Básico

01- Comentários

Comentários em Python são feitos a partir do caractere "#". Exemplo:

```
#Isso é um comentário
```

Para comentar múltiplas linhas, use '''. Exemplo:

```
'''
linha1
linha2
linha3
'''
```

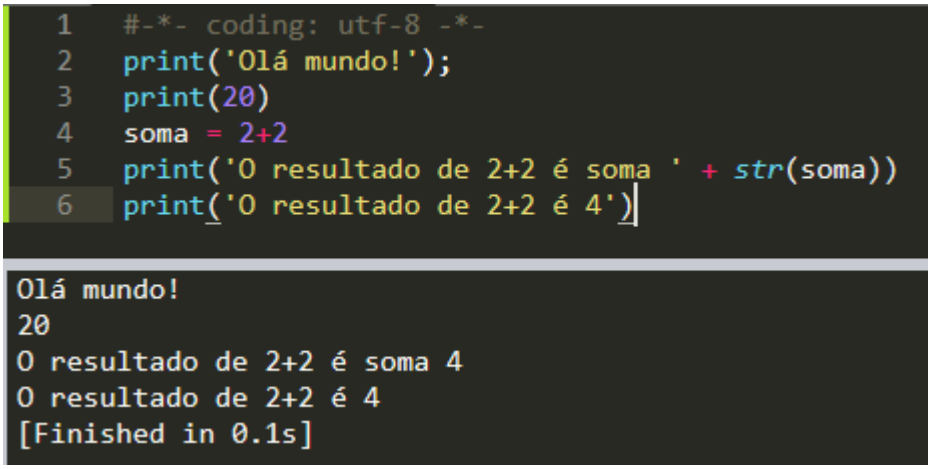
Por padrão, Python não compreende acentos e caracteres especiais, logo é necessário declarar que queremos usar esta formatação com o seguinte comando (com #):

```
#-*- coding: utf-8 -*-
```

02- Escrever na tela

```
print()
```

No caso de texto, colocar entre aspas simples. Caso seja uma variável, números ou operações aritméticas, não é necessário aspas. Exemplo:



03- Operadores

Matemáticos:

Operador	Operação
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
**	Exponenciação
%	Módulo

Relacionais:

Operador	Operação
==	Igual
!=	Diferente
>	Maior
<	Menor
>=	Maior ou igual
<=	Divisão

Lógicos:

Operador	Operação
AND	Duas condições sejam verdadeiras
OR	Pelo menos uma condição seja verdadeira
NOT	Inverte o valor

Relacionais:

if, else, elif:

```
1   #-*- coding: utf-8 -*-
2   x = 2
3   y = 1
4
5   if (x > y):
6       print ('x é maior que y')
7   elif (x == y):
8       print ('x é igual a y')
9   else:
10      print ('x e y são diferentes')
```

```
x é maior que y
[Finished in 0.1s]
```

04- Laços de repetição:

While:

```
#-*- coding: utf-8 -*-
x = 0
while x < 5:
    print(x)
    x += 1
```

```
0
1
2
3
4
[Finished in 0.1s]
```

For:

```
#-*- coding: utf-8 -*-
lista = [1,2,3,4]
for i in lista:
    print(i)
```

```
1
2
3
4
[Finished in 0.1s]
```

05- Ferramentas de strings

Tamanho da string (**len**):

```
#-*- coding: utf-8 -*-
name = "Rafael"
print(len(name))
```

```
6
[Finished in 0.1s]
```

Posição de um caractere na string:

```
#-*- coding: utf-8 -*-
name = "Rafael"
print(name[0])
```

```
R
[Finished in 0.1s]
```

** Caso **name** fosse uma lista, "[0]" retornaria o primeiro elemento dessa lista.*

Parte de uma string:

```
1  #-*- coding: utf-8 -*-
2  name = "Rafael"
3  print(name[0:4])
4
```

```
Rafa
[Finished in 0.1s]
```

** O primeiro delimitador começa com 0 e segundo delimitador, com 1 (uma vez que "0" já está sendo utilizado pelo primeiro delimitador*

Tudo maiúsculo, tudo minúsculo (**lower, upper**):

```
1  #-*- coding: utf-8 -*-
2  name = "Rafael"
3  print(name.lower())
4  print(name.upper())
5
```

```
rafael
RAFAEL
[Finished in 0.1s]
```

Remover espaços e linhas em branco (**strip**):

```
1  #-*- coding: utf-8 -*-
2  name = "Rafael " + "\n"
3  print(name)
4  print(name.strip())
5
```

```
Rafael

Rafael
[Finished in 0.1s]
```

Limitador (**split**):

```
1  #-*- coding: utf-8 -*-
2  name = input('Qual o seu nome: ')
3  name_to_list = name.split(" ")
4  print(name_to_list)
5  name_len = len(name_to_list)
6  print(name_to_list[0] + " " + name_to_list[name_len - 1])
```

```
C:\Windows\system32\cmd.exe

C:\Users\F12096605717>desktop\demo.py
Qual o seu nome: Rafael José Santana da Costa
['Rafael', 'José', 'Santana', 'da', 'Costa']
Rafael Costa

C:\Users\F12096605717>
```

