

Primeiros Exemplos

In [3]:

```
print('Primeiro Programa')
```

Primeiro Programa

In [4]:

```
print(1+2+3)  
print(4+5+6)
```

6
15

Tipos Básicos

In [45]:

```
print(True)  
print(False)  
print(1.2+1)  
print('Aqui eu falo minha lingua')  
print("Tb funciona")  
print('Você é ' + 3 * 'muito ' + 'legal!')  
#print(3 + '3') -> Devido a ambiguidade  
print([1,2,3])  
print({'Nome': 'Pedro', 'Idade': 20})  
print(None)
```

True
False
2.2
Aqui eu falo minha lingua
Tb funciona
Você é muito muito muito legal!
[1, 2, 3]
{'Nome': 'Pedro', 'Idade': 20}
None

Variáveis

In [25]:

```
a = 10
b = 5.2

print(a + b)

a = 'Agora sou uma String'
print(a)

#print(a + b) ->Não imprime devido a ambiguidade gerada.
```

15.2
Agora sou uma String

Comentários

In [40]:

```
# minhas variáveis
salario = 3450.45
despesas = 2456.2

'''
A ideia é calcular o
quanto vai sobra no final do mês!
'''

print(salario - despesas)#Calculo da Renda

print('Fim')
```

994.25
Fim

Operadores Aritméticos

In [56]:

```
print(2 + 3)
print(4 - 7)
print(2 * 5.3)
print(9.4 / 3)

print(9.4 // 3)
print(2 ** 8)
print(10 % 3)

a = 12
b = a
print(a + b)
```

```
5
-3
10.6
3.1333333333333333
3.0
256
1
24
```

In [63]:

```
# minhas variáveis
salario = 3450.45
despesas = 2456.2

Percetual_de_despesas = (despesas*100)//salario
print(Percetual_de_despesas)
```

```
71.0
```

Operadores Relacionais

In [67]:

```
3 > 4
4 >= 3
1 < 2
3 <= 1
3 != 2
3 == 3
2 == '2'
```

Out[67]:

```
False
```

Operadores de Atribuição

In [77]:

```
a = 3
a = a + 7
print(a)

a += 5
print(a)

a -= 3
print (a)

a *= 2
print(a)

a /= 4
print(a)

a %= 4
print(a)

a **= 8
print(a)

a //= 256
print(a)
```

```
10
15
12
24
6.0
2.0
256.0
1.0
```

Operadores Lógicos

In [5]:

```
True or False
7 != 3 and 2 > 3

# Tabela Verdade do AND
True and True # V
True and False # F
False and True # F
False and False # F

# Tabela Verdade do or
True or True # V
True or False # V
False or True # V
False or False # F

# Tabela Verdade do Xor
True != True # F
True != False # V
False != True # V
False != False # F

# Operador de Negação (Unário)

not True
not False

not 0
not 1
not not -1
not not True

#Cuidado!!

True & False
False | True
True ^ False

# AND Bit a Bit
# 3 = 11
# 2 = 10
#Resultado = 10

3 & 2

# Or Bit a Bit
# 3 = 11
# 2 = 10
#Resultado = 11

3 | 2

# Xor Bit a Bit
# 3 = 11
# 2 = 10
#Resultado = 01

3 ^ 2
```

Out[5]:

1

Operadores Unários

In [39]:

```
a = 3
#a++ não funciona em python
#a += 1
#a
#a-- não funciona em python
++a
-a
+a
print(a)

not 0
not 1
not -2
not False
not not True
```

3

Out[39]:

True

Operadores Ternários

In [21]:

```
esta_chuvendo = False
print("Hoje estou com as roupas " + ('secas! ', 'Molhadas! ')[esta_chuvendo])

print('Hoje estou com as roupas ' + ('Molhadas.' if esta_chuvendo else 'secas! '))
```

Hoje estou com as roupas secas!
Hoje estou com as roupas secas!

Mais Operadores

In [30]:

```
# Operador de Membro

lista = [1,2,3,'Ana','Carla']

2 in lista
'Ana' not in lista

x = 3
y = x
z = 3

x is y
y is z
x is z

x is not z

lista_a = [1,2,3]
lista_b = lista_a
lista_c = [1,2,3]

lista_a is lista_b
lista_b is lista_c #False
lista_a is not lista_c
```

Out[30]:

True

Conversão de Tipos

In [41]:

```
2 + 3
'2' + '3'
# 2 + '3'
# print(2 + '3')

a = 2
b = '3'

print(type(a))
print(type(b))

print(a + int(b))
print(str(a) + b)
```

```
<class 'int'>
<class 'str'>
5
23
```

Coerção Automática

In [54]:

```
10 / 2
type(10/2)
10 / 3
10 // 3
type(10 // 3)
10 // 3.3
type(10 // 3.3)
10 / 2.5
2 + True
2 + False
type(1 + 2)
type(1 + 2.5)
```

Out[54]:

float

Tipos Numéricos

In [9]:

```
dir(int)
dir(float)

a = 5
b = 2.5

a / b
a + b
a * b

type(a)
type(b)

b.is_integer()
5.0.is_integer()
```

Out[9]:

True

In [20]:

```
#1.1 + 2.2

from decimal import Decimal, getcontext

Decimal(1) / Decimal(7)

getcontext().prec = 4
Decimal(1) / Decimal(7)
Decimal.max(Decimal(1),Decimal(7))

dir(Decimal)

1.1 + 2.2

getcontext().prec = 10
Decimal(1.1) + Decimal(2.2)
```

Out[20]:

```
Decimal('3.3000000000')
```

Tipo String

In [3]:

```
dir(str)

nome = "Luiz Junior"

nome[0]

"Dias D'Avila" == 'Dias D\'Avila'

texto = 'Texto entre apostófos pode ter "aspas"'

doc = """Texto com multiplas
... linhas"""

doc2 = '''Texto com multiplas
... linhas'''

doc
doc2
```

Out[3]:

```
'Texto com multiplas\n... linhas'
```

In [22]:

```
nome = 'Ana Paula'

nome[0]
nome[6]
nome[-3]
nome[4:]
nome[-5:]
nome[:3]
nome[2:5]

numeros = '1234567890'
numeros
numeros[::]
numeros[::2]
numeros[1::2]
numeros[::-1]
numeros[::-2]

nome[::-1]
```

Out[22]:

'aluaP anA'

In [34]:

```
frase = 'Python é uma linguagem excelente'
'py' in frase
'ing' in frase
len(frase)
frase.lower()
frase
frase.upper()
frase

frase = frase.upper()
frase

frase.split()
frase.split('E')
```

Out[34]:

['PYTHON É UMA LINGUAG', 'M ', 'XC', 'L', 'NT', '']

In [45]:

```
a = '123'
b = ' de Oliveira 4'

a + b

#dir(str)
```

Out[45]:

'123 de Oliveira 4'

Lista

In [57]:

```
lista = []
type(lista)
dir(lista)
len(lista)
lista.append(1)
lista.append(5)
lista
len(lista)

nova_lista = [1,5,'Ana','Bia']
nova_lista
nova_lista.remove(5)
nova_lista
nova_lista.reverse()
nova_lista
```

Out[57]:

```
['Bia', 'Ana', 1]
```

In [73]:

```
Lista = [1,5,'Rebeca','Guilherme',3.1415]
Lista.index('Guilherme')
#Lista.index(42)
Lista[2]
1 in Lista
'Rebeca' in Lista
'Pedro' not in Lista
Lista[0]
Lista[4]
#Lista[5]
Lista[-1]
Lista[-5]

Lista
```

Out[73]:

```
1
```

In [11]:

```
lista = ['Ana', 'Lia', 'Rui', 'Paulo', 'Dani']
lista[1:3]
lista[1:-1]
lista[1:]
lista[:-1]
lista[:]
lista[:2]
lista[::-1]
del lista[2]
lista
del lista[1:]
lista
```

Out[11]:

```
['Ana']
```

Tupla

In [23]:

```
tupla = ()
dir(tupla)
tupla = ('um',)
type(tupla)
tupla[0]

cores = ('verde', 'amarelo', 'azul', 'branco')
cores[0]
cores[-1]
cores[1:]

cores.index('amarelo')# numero do index que se encontra o amarelo
cores.count('azul')# numeros de elementos repetidos
len(cores)
```

Out[23]:

```
4
```

Dicionário

In [20]:

```
peessoa = {'nome': 'Ana', 'idade': 38, 'cursos': ['Inglês', 'Português ']}
type(peessoa)

dir(dict)

len(peessoa)

peessoa['nome']
peessoa['idade']
peessoa['cursos'][0]

peessoa.keys()
peessoa.values()
peessoa.items()
peessoa.get('idade')
peessoa.get('tag', 'Dado não existente')
```

Out[20]:

'Dado não existente'

In [35]:

```
peessoa = {'nome': 'Alberto', 'idade': 43, 'cursos': ['React', 'python ']}
peessoa['idade'] = 44
peessoa['cursos'].append('Angular')
peessoa

peessoa.pop('idade')
peessoa

peessoa.update({'idade': 40, 'Sexo': 'M'})
peessoa

del peessoa['cursos']
peessoa

peessoa.clear()
peessoa
```

Out[35]:

{}

Set / Conjunto

In [57]:

```

a = {1,2,3}
type(a)

a = set('T4KUMA')
print(a)

print('4' in a , 3 not in a)
{1,2,3} == {3,2,1,3}

#operações

c1 = {1,2}
c2 = {2,3}
c1.union(c2)

dir(set)
c1.intersection(c2)
c1.update(c2)
c1

c2 <= c1 #verificar se e C2 é subconjunto de C1
c1 >= c2 #verificar se C1 é superconjunto de C2

{1,2,3} - {2}
c1 - c2
c1 -= {2}
c1

```

```

{'A', 'K', '4', 'M', 'T', 'U'}
True True

```

Out[57]:

```

{1, 3}

```

Interpolação

In [68]:

```

from string import Template

nome, idade = 'Ana' , 30

print('Nome: %s Idade: %d' %(nome,idade)) #mais antiga
print('Nome: {0} Idade: {1}'.format(nome, idade)) #python < 3.6
print(f'Nome: {nome} Idade: {idade}') # python >= 3.6

s = Template('Nome: $nome Idade: $idade')
print(s.substitute(nome = nome , idade = idade))

```

```

Nome: Ana Idade: 30
Nome: Ana Idade: 30
Nome: Ana Idade: 30
Nome: Ana Idade: 30

```