

# FOLHA DE CONSULTA – Python

**\*\* RESUMO PARA UTILIZAÇÃO NA AVALIAÇÃO \*\***

## 1. Operadores Aritméticos:

- **+** (adição), **-** (subtração)
- **\*** (multiplicação), **/** (divisão real)
- **//** (quociente da divisão)
- **%** (resto da divisão)

```
x1 = 4 * 3      #(resulta em 12)
x2 = 5 // 2     #(resulta em 2)
x3 = 5 % 2      #(resulta em 1)
x4 = 5 / 2      #(resulta em 2.5)
```

## 2. Módulo math: Disponibiliza funções matemáticas (precisa ser importado com import)

```
import math

math.inf          # valor float que representa +∞
-math.inf         # valor float que representa -∞
math.sqrt(n)      # retorna a raiz quadrada de n
math.log2(n)       # retorna o logaritmo de n na base 2
math.log10(n)      # retorna o logaritmo de n na base 10
math.log(n, b)     # retorna o logaritmo de n na base b
math.exp(n)        # retorna en
```

## 3. Operadores Relacionais:

**==** (igual), **!=** (diferente), **<**, **<=**, **>**, **>=**

```
x1 = 0; x2 = 1

x1 != x2          #(resulta em True)
x1 >= x2          #(resulta em False)
```

## 4. Operadores Lógicos:

**and**, **or**, **not**

```
if (a > b) and (a > c) and (a > d):
    print(a, "é maior que", b, c, d)

if nota < 0 or nota > 10:
    print("essa nota é inválida")
```

## 5. Entrada de Dados

**input()**: lê um valor do teclado como uma string e armazena numa variável. As funções **int()** e **float()** devem ser usadas para converter o valor para int ou float, respectivamente.

```
nome = input("Digite seu nome: ")
idade = int(input("Digite sua idade: "))
salario = float(input("Digite seu salário: "))
```

## 6. Saída de Dados

**print():** imprime uma mensagem, valor de variável ou resultado de operação aritmética na tela.

```
x = 1; y = 2.666666

print('o valor de x é:', x)      # sem usar fstring
print(f'o valor de x é: {x}')    # usando fstring
print(f'o valor de y é: {y: .2f}') # usando fstring imprime y com 2 casas decimais
```

**7a. if sozinho:** se teste do if for verdadeiro, seus comandos subordinados serão executados.

```
if (a > b):
    print(a, 'é maior que', b)
```

**7b. if com else:** se teste do if for verdadeiro, seus comandos subordinados serão executados. Caso contrário, os comandos subordinados ao else serão executados

```
if (a > b):
    print(a, 'é maior que', b)
else:
    print(b, 'é maior que', a)
```

**7c. if com elif:** possibilita a avaliação de diversas alternativas.

```
if (nota >= 9):
    conceito = "A"
elif (nota >= 7):
    conceito = "B"
elif (nota >= 5):
    conceito = "C"
else:
    conceito = "D"
```

**8. Repetição (laço ou loop):** um bloco de comandos é executado repetidas vezes.

**while:** repetição com teste no início.

```
i = 1;
while (i <= 3):
    print('ENCE', i)
    i += 1
```

*saída:*

ENCE 1

ENCE 2

ENCE 3

**for-range():** iteração através dos números de uma sequência gerada pelo range()

```
for i in range(3):  
    print('ENCE', i)
```

*saída:*  
ENCE 0  
ENCE 1  
ENCE 2

```
for i in range(1, 4):  
    print('ENCE', i)
```

*saída:*  
ENCE 1  
ENCE 2  
ENCE 3

## 9. Lista: principais operações

```
lst = [60, 100, 25, 15]  
lst[0]                # 60 (indexação)  
100 in lst            # True (verifica se valor está na lista)  
999 in lst            # False  
len(lst)              # 4 - tamanho de lst (número de elementos)  
sum(lst)              # 200 - soma os elementos da lista  
max(lst)              # 100 - maior elemento da lista  
min(lst)              # 15 - menor elemento da lista  
  
numeros = [5, 10]  
  
numeros.append(20)     # insere 20 no final: [5, 10, 20]  
numeros.append(10)     # insere 10 no final: [5, 10, 20, 10]  
numeros.insert(2,15)   # insere 15 na posição 2: [5, 10, 15, 20, 10]  
numeros.insert(0,10)   # insere 10 na posição 0: [10, 5, 10, 15, 20, 10]  
  
numeros.pop()          # remove o último elemento: [10, 5, 10, 15, 20]  
numeros.pop(3)         # remove o quarto elemento: [10, 5, 10, 20]  
numeros.remove(10)     # remove o primeiro 10: [5, 10, 20]  
  
numeros.extend([40, 50]) # estende a lista: [5, 10, 20, 40, 50]  
  
numeros.clear()        # esvazia a lista: [ ]
```