Busca (**find**):

```python
#-*- coding: utf-8 -*-
name = 'Rafael José Santana da Costa'
busca = name.find('José')
busca_erro = name.find('Andressa')
print(busca)
print(busca_erro)
print(name[busca:len(name)])
```

```
7
-1
José Santana da Costa
[Finished in 0.1s]
```

Substituir (**replace**):

```python
#-*- coding: utf-8 -*-
name = 'Rafael José Santana da Costa'
new_name = name.replace("Santana","Sant'Ana")
print(new_name)
```

```
Rafael José Sant'Ana da Costa
[Finished in 0.1s]
```

05.1- Expressões regulares

Métodos **search, findall, search**

```python
import re
string = 'Este teste é um teste de expressões regulares'
print(re.search(r'teste', string))
print(re.findall(r'teste',string))
print(re.sub(r'teste','',string, count=1))
print()

regexp = re.compile(r'teste')
print(regexp.search(string))
print(regexp.findall(string))
print(regexp.sub('DEF', string))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\Rafael> & python d:/PerfilRafael/Desktop/current.py
<re.Match object; span=(5, 10), match='teste'>
['teste', 'teste']
Este  é um teste de expressões regulares

<re.Match object; span=(5, 10), match='teste'>
['teste', 'teste']
Este DEF é um DEF de expressões regulares
```

Meta caracteres:

```
[]  A set of characters "[a-m]"
\   Signals a special sequence (can also be used to escape special characters)  "\d"
.   Any character (except newline character)    "he..o"
^   Starts with "^hello"
$   Ends with   "world$"
*   Zero or more occurrences    "aix*"
+   One or more occurrences "aix+"
{}  Exactly the specified number of occurrences "al{2}"
|   Either or   "falls|stays"
()  Capture and group
```

```
texto = '''
João trouxe   flores para sua amada namorada em 10 de janeiro de 1970, Maria era o nome dela.


Foi um ano excelente na vida de joão. Teve 5 filhos, todos adultos atualmente. ma-
ria, hoje sua esposa, ainda faz aquele café com pão de queijo nas tardes de domingo. Tam-
bém né! Sendo a boa mineira que é, nunca esquece seu famoso pão de queijo.

Não canso de ouvir a Maria:
"Joooooõaooooooo, o café tá prontinho aqui. Veeemm"!
'''
```

```
25  print(re.findall(r'João|Maria',texto))
26  print(re.findall(r'Joã.|Ma..a',texto))
27  print(re.findall(r'[Jj]oã.|[Mm]aria',texto))
28  print(re.findall(r'[a-zA-Z]oã.|[Mm]aria',texto))
29  print(re.findall(r'JOão|maRIA',texto,flags=re.IGNORECASE))
```

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL

PS C:\Users\Rafael> & python d:/PerfilRafael/Desktop/Misc/Scripts/PYTHON/Códigos/script.py
['João', 'Maria', 'Maria']
['João', 'Maria', 'Maria']
['João', 'Maria', 'joão', 'maria', 'Maria']
['João', 'Maria', 'joão', 'maria', 'Maria', 'ooão']
['João', 'Maria', 'joão', 'maria', 'Maria']
PS C:\Users\Rafael>
```

06- Ferramentas de listas:

Ordenar (**sorted**):

```
22  lista1 = ["Rafael", "José", "Santana", "da", "Costa"]
23  print(sorted(lista1))
24  lista2 = [3,4,1,5,2]
25  print(sorted(lista2))
```

```
['Costa', 'José', 'Rafael', 'Santana', 'da']
[1, 2, 3, 4, 5]
[Finished in 0.1s]
```

Adicionar item a uma lista (**append**):

```
1  #-*- coding: utf-8 -*-
2  lista1 = ["Banana","Uva","Maçã"]
3  lista1.append("Abacate")
4  lista2 = [1,2,3]
5  lista2.append(4)
6  print(lista1)
7  print(lista2)
```

```
['Banana', 'Uva', 'Maçã', 'Abacate']
[1, 2, 3, 4]
[Finished in 0.1s]
```

Verificar se um item está na lista:

```
1  #-*- coding: utf-8 -*-
2  lista1 = ["Banana","Uva","Maçã"]
3  if ("Banana" in lista1):
4      print('Lista1 contém Banana')
```

```
Lista1 contém Banana
[Finished in 0.1s]
```

Remover item de uma lista (**del**):

```
1  #-*- coding: utf-8 -*-
2  lista1 = ["Banana","Uva","Maçã"]
3  del lista1[1]
4  print(lista1)
```

```
['Banana', 'Maçã']
[Finished in 0.1s]
```

Apagar todo conteúdo da lista:

```
1  #-*- coding: utf-8 -*-
2  lista1 = ["Banana","Uva","Maçã"]
3  del lista1[:]
4  print(lista1)
```

```
[]
[Finished in 0.1s]
```

Ordenando listas (**sort**):

