

# GABARITO DOS EXERCÍCIOS

dnc

Segue abaixo gabarito dos exercícios dados em aula:

## VARIÁVEIS

1. Na célula abaixo crie uma variável de cada tipo aprendido e atribua um valor para cada uma delas, lembre-se de utilizar as dicas da aula.

```
inteiro = 5 #um numero qualquer sem "." e sem estar dentro de '' ou ""  
real = 4.5 # um numero qualquer com o "." e sem estar dentro de '' ou ""  
texto = 'qualquer texto dentro de aspas simples ou duplas'
```

2. Na célula abaixo imprima os valores atribuídos em cada variável, pula uma linha de uma variável para a outra.

+ Código

+ Texto

```
print(inteiro, '\n', real, '\n', texto)
```

```
5  
4.5
```

3. Na Célula abaixo imprima os tipos de cada variável.

```
print(inteiro, type(inteiro), real, type(real), texto, type(texto))
```

```
5 <class 'int'> 4.5 <class 'float'> qualquer texto dentro de aspas simples ou duplas <cl
```

## OPERAÇÕES

1. Faça pelo menos 1 exemplo com cada operador visto utilizando variáveis do mesmo tipo.

```
inteiro_1 = 3  
inteiro_2 = 4
```

```
print(inteiro_1 / inteiro_2)  
print(inteiro_1 // inteiro_2)  
print(inteiro_1 % inteiro_2)  
print(inteiro_1 * inteiro_2)  
print(inteiro_1 ** inteiro_2)  
print(inteiro_1 + inteiro_2)  
print(inteiro_1 - inteiro_2)  
print(inteiro_1 > inteiro_2)  
print(inteiro_1 == inteiro_2)  
print(inteiro_1 != inteiro_2)
```

```
0.75  
0  
3  
12  
81  
7  
-1  
False  
False  
True
```

Segue abaixo gabarito dos exercícios dados em aula:

2. Repita o exercício 1 mas dessa vez utilizando variáveis de tipos diferentes, dessa vez verifique se o tipo dela alterou ou não.

```
inteiro_1 = 3
inteiro_2 = 4.3
```

```
print(inteiro_1 / inteiro_2)
print(inteiro_1 // inteiro_2)
print(inteiro_1 % inteiro_2)
print(inteiro_1 * inteiro_2)
print(inteiro_1 ** inteiro_2)
print(inteiro_1 + inteiro_2)
print(inteiro_1 - inteiro_2)
print(inteiro_1 > inteiro_2)
print(inteiro_1 == inteiro_2)
print(inteiro_1 != inteiro_2)
```

```
0.6976744186046512
0.0
3.0
12.899999999999999
112.62152279558863
7.3
-1.2999999999999998
False
False
True
```

3. Faça 3 operações com operadores diferentes utilizando o formato passado na em Dicas.

```
print(inteiro_1)
inteiro_1+=1
print(inteiro_1)
inteiro_1 -= 1
print(inteiro_1)
inteiro_1 *= 2
print(inteiro_1)
inteiro_1 /= 2
print(inteiro_1)
```

```
5
6
5
10
5.0
```



Segue abaixo gabarito dos exercícios dados em aula:

## CONDICIONAIS

1. Construa uma condição para verificar se o número é múltiplo de 3.

```
inteiro = 9
if inteiro % 3==0:
    print('é múltiplo de 3')
```

☞ é múltiplo de 3

2. Crie 3 condicionais utilizando a estrutura elif.

```
inteiro = 11
if inteiro % 3==0:
    print('é múltiplo de 3')
elif inteiro % 4==0:
    print('é múltiplo de 4')
elif inteiro % 2==0:
    print('é par')
else:
    print('é ímpar')
```

