

# Lista de Exercícios - João Alonso Casella

João Alonso Casella - Trainee Quant 2022-1 |Feito em Jupyter Notebook

## Lista 1 - Python Básico

In [1]:

```
import math as m
```

In [70]:

```
# 1a)
print('Exercício 1: ')
print('Digite seu nome:')
nome = input()
print('Olá, ' + nome, ". Digite sua idade e seu semestre (número apenas): ")
idade=input()
semestre=input()
print(nome, ', qual é o seu curso?')
curso = input()
print('Oi, meu nome é ' + nome + ', tenho ' + idade + ' anos, e estou no ' + semestre + 'º s
emestre de ' + curso + ' no Insper.')
```

Exercício 1:

Digite seu nome:

Marcela

Olá, Marcela . Digite sua idade e seu semestre (número apenas):

19

3

Marcela , qual é o seu curso?

Economia

Oi, meu nome é Marcela, tenho 19 anos, e estou no 3º semestre de Economia no Insper.

In [71]:

```
# 1b)
frase = 'Oi, meu nome é ' + nome + ', tenho ' + idade + ' anos, e estou no ' + semestre + 'º
semestre de ' + curso + ' no Insper.'
print("Posição do nome: ", frase.find(nome))
print("Posição da idade: ", frase.find(idade))
print("Número de caracteres no nome: ", len(nome))
```

Posição do nome: 15

Posição da idade: 30

Número de caracteres no nome: 7

In [72]:

```
# 1c)
print( 'Oi,', '\nmeu nome é '+ nome +', \ntenho '+ idade + ' anos, \ne estou no '+ semes
tre + 'º semestre de '+ curso + ' no Insper.')
```

Oi,  
meu nome é Marcela,  
tenho 19 anos,  
e estou no 3º semestre de Economia no Insper.

In [7]:

```
# 2a)
print('Exercício 2: ')
a = 1875+321
print('a'),a)
# 2b)
b = 2641-469
print('b'),b)
# 2c)
c=475/25
print('c'),round(c,))
# 2d)
d=36*7
print('d'),d)
# 2e)
e=7**4
print('e'),e)
# 2f)
f=abs(-8)
print('f'),f)
# 2g)
g = 785%3
print('g'),g)
# 2h)
h=641//6
print('h'),h)
# 2i)
i=m.sqrt(19881)
print('i'),round(i,))
# 2j)
j=round(8*1.33,)
print('j'),j)
```

Exercício 2:

- a) 2196
- b) 2172
- c) 19
- d) 252
- e) 2401
- f) 8
- g) 2
- h) 106
- i) 141
- j) 11

In [ ]:

```
# 3a)
print('Exercício 3')
a3= [(3**2),(2**4)]
max_a3 = max(a3)
min_a3 = min(a3)
print('a) Máximo: ',max_a3,'; Mínimo: ',min_a3)

# 3b)
b3 = [round(25/5,),round(28/7,)]
max_b3 = max(b3)
min_b3 = min(b3)
print('b) Máximo: ',max_b3,'; Mínimo: ',min_b3)

# 3c)
c3 = [(6*9),(8*7)]
max_c3 = max(c3)
min_c3 = min(c3)
print('c) Máximo: ',max_c3,'; Mínimo: ',min_c3)
```

In [1]:

```
# 4a)
carteira=['VVAR3.SA','SLCE3.SA','BBSE3.SA','EC0011.SA','PETZ3.SA']
print("Adicione um ativo: ")
carteira.append(input())
continuar = input("Deseja continuar? (s/n)")
while continuar == 's':
    carteira.append(input())
    continuar = input("Deseja continuar? (s/n)")

print(carteira)
```

Adicione um ativo:

PETR4

Deseja continuar? s

VALE3

Deseja continuar? s

BRK.SA

Deseja continuar? n

```
['VVAR3.SA', 'SLCE3.SA', 'BBSE3.SA', 'EC0011.SA', 'PETZ3.SA', 'PETR4', 'VALE3', 'BRK.SA']
```

In [2]:

```
# 4b)
print("Qual ativo se deseja remover?")
x = carteira.index(input())
del(carteira[x])
print(carteira)
continuar = input("Deseja continuar? (s/n)")
while continuar == 's':
    x = carteira.index(input())
    del(carteira[x])
    continuar = input("Deseja continuar? (s/n)")

print(carteira)
```

Qual ativo se deseja remover?