```
#-*- coding: utf-8 -*-
lista1 = ["3","2","4","1"]
lista1.sort()
print(lista1)
lista1.sort(reverse=True)
print(lista1)
```

```
['1', '2', '3', '4']
['4', '3', '2', '1']
[Finished in 0.1s]
```

Reverter lista (**reverse**):

```
#-*- coding: utf-8 -*-
lista1 = ["3","2","4","1"]
lista1.reverse()
print(lista1)
```

```
['1', '4', '2', '3']
[Finished in 0.1s]
```

Intermediário

07- Dicionários:

```
#-*- coding: utf-8 -*-
dicionario = {"Fruta":"Goiaba","Celular":"Motorola","Carro":"BMW"}
for i in dicionario.items():
    print(i)
print()
for i in dicionario.values():
    print(i)
print()
for i in dicionario.keys():
    print(i)

```

```
('Fruta', 'Goiaba')
('Celular', 'Motorola')
('Carro', 'BMW')

Goiaba
Motorola
BMW

Fruta
Celular
Carro
[Finished in 0.1s]
```

Removendo itens, realizando buscas:

```python
dic = {1:"aaa", 2:"bbb", 3:"ccc"}
del(dic[1])
print(dic)
print()

busca1 = 1 in dic
print(busca1)
busca2 = 2 in dic
print(busca2)
```

```
{2: 'bbb', 3: 'ccc'}

False
True
[Finished in 0.1s]
```

Copiando e referenciando:

```python
a = {10:"Máximo", 5:"Meio", 1:"Mínimo"}
b = a
c = a.copy()
del(a[5])
print(a)
print(b)
print(c)
```

```
{10: 'Máximo', 1: 'Mínimo'}
{10: 'Máximo', 1: 'Mínimo'}
{10: 'Máximo', 5: 'Meio', 1: 'Mínimo'}
[Finished in 0.1s]
```

Mesclando:

```python
a = {"aaa":10, "bbb":20, "ccc":30}
b = {"ddd":40, "eee":50, "ddd":60}
a.update(b)
print(a)
```

```
{'aaa': 10, 'bbb': 20, 'ccc': 30, 'ddd': 60, 'eee': 50}
[Finished in 0.1s]
```

Converter lista para dicionário:

```python
l1 = ['aaa','bbb','ccc']
l2 = ['1','2','3']
dic = list(zip(l1, l2))
print(dic)
```

```
[('aaa', '1'), ('bbb', '2'), ('ccc', '3')]
[Finished in 0.1s]
```

08- Arquivos

Modo	Função
r	somente leitura
w	escrita (caso o arquivo já exista, ele será apagado e um novo arquivo vazio será criado)
a	leitura e escrita (adiciona o novo conteúdo ao fim do arquivo)
r+	leitura e escrita
w+	escrita (o modo w+, assim como o w, também apaga o conteúdo anterior do arquivo)
a+	leitura e escrita (abre o arquivo para atualização)

Lendo um arquivo:

```
1  #-*- coding: utf-8 -*-
2  arquivo = open("ola.txt")
3  linhas = arquivo.readlines()
4  print(linhas)
5
6  print()
7  for linha in linhas:
8      print(linha.strip())
```

```
['Olá, Mundo!\n', 'Meu nome é Rafael!']

Olá, Mundo!
Meu nome é Rafael!
[Finished in 0.1s]
```

Criar e atualizar um arquivo:

```
1  #-*- coding: utf-8 -*-
2  arquivo = open("ola2.txt", "w")
3  arquivo.write("Arquivo criado pelo Python")
4  arquivo.close()
5  arquivo = open("ola2.txt", "a")
6  arquivo.write("\nNova linha")
```

09- Funções

```python
#-*- coding: utf-8 -*-
def soma(x,y):
    return x+y

def multiplicacao(x,y):
    return x*y

s = soma(2,3)
print(s)

m = multiplicacao(3,4)
print(m)

print(soma(m,s))
```

```
5
12
17
[Finished in 0.1s]
```

Funções submetem variáveis a uma série de comandos

10- Caracteres aleatórios:

```python
#-*- coding: utf-8 -*-
import random
import string

numero = random.randint(0,10)
print(numero)
lista = [1,5,10,8,44]
numero = random.choice(lista)
print(numero)
letra_min = random.choice(string.ascii_lowercase)
print(letra_min)
letra_mas = random.choice(string.ascii_uppercase)
print(letra_mas)
```

```
4
5
b
H
[Finished in 0.1s]
```

11- Tratamento de exceções:

```python
result = []
while len(result) == 0:
    num1 = int(input('Primeiro número: '))
    num2 = int(input('Segundo número: '))
    try:
        result = str(num1/num2)
    except:
        print('Não foi possível realizar a operação')
    else:
        print('Operação realizada com sucesso!')
```

```
Primeiro número: 10
Segundo número: 0
Não foi possível realizar a operação
Primeiro número: 10
Segundo número: 3
Operação realizada com sucesso!


...Program finished with exit code 0
Press ENTER to exit console.
```