3. Crie uma lista e verifique se ela contém os números 7 e 3.

```
lista =[1,2,3,4,2,4,7,6,5,3,7]
if 7 in lista and 3 in lista:
    print('contém 7 e 3')
elif 3 in lista:
    print('contém 3')
elif 7 in lista:
    print('contém 7')
else:
    print('não contém nem o 7 nem o 3')
```

contém 7 e 3

Segue abaixo gabarito dos exercícios dados em aula:

## Desafio:

Crie uma lista com 5 valores e uma tupla com 10 valores, verifique se algum valor da lista está contido na tupla.

```
lista = [77,2,3,4,5]
tupla = [3,2,4,2,5,2,1,3,2,7]
if lista[0] in tupla:
    print(lista[0], 'contém na tupla')
if lista[1] in tupla:
    print(lista[1], 'contém na tupla')
if lista[2] in tupla:
    print(lista[2], 'contém na tupla')
if lista[3] in tupla:
    print(lista[3], 'contém na tupla')
if lista[4] in tupla:
    print(lista[4], 'contém na tupla')
```

```
2 contém na tupla
3 contém na tupla
4 contém na tupla
5 contém na tupla
```

## LOOPS

1. Crie uma lista contendo os números pares de 0 até 1000

```
lista=[]
for a in range(0,1000,2):
    lista.append(a)
lista
```

```
78,
80,
82,
84,
86,
88,
90
```



Segue abaixo gabarito dos exercícios dados em aula:

2. Em uma nova lista, transforme cada valor da lista anterior em um string.

```
lista_2 = []
for a in lista:
    lista_2.append(str(a))
lista_2
```

```
['0',
 '2',
 '4',
 '6',
 '8',
 '10',
 '12',
 '14',
 '16',
 '18',
 '20',
 '22',
 '24',
 '26',
 '28',
 '30',
 '32',
 '34',
 '36',
 '38',
 '40',
 '42',
```

3. Compare os valores entre as listas e some os valores das 2 listas.

```
for a in lista:
    if a in lista_2:
        print(a, 'contém na lista')
    else:
        print(a, 'não contem na lista')
```

```
0 não contem na lista
2 não contem na lista
4 não contem na lista
6 não contem na lista
8 não contem na lista
10 não contem na lista
12 não contem na lista
14 não contem na lista
16 não contem na lista
18 não contem na lista
20 não contem na lista
22 não contem na lista
24 não contem na lista
26 não contem na lista
28 não contem na lista
30 não contem na lista
32 não contem na lista
34 não contem na lista
```

Segue abaixo gabarito dos exercícios dados em aula:

4. Crie um for para reproduzir a [sequência de Fibonacci](#).

```
x = 0
y = 1
for a in range(10):
    print(x)
    b = x + y
    x = y
    y=b
```

```
0
1
1
2
3
5
8
13
21
34
```

5. Crie um for que execute até o usuário digitar "parar".

```
while a != "parar":
    a= input()#código para permitir o usuário digitar
    oi
    para
    parar
```

## LOOPS 2

1. Crie um Loop que percorra uma string e coloque a posição de cada caracter contida nela.

```
texto = 'ola tudo bem'
for a,b in enumerate(texto):
    print(a,b)
```

```
0 o
1 l
2 a
3
4 t
5 u
6 d
7 o
8
9 b
10 e
11 m
```

Segue abaixo gabarito dos exercícios dados em aula:

2. Refaça o exercício 1 mas dessa vez pulando os espaços.

```
texto = 'ola tudo bem'
for a,b in enumerate(texto):
    if b != " ":
        print(a,b)

0 o
1 l
2 a
4 t
5 u
6 d
7 o
9 b
10 e
11 m
```

3. crie duas listas e percorra as duas no mesmo for, se elas tiverem tamanhos diferentes descreva o que acontece.

```
lista1 = [1,2,3,4,5]
lista2 = [1,2,3,4,5,6,7]
for a,b in zip(lista1,lista2):
    print(a,b)
#É percorrido até o ultimo elemento da lista menor

1 1
2 2
3 3
4 4
5 5
```