EC0011.SA

['VVAR3.SA', 'SLCE3.SA', 'BBSE3.SA', 'PETZ3.SA', 'PETR4', 'VALE3', 'BRK.SA']

Deseja continuar? s

BBSE3.SA

Deseja continuar? n

['VVAR3.SA', 'SLCE3.SA', 'PETZ3.SA', 'PETR4', 'VALE3', 'BRK.SA']

In [28]:

```
# 5a)
def ret():
    lst=[]

    print("Insira o valor no tempo inicial: ")
    a = int(input())
    lst.append(a)

    print(lst)

    print("\nAgora insira o novo valor: ")
    b = int(input())
    lst.append(b)

    r=(lst[1]-lst[0])/lst[0]

    return r

x = ret()
print(x)

# 5b)
if x > 0:
    print("Retorno Positivo")
elif x == 0:
    print("Retorno = 0")
else:
    print("Retorno Negativo")
```

Insira o valor no tempo inicial:

2

[2]

Agora insira o novo valor:

2

0.0

Retorno = 0

In [7]:

```
# 6)

print("Olá! Você possui algum animal de estimação? (s/n)")
resposta1 = []
resposta2 = []
resposta3 = []

resposta1.append(input())
if resposta1[0] == 's':
    print("Ok. Gostaria de ter mais? (s/n)")
    resposta2.append(input())
    if resposta2[0] == 's':
        print('Perfeito! Recomenda-se a adoção!')
    else:
        print('Ok. Então continue cuidando dos que já possui.')
else:
    print('Ok. Quer ter algum dia? (s/n)')
    resposta3.append(input())
    if resposta3[0] == 's':
        print('Ok. Boa sorte, algum dia você terá um animal de estimação!')
    else:
        print('Ok... entendi.Você não vai ter animais de estimação.')
```

```
Olá! Você possui algum animal de estimação? (s/n)
s
Ok. Quer ter algum dia? (s/n)
s
Ok... entendi.Você não vai ter animais de estimação.
```

In [27]:

```
# 7)

ing = {}
ing["Hi"]="Olá"
ing["How are you?"]="Como você está?"
ing["How old are you?"]="Quantos anos você tem?"
ing["Brazilian"]="Brasileiro(a)"
ing["Financial market"]="Mercado financeiro"
ing["Money"]="Dinheiro"
print("=====TRADUTOR =====")
print("\nBem-vindo ao tradutor de Ingles para Português! \n\nEscreva a frase que se deseja traduzir:")
x = ing[input()]

print("\nTradução: ", x)
```

```
=====TRADUTOR =====
```

Bem-vindo ao tradutor de Ingles para Português!

Escreva a frase que se deseja traduzir:

Money

Tradução: Dinheiro

In [42]:

```
# 8)
print("Coloque o número desejado:")
x = int(input())
x2 = x**2
while x+10 < x2:
    x=x+10
    print(x)
print('=====FIM=====')
```

Coloque o número desejado:

12

22

32

42

52

62

72

82

92

102

112

122

132

142

```
=====FIM=====
```

In [8]:

```
# 9)

try:
    print("Insira sua nota da Atividade Prática Supervisionada: ")
    aps = int(input())
    print("Insira sua nota da Prova Intermediária: ")
    pi = int(input())
    print("Insira sua nota da Prova Final: ")
    pf = int(input())
    x = (aps+pi+pf)/3

    while x > 0 and x<=10:
        print("Média aluno: ",x)
        if x<5:
            print("Média Baixa! Reprovado!")
        elif x==5:
            print("Está na Média! Passou raspando!")
        else:
            print("Acima da Média! Passou com sobra!")
        break
except:
    print("Erro: por favor, coloque um número entre 0 e 10.")
```

Insira sua nota da Atividade Prática Supervisionada:

e

Erro: por favor, coloque um número entre 0 e 10.

