.upper())
    return

listaString('Rumo à Análise de Dados')
```

ONDE ESTÁ A SUA FELICIDADE?

```
In [ ]: # Exercício 3 - Crie uma função que receba como parâmetro uma lista de 4 elementos,
# imprima a lista
```

```
In [7]: #Tentativa 1: Utilizando extend
def add_fruta(lista):
    frutas=["manga","abacaxi"]
    lista.extend(frutas)
    return print(lista)
```

```
In [8]: lst=["morango","pêra","uva","maça"]
add_fruta(lst)

['morango', 'pêra', 'uva', 'maça', 'manga', 'abacaxi']
```

```
In [129... #Tentativa 2: Utilizando append
def add_fruta(lista):
    lista.append("manga")
    lista.append("abacaxi")
    return print(lista)
```

```
In [130... lst=["morango","pêra","uva","maça"]
add_fruta(lst)

['morango', 'pêra', 'uva', 'maça', 'manga', 'abacaxi']
```

```
In [131... #Tentativa 3: Utilizando extend e input
def add_fruta(lista):
    f1=input("Escreva o nome da fruta: ")
    f2=input("Escreva o nome da fruta: ")
    frutas=[f1,f2]
    lista.extend(frutas)
    return print(lista)
```

```
In [132... lst=["morango","pêra","uva","maça"]
add_fruta(lst)

Escreva o nome da fruta: manga
Escreva o nome da fruta: abacaxi
['morango', 'pêra', 'uva', 'maça', 'manga', 'abacaxi']
```

```
In [133... #Tentativa 3: Utilizando extend e input com menos linhas de código
def add_fruta(lista):
    frutas=input("Escreva o nome da fruta: "),input("Escreva o nome da fruta: ")
    lista.extend(frutas)
    return print(lista)
```

```
In [134... lst=["morango", "pêra", "uva", "maça"]
add_fruta(lst)

Escreva o nome da fruta: manga
Escreva o nome da fruta: abacaxi
['morango', 'pêra', 'uva', 'maça', 'manga', 'abacaxi']
```

```
In [ ]: # Exercício 4 - Crie uma função que receba um argumento formal e uma possível lista
# à função, com apenas 1 elemento e na segunda chamada com 4 elementos
```

```
In [135... #Tentativa 1
def turma_alunos(turma,*alunos):
    print(turma)
    for i in alunos:
        print(i)
```

```
In [139... turma_alunos(2)
```

2

```
In [140... turma_alunos(2, "Pedro", "Marcos", "Jose")
```

2
Pedro
Marcos
Jose

```
In [141... #Resposta DSA
def printNum( arg1, *lista ):
    print (arg1)
    for i in lista:
        print (i)
    return;

# Chamada à função
printNum( 100 )
printNum( 'A', 'B', 'C' )
```

100
A
B
C

```
In [ ]: # Exercício 5 - Crie uma função anônima e atribua seu retorno a uma variável chamada
# números como parâmetro e retornar a soma deles
```

```
In [22]: Soma=lambda x,y:x+y
Soma(3,4)
```

Out[22]: 7

```
In [142... # Resposta DSA
soma = lambda arg1, arg2: arg1 + arg2
print ("A soma é : ", soma( 452, 298 ))
```

A soma é : 750

```
In [23]: # Exercício 6 - Execute o código abaixo e certifique-se que compreende a diferença e
total = 0
def soma( arg1, arg2 ):
    total = arg1 + arg2;
    print ("Dentro da função o total é: ", total) #variável local
    return total;

soma( 10, 20 );
print ("Fora da função o total é: ", total) #variável global
```

Dentro da função o total é: 30

Fora da função o total é: 0

```
In [26]: # Exercício 7 - Abaixo você encontra uma lista com temperaturas em graus Celsius
# Crie uma função anônima que converta cada temperatura para Fahrenheit
# Dica: para conseguir realizar este exercício, você deve criar sua função lambda, d
# (que será estudada no próximo capítulo). Isso permite aplicar sua função a cada el
# Como descobrir a fórmula matemática que converte de Celsius para Fahrenheit? Pesqu
```