4. Faça a taboada feita em aula começando do número 9.

```
for a in range(9,0,-1):
    for b in range(10):
        print(a,'x',b,'=',a*b)

6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
5 x 0 = 0
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
4 x 0 = 0
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
```



Segue abaixo gabarito dos exercícios dados em aula:

```
4 x 0 = 0
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
3 x 0 = 0
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
2 x 0 = 0
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
1 x 0 = 0
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
```

## LOOPS 3

1. Percorra cada pet no dicionário acima e imprima sua espécie.

```
for a in itemData:
    print(a['species'])

cat
dog
cat
```

2. Percorra cada pet no dicionário e imprima o pet com maior quantidade de dislikes.

```
pet=''
len1=0
for a in itemData:
```

Segue abaixo gabarito dos exercícios dados em aula:

```
print(len(a['foods']['dislikes']),a['name'])
if len1 < len(a['foods']['dislikes']):
    len1 = len(a['foods']['dislikes'])
pet=a['name']
pet
```

```
2 Meowsy
1 Barky
1 Purrpaws
'Meowsy'
```

3. Conte o total de comidas citadas no dicionário e quantas se repetem.

```
comidas=[]
for a in itemData:
    for b in a['foods']['dislikes']:
        comidas.append(b)
    for c in a['foods']['likes']:
        comidas.append(c)
comidas
```

```
['ham',
 'zucchini',
 'tuna',
 'catnip',
 'tuna',
 'bones',
 'carrots',
 'cookies',
 'mice']
```

```
lista.count(valor)#conta a ocorrência do valor na lista
for a in comidas:
    print(comidas.count(a),a)
```

```
1 ham
1 zucchini
2 tuna
1 catnip
2 tuna
1 bones
1 carrots
1 cookies
1 mice
```

## LOOPS 3



Segue abaixo gabarito dos exercícios dados em aula:

## FUNÇÕES NATIVAS PYTHON

Utilize pelo menos outras 3 funções contidas no link acima.

```
tuple([1,2,3,4,5])
```

```
(1, 2, 3, 4, 5)
```

```
chr(10234)
```

```
'⇔'
```

```
list(reversed(tuple([1,2,3,4,5])))
```

```
[5, 4, 3, 2, 1]
```

## FUNÇÕES

1. Crie uma função que receba o input do cpf e nome do usuário.

```
def cadastro ():  
    print('digite seu nome')  
    nome=input()  
    print('digite seu cpf')  
    cpf=input()  
    print(nome,cpf)  
cadastro()
```

```
digite seu nome  
peter  
digite seu cpf  
1234  
peter 1234
```

2. Agora nessa mesma função coloque o retorno como um dicionário onde a chave é o cpf e o valor o nome do usuário.

```
def cadastro ():  
    print('digite seu nome')  
    nome=input()  
    print('digite seu cpf')  
    cpf=input()  
    dicio =dict({cpf:nome})
```

Segue abaixo gabarito dos exercícios dados em aula:

```
print(dicio)
return dicio
user =cadastro()
print(user)
```

```
digite seu nome
peter
digite seu cpf
21312
{'21312': 'peter'}
{'21312': 'peter'}
```

### 3. Crie uma função para retornar a quantidade de CPF cadastrados

```
def cadastro ():
    cpfs=[]
    continuar=''
    while continuar != 'não':
        print('digite seu nome')
        nome=input()
        print('digite seu cpf')
        cpf=input()
        cpfs.append(cpf)
        print('quer continuar?')
        continuar = input()
    return len(cpfs)
qtd =cadastro()
qtd
```

```
digite seu nome
peter
digite seu cpf
1231
quer continuar?
sim
digite seu nome
pefe
digite seu cpf
asdad
quer continuar?
não
2
```



Segue abaixo gabarito dos exercícios dados em aula:

## FUNÇÕES LAMBDA

1. Crie uma função Lambda para retornar a soma de 4 valores.

```
f1 =lambda a,b,c,d:a+b+c+d  
f1(1,2,3,4)
```

10

2. Utilize uma função filter para retornar somente os números pares da lista abaixo.

[5,2,5,7,4,2,6,10,342,54,23,6,7,9,12]

```
par = list(filter(lambda x: x%2==0,[5,2,5,7,4,2,6,10,342,54,23,6,7,9,12])) # retorna os números  
#printando a variavel par  
print(par)
```

[2, 4, 2, 6, 10, 342, 54, 6, 12]

3. Utilize a função reduce e retorne a soma da lista anterior.

```
from functools import reduce
```

```
#atribuindo o resultado do reduce a uma nova variável e criando uma função lambda para somar  
r = reduce(lambda x,y: x+y,[5,2,5,7,4,2,6,10,342,54,23,6,7,9,12]) # irá acumular as soma dos  
#printando o resultado acima  
print(r)
```

494

## CLASSES

1. Recrie a classe Carro e adicione um novo atributo e um novo método (função).

```
class Carro():  
    def __init__(self, litros, velocidade=0,buzina='BIBI!!'):  
        self.litros = litros  
        self.velocidade = velocidade  
        self.buzina= buzina  
  
    def aciona_buzina(self,litros=20):  
        print(self.buzina)  
  
    def gasolina(self,litros=20):  
        print("Tem ",self.litros," de gasolina")  
  
    def acelera(self,velocidade=0):  
        velocidade=self.velocidade  
        if self.velocidade < 120:  
            self.litros = self.litros-2  
            print("Velocidade: ",self.velocidade,"\nQuantidade de litros de gasolina: ",self.litros  
        else:  
            self.litros = self.litros-5  
            print("Velocidade: ",self.velocidade,"\nQuantidade de litros de gasolina: ",self.litros
```



Segue abaixo gabarito dos exercícios dados em aula:

2. Crie uma sub classe a partir da classe criada no exercício 1.

```
class gol_bolinha(Carro):
    def __init__(self): # o construtor da subclasse chama o construtor da superclasse
                        # com parametros desejados
        Carro.__init__(self,litros=9,velocidade = 50)
```

3. Sobreescreva um dos operadores dentro de uma nova classe.

```
class gol_bolinha(Carro):
    def __init__(self): # o construtor da subclasse chama o construtor da superclasse
                        # com parametros desejados
        Carro.__init__(self,litros=9,velocidade = 50)

    def __lt__(self,p): # operador '<'
        return(self.idade < p.idade)
```

4. Crie uma classe chamada Calculadora com os atributos Digito 1 e Digito2, crie os métodos de

```
class Calculadora():
    def __init__(self, x, y):
        self.x = x
        self.y = y
```

```
    def subtrai (self,x,y):
        return x - y
```

```
    def add (self,x,y):
        return x+y
```

```
soma = Calculadora()
soma.add(10,20)
```

30

5. Crie uma subclasse de Calculadora e adicione os métodos de divisão, multiplicação e exponencial.

```
class nova_calculadora(Calculadora):
    def __init__(self):
        Calculadora.__init__(self,x=0,y=0)
```

```
    def mult (self,x,y):
        return x*y
```

```
    def divide (self, x,y):
        return x/y
```

```
mul = nova_calculadora()
mul.mult(10,10)
```

100



Segue abaixo gabarito dos exercícios dados em aula:

## OPERAÇÕES COM LISTAS

1. Construa uma lista de inteiros e ordene elas do maior para o menor.

```
lista=[10,2,3,1,7,2,3,5,4,12,333,12]
lista.sort(reverse=True)
print(lista)
```

```
[333, 12, 12, 10, 7, 5, 4, 3, 3, 2, 2, 1]
```

2. Adicione os valores 10,7,33,22 com uma única linha de código e deixe-os ordenados também em uma segunda linha de código.

```
lista.extend([10,7,33,22])
lista.sort(reverse=True)
print(lista)
```

```
[333, 33, 22, 12, 12, 10, 10, 7, 7, 5, 4, 3, 3, 2, 2, 1]
```

3. Gere uma lista de nomes e ordene ela pelo tamanho de cada nome.

```
nomes=['Peter','Daniel','Breno','Leticia','Dani']
nomes.sort(key=len)
print(nomes)
```

```
['Dani', 'Peter', 'Breno', 'Daniel', 'Leticia']
```

4. Crie uma lista que vá de 1 até 100.

```
[a for a in range(1,101)]
42,
43,
44,
45,
46,
47,
48,
49,
50,
51,
52,
53,
54,
55,
56,
57,
58,
```

Segue abaixo gabarito dos exercícios dados em aula:

```
59,  
60,  
61,  
62,  
63,  
64,  
65,  
66,  
67,  
68,  
69,  
70,  
71,  
72,  
73,  
74,  
75,  
76,  
77,  
78,  
79,  
80,  
81,  
82,  
83,  
84,  
85,  
86,  
87,  
88,  
89,  
90,  
91,  
92,  
93,  
94,  
95,  
96,  
97,  
98,  
99,  
100]
```

5. Em uma nova lista adicione somente os valores ímpares de 1 a 100.

```
[a for a in range(1,101,2)]
```

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



Segue abaixo gabarito dos exercícios dados em aula:

```
31,  
33,  
35,  
37,  
39,  
41,  
43,  
45,  
47,  
49,  
51,  
53,  
55,  
57,  
59,  
61,  
63,  
65,  
67,  
69,  
71,  
73,  
75,  
  
77,  
79,  
81,  
83,  
85,  
87,  
89,  
91,  
93,  
95,  
97,  
99]
```

## OPERAÇÕES STRINGS

1. Crie uma parágrafo contando uma experiência de vida e armazene isso em uma string. Depois retorne a quantidade de palavras desse parágrafo.

```
texto = 'Meu nome é Fulano e estou me dedicando muito para me tornar um data expert. \nEstudo  
palavras = texto.split()  
print(len(palavras))
```

Segue abaixo gabarito dos exercícios dados em aula:

2. Conte a quantidade de nomes próprios (ou palavras com letras maiúsculas) e de números que aparecem na história.

```
titles = [a for a in texto.split() if a.istitle()]
numeros = [a for a in texto.split() if a.isnumeric()]
print('Quantidade de nomes ou palavras com letra maiúscula:', len(titles), '\nQuantidade de Num

Quantidade de nomes ou palavras com letra maiúscula: 4
Quantidade de Numeros: 1
```

3. Conte a quantidade de vogais que aparecem na história.

```
vogais = [a for a in texto if a in ['a', 'e', 'i', 'o', 'u']]
print('Quantidade de vogais:', len(vogais))

Quantidade de vogais: 48
```

4. Conte a quantidade de linhas que aparece na string.

```
linhas = [a for a in texto.splitlines()]
print('Quantidade de linhas:', len(linhas))

Quantidade de linhas: 2
```

5. Utilize pelo menos mais 3 métodos próprios pra string que não apareceu em aula mas estão na referência.

```
print(texto.encode())

b'Meu nome \xc3\xa9 Fulano e estou me dedicando muito para me tornar um data expert. \n

print(texto.count('Python'))

1

py = [a for a in palavras if a.startswith('Py')]
print(py)

['Python!']
```



Segue abaixo gabarito dos exercícios dados em aula:

## OPERAÇÕES TUPLAS

1. Crie uma tupla e ordene ela.

```
tupla =sorted((2,3,1,3,4,1))  
print(tupla)
```

```
[1, 1, 2, 3, 3, 4]
```

2. Remova todos os valores duplicados da tupla.

```
set(tupla)
```

```
{1, 2, 3, 4}
```

3. Crie uma função que busque um determinado valor em uma tupla e retorne a quantidade de vezes que esse valor aparece.

```
def conta (x,y):  
    return x.count(y)  
conta(tupla,3)
```

```
2
```