## Lista 2 - Pandas



In [1]:

```
# importando bibliotecas
import pandas as pd
df = pd.read_excel("basededados_listapandas.xlsx")
df
```

Out[1]:

	TICKER	Setor	Close	Classe de Ativo	Float	Variação	Funcionários	índice BETA
0	NaN	Beverages	14.71	Ações	15744670000	-0.004079	53000.0	0.7236
1	NaN	Textiles & Apparel	22.17	Ações	683062200	0.003157	17000.0	0.8043
2	NaN	Diversified Retail	28.36	Ações	924608900	-0.012341	1500.0	1.2025
3	NaN	Food & Drug Retailing	16.03	Ações	1346914000	0.019339	50000.0	NaN
4	NaN	Passenger Transportation Services	22.73	Ações	333680000	0.032556	11968.0	1.7894
...	...	...	...	...	...	...	...	..
78	NaN	Oil & Gas	22.67	Ações	1165000000	-0.007940	3370.0	1.1908
79	NaN	Diversified Retail	3.44	Ações	1598426000	-0.008721	53101.0	2.0753
80	NaN	Telecommunications Services	53.45	Ações	1676938000	-0.010290	34000.0	0.1819
81	NaN	Machinery, Equipment & Components	31.91	Ações	4197318000	-0.016923	36987.0	0.5488
82	NaN	Miscellaneous Educational Service Providers	17.31	Ações	309088800	0.013865	17726.0	1.7052

83 rows × 9 columns



In [2]:

```
# 1)
print(type(df["Setor"]))
#ou
print(type(df.Setor))
```

```
<class 'pandas.core.series.Series'>
<class 'pandas.core.series.Series'>
```

In [3]:

```
# 2)
# A coluna "Classe de Ativo" é redundante, todos os ativos na planilha são sobre ações
df.drop(['Classe de Ativo'],axis=1,inplace=True)
```

In [4]:

```
# 3)
# Maior volume de transação
print("Maior volume de transação: ", max(df.iloc[:,7]))
# Menor volume de transação
print('Menor volume de transação: ', min(df.iloc[:,7]))
```

Maior volume de transação: 88289000

Menor volume de transação: 1280400

In [48]:

```
# 4)

df_fun = df["Funcionários"]
df_fun.dropna(inplace=True)
round(df_fun.mean(),2)
```

Out[48]:

30551.56

In [6]:

```
# 5)

df["Abertura"]=df["Close"]/(1+df["Variação"])
df
```

Out[6]:

	TICKER	Setor	Close	Float	Variação	Funcionários	índice BETA	Volume de Tradição
0	NaN	Beverages	14.71	15744670000	-0.004079	53000.0	0.7236	259468
1	NaN	Textiles & Apparel	22.17	683062200	0.003157	17000.0	0.8043	57779
2	NaN	Diversified Retail	28.36	924608900	-0.012341	1500.0	1.2025	32412
3	NaN	Food & Drug Retailing	16.03	1346914000	0.019339	50000.0	NaN	77530
4	NaN	Passenger Transportation Services	22.73	333680000	0.032556	11968.0	1.7894	41179
...	...	...	...	...	...	...	...	...
78	NaN	Oil & Gas	22.67	1165000000	-0.007940	3370.0	1.1908	46530
79	NaN	Diversified Retail	3.44	1598426000	-0.008721	53101.0	2.0753	360580
80	NaN	Telecommunications Services	53.45	1676938000	-0.010290	34000.0	0.1819	20180
81	NaN	Machinery, Equipment & Components	31.91	4197318000	-0.016923	36987.0	0.5488	120800
82	NaN	Miscellaneous Educational Service Providers	17.31	309088800	0.013865	17726.0	1.7052	25010

83 rows × 9 columns



In [7]:

```
# 6)

df.Setor.iloc[df["Volume de Tradição"].idxmax()]
```

Out[7]:

'Oil & Gas'