```
In [43]: Celsius = [39.2, 36.5, 37.3, 37.8]
Fahrenheit=map(lambda c: (c*9/5)+32, Celsius)
print (list(Fahrenheit))
```

[102.56, 97.7, 99.14, 100.03999999999999]

```
In [ ]: #Exercício 8
#Crie um dicionário e liste todos os métodos e atributos do dicionário
```

```
In [143... Dic={"Fruta":"Pêra", "Idade":26, "Nome":"Carla"}
dir(Dic)
```

```
Out[143... ['__class__',
 '__class_getitem__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattr__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__ior__',
 '__iter__',
```

```
'__le__',
'__len__',
'__lt__',
'__ne__',
'__new__',
'__or__',
'__reduce__',
'__reduce_ex__',
'__repr__',
'__reversed__',
'__ror__',
'__setattr__',
'__setitem__',
'__sizeof__',
'__str__',
'__subclasshook__',
'clear',
'copy',
'fromkeys',
'get',
'items',
'keys',
'pop',
'popitem',
'setdefault',
'update',
'values']
```

In [126...

```
# Exercício 9
# Abaixo você encontra a importação do Pandas, um dos principais pacotes Python para
# Analise atentamente todos os métodos disponíveis. Um deles você vai usar no próximo
import pandas as pd
dir(pd)
```

Out[126...

```
['BooleanDtype',
'Category',
'CategoryDtype',
'CategoryIndex',
'DataFrame',
'DateOffset',
'DatetimeIndex',
'DatetimeTZDtype',
'ExcelFile',
'ExcelWriter',
'Flags',
'Float32Dtype',
'Float64Dtype',
'Float64Index',
'Grouper',
'HDFStore',
'Index',
'IndexSlice',
'Int16Dtype',
'Int32Dtype',
'Int64Dtype',
'Int64Index',
'Int8Dtype',
'Interval',
'IntervalDtype',
'IntervalIndex',
'MultiIndex',
'NA',
'NaT',
```

```
'NamedAgg',
'Period',
'PeriodDtype',
'PeriodIndex',
'RangeIndex',
'Series',
'SparseDtype',
'StringDtype',
'Timedelta',
'TimedeltaIndex',
'Timestamp',
'UInt16Dtype',
'UInt32Dtype',
'UInt64Dtype',
'UInt64Index',
'UInt8Dtype',
'__builtins__',
'__cached__',
'__doc__',
'__docformat__',
'__file__',
'__getattr__',
'__git_version__',
'__loader__',
'__name__',
'__package__',
'__path__',
'__spec__',
'__version__',
'_config',
'_hashtable',
'_is_numpy_dev',
'_lib',
'_libs',
'_np_version_under1p18',
'_testing',
'_tslib',
'_typing',
'_version',
'api',
'array',
'arrays',
'bdate_range',
'compat',
'concat',
'core',
'crosstab',
'cut',
'date_range',
'describe_option',
'errors',
'eval',
'factorize',
'get_dummies',
'get_option',
'infer_freq',
'interval_range',
'io',
'isna',
'isnull',
'json_normalize',
'lreshape',
'melt',
'merge',
```

```
'merge_asof',  
'merge_ordered',  
'notna',  
'notnull',  
'offsets',  
'option_context',  
'options',  
'pandas',  
'period_range',  
'pivot',  
'pivot_table',  
'plotting',  
'qcut',  
'read_clipboard',  
'read_csv',  
'read_excel',  
'read_feather',  
'read_fwf',  
'read_gbq',  
'read_hdf',  
'read_html',  
'read_json',  
'read_orc',  
'read_parquet',  
'read_pickle',  
'read_sas',  
'read_spss',  
'read_sql',  
'read_sql_query',  
'read_sql_table',  
'read_stata',  
'read_table',  
'read_xml',  
'reset_option',  
'set_eng_float_format',  
'set_option',  
'show_versions',  
'test',  
'testing',  
'timedelta_range',  
'to_datetime',  
'to_numeric',  
'to_pickle',  
'to_timedelta',  
'tseries',  
'unique',  
'util',  
'value_counts',  
'wide_to_long']
```

Fim

Obrigado

Visite o Blog da Data Science Academy - [Blog DSA](#)