4. Crie dois sets e atribua no primeiro a diferença simétrica entre os dois.

```
set1 ={1,2,3,4,5,3,5,6,7}  
set2={2,5,20,4,95,2,9,77,33,12}  
set1.symmetric_difference_update(set2)  
print(set1)
```

```
{1, 3, 6, 7, 33, 9, 12, 77, 20, 95}
```

5. Faça a união entre os dois sets e salve o resultado em um terceiro set.

```
set3={100,222,333,444,555,777,333,999,333}  
set3.update(set1)  
print(set3)
```

```
{1, 33, 3, 100, 6, 999, 7, 777, 9, 555, 12, 333, 77, 20, 444, 222, 95}
```

Segue abaixo gabarito dos exercícios dados em aula:

## OPERAÇÕES DICIONÁRIOS

1. Crie um dicionário com nome, sobrenome e apelido de 3 usuários.

```
dic1 = {'user1':{'nome':'Peter','sobrenome':'Almeida','apelido':'Pet'},
        'user2':{'nome':'Fulano','sobrenome':'Silva','apelido':'Fula'},
        'user3':{'nome':'Daniel','sobrenome':'Soria','apelido':'Raj'}}
dic1
```

```
{'user1': {'apelido': 'Pet', 'nome': 'Peter', 'sobrenome': 'Almeida'},
 'user2': {'apelido': 'Fula', 'nome': 'Fulano', 'sobrenome': 'Silva'},
 'user3': {'apelido': 'Raj', 'nome': 'Daniel', 'sobrenome': 'Soria'}}
```

2. Procure o atributo idade para o usuário 1 e caso não tenha adicione o valor 30 nesse campo.

```
for a,b in dic1.items():
    b.setdefault('idade',30)
dic1

{'user1': {'apelido': 'Pet',
            'idade': 30,
            'nome': 'Peter',
            'sobrenome': 'Almeida'},
 'user2': {'apelido': 'Fula',
            'idade': 30,
            'nome': 'Fulano',
            'sobrenome': 'Silva'},
 'user3': {'apelido': 'Raj',
            'idade': 30,
            'nome': 'Daniel',
            'sobrenome': 'Soria'}}
```

3. Conte a quantidade de nomes e sobrenomes contidos no dicionário.

```
"""
s=0
for a,b in dic1.items():
    nome =b.get('nome',0)
    snome =b.get('sobrenome',0)
    if nome != 0:
        n+=1
    if snome!=0:
        s+=1
print('total nomes:',n,'\ntotal sobrenomes:',s,'\ntotal de nomes e sobrenomes:',s+n)

total nomes: 3
total sobrenomes: 3
total de nomes e sobrenomes: 6
```



Segue abaixo gabarito dos exercícios dados em aula:

4. Verifique se há dois nomes repetidos.

```
nomes=[]
for a,b in dic1.items():
    nome =b.get('nome',0)
    nomes.append(nome)

for a,b in dic1.items():
    nome =b.get('nome',0)
    print(nomes.count(nome),nome)

1 Peter
1 Fulano
1 Daniel
```

## CORRIGINDO ERROS

### Erro de sintaxe

```
if 30 >20:
    print("maior")
    maior
```

```
lista = [1,2,3,4] #as vezes o erro da sintaxe faz o compilador entender outra coisa, como uma
lista

[1, 2, 3, 4]
```

### Divisão por zero

```
10/2

5.0
```

### Erros de tipos diferentes não compatíveis em operações

```
"2"+"2"

'22'
```



Segue abaixo gabarito dos exercícios dados em aula:

## Uso de função de uma biblioteca não importada ainda

```
from functools import reduce  
reduce(lambda x,y:x+y, range(10))
```

45

## Alterar lista original

```
original=[1,2,3,4,5]  
copia = original.copy()  
copia.append(7)  
print(original, copia)
```

[1, 2, 3, 4, 5] [1, 2, 3, 4, 5, 7]