In [8]:

```
# 7)
df_float = df[df.Float>=10000000]
df_f = df_float["Setor"]
df_f
```

Out[8]:

```
0          Beverages
1    Textiles & Apparel
2    Diversified Retail
3    Food & Drug Retailing
4    Passenger Transportation Services
...
78         Oil & Gas
79    Diversified Retail
80    Telecommunications Services
81    Machinery, Equipment & Components
82    Miscellaneous Educational Service Providers
Name: Setor, Length: 83, dtype: object
```

In [26]:

```
# 8)
# descobri que pode usar | como sinônimo do "+" que se usa na função print(), para quando se quer fazer um filtro que, para uma mesma coluna, escolha mais de um nome.
df_filtro = df[(df["Setor"] == "Banking Services") | (df["Setor"] == "Electrical Utilities & IPPs") | (df["Setor"] == "Food & Tobacco")]
df_filtro
df_f2 = df_filtro.loc[(df_filtro["Funcionários"] > 10000) | (df_filtro["Float"] < 2000000), "índice BETA"]
print(df_f2)
print("O menor Índice BETA após os filtros é: \nBeta =", min(df_f2))
```

```
6      1.3368
7      1.2646
8      1.2646
13     1.2706
22     0.7051
32     1.4215
33     1.4215
46     1.0865
47     1.0631
48     0.4761
54     0.5636
Name: índice BETA, dtype: float64
O menor Índice BETA após os filtros é:
Beta = 0.4761
```

In [31]:

```
# 9)
df.loc[df["índice BETA"] == 2.1356, "Setor"]
```

Out[31]:

```
42    Passenger Transportation Services
Name: Setor, dtype: object
```

In [75]:

```
# 10)
x = df["Funcionários"]
x.dropna(inplace=True)
y = df["Float"]
y.dropna(inplace=True)

z = round(x.corr(y),2)

print("=====
====\nCorrelação: ",z)

print("=====
====\nSozinho o valor não possui sentido prático. \nA correlação pode muitas vezes ser
mal-interpretada como sinônimo de causalidade. \nRecomenda-se uma análise mais abrangente da relação entre as duas variáveis.\n=====
=====")
```

```
=====
=====
Correlação:  0.33
=====
=====
Sozinho o valor não possui sentido prático.
A correlação pode muitas vezes ser mal-interpretada como sinônimo de causalidade.
Recomenda-se uma análise mais abrangente da relação entre as duas variáveis.
=====
=====
```

In [87]:

```
# 11)
df.rename(columns = {"Close":"Fechamento"},inplace=True)
column_names = ["Setor","Fechamento","Float","Variação","Funcionários","Volume de Trasação","índice BETA","Abertura"]
df=df.reindex(columns=column_names)
df.sort_values(by="Float",ascending=False,inplace=True)
df
```

Out[87]:

	Setor	Fechamento	Float	Variação	Funcionários	Volume de Trasação	índice BETA	Ab
0	Beverages	14.71	15744670000	-0.004079	53000.0	25946800	0.7236	14.7
59	Oil & Gas	34.12	13044500000	0.017292	45532.0	14921300	1.5059	33.5
60	Oil & Gas	30.72	13044500000	0.030273	45532.0	88289000	1.5059	29.8
20	Metals & Mining	5.81	11182490000	0.000000	7432.0	3090800	NaN	5.8
47	Banking Services	26.48	9804136000	-0.017749	99600.0	34387000	1.0631	26.9
...	...	...	...	...	...	...	...	...
58	Food & Drug Retailing	23.78	269395400	-0.000841	110000.0	1614100	NaN	23.8
38	Real Estate Operations	16.56	227000000	0.015700	338.0	1280400	1.3240	16.3
27	Hotels & Entertainment Services	14.33	224934800	0.011165	2000.0	10675500	1.9679	14.1
69	Oil & Gas	44.71	202593100	-0.032431	NaN	3342700	NaN	46.2
62	Computers, Phones & Household Electronics	8.66	141800000	0.010393	4000.0	1425700	1.6886	8.5

83 rows × 8 columns

Obrigado! Bom Econo!

=====

In [ ]: