



## PYTHON PARA CIÊNCIA DE DADOS

### ESTRUTURA CONDICIONAL



**Conceito:** serve para testar variáveis e fazer comparações, criando fluxos de execução independentes dentro do programa.

SINTAXE GERAL			
Estrutura Simples	Estrutura Composta	Estrutura Encadeada/Aninhada	Estrutura Aninhada/Encadeada
<pre>if condição:     comandos</pre>	<pre>if condição:     comandos else:     comandos</pre>	<pre>if condição1:     comandos elif condição2:     comandos else:     comandos</pre>	<pre>if condição 1:     comandos     if condição 2:         comandos     else:         comandos else:     comandos</pre>

**Exemplo resolvido:** ler do usuário a temperatura do dia de hoje (em °C). Se o valor estiver abaixo de 15°C, deve-se mostrar na tela a mensagem: “*está frio!*”. Se estiver acima de 25°C, mostrar a mensagem: “*Está calor!*”. Caso contrário, mostrar a mensagem: “*Temperatura agradável!*”.

```
# lê valor do usuário
t = float(input("Digite a temperatura
(em °C) de hoje:"))

# testa o valor lido
if (t < 15):
    print("Está frio!")
elif (t < 25):
    print("Temperatura agradável")
else:
    print("Está calor!")
```

Também é possível testar mais de uma condição ao mesmo tempo.

```
if (15<=t<=25):
    print('Temp. agradável')
```

```
if (t>=15 and t<=25):
    print('Temp. agradável')
```

Essas duas formas são equivalentes. Aqui, o *if* só executa se **as duas** condições forem verdadeiras **ao mesmo tempo**.

```
if (t<15 or t>25):
    print('Temp. desagradável: quente ou frio')
```

Aqui, o *if* executa se pelo menos uma das condições for verdadeira



## PYTHON PARA CIÊNCIA DE DADOS

### FUNÇÕES (MODULARIZAÇÃO)



Na programação, é possível **modularizar** trechos do código por meio de **funções**.

**Função:** conjunto de comandos que executam uma **tarefa particular** agrupados em um **bloco**, que recebe um **nome** e através deste pode ser **chamado**.

Abaixo uma comparação entre a solução tradicional e a solução usando função para um programa que calcula a soma entre dois valores lidos do usuário.

#### PROGRAMA – SOLUÇÃO TRADICIONAL

```
# 1ª etapa: lê dois valores do usuário
a = int(input("Digite o 1º valor:"))
b = int(input("Digite o 2º valor:"))

# 2ª etapa: soma os valores de forma simples
c = a+b;

# 3ª etapa: mostra o resultado na tela
print("A soma vale:",c)
```

Nesta solução, os comandos são executados sequencialmente, de cima para baixo, sem nenhum desvio.

#### PROGRAMA – SOLUÇÃO COM FUNÇÃO

```
# função de soma
def soma(x,y):
    result = x+y
    return result

# 1ª etapa: lê dois valores do usuário
a = int(input("Digite o 1º valor:"))
b = int(input("Digite o 2º valor:"))

# 2ª etapa: soma os valores por meio da função
c = soma(a,b)

# 3ª etapa: mostra o resultado na tela
print("A soma vale:",c)
```

Isso é uma **função!**

Aqui a função está sendo **chamada**

Nesta solução, ao chamar a função, a **execução** é desviada para que os comandos da função possam ser executados. Ao término, o programa retorna na linha em que ocorreu a chamada da função (**c = soma(a,b)**).

## EXERCÍCIOS

Em todos os exercícios abaixo:

- Utilize função no desenvolvimento.
  - Garanta que as entradas lidas do usuário sejam válidas, de maneira que o programa não prossiga enquanto forem informadas entradas inválidas.
- 1) Criar um programa em Python que, a partir dos valores de peso (em kg) e de altura (em m) de uma pessoa digitados pelo usuário, imprime o IMC daquela pessoa e também sua classificação conforme a diretriz de saúde abaixo. Utilizar duas funções: uma para calcular e retornar o IMC, e outra para obter e retornar a classificação de peso do indivíduo.

IMC	Classificação
(0,0 - 18,5)	Abaixo do peso
[18,5 - 25,0)	Peso saudável
[25,0 - $\infty$ )	Sobrepeso

parênteses representam *intervalo aberto* e colchetes representam *intervalo fechado*.

- 2) Alterar o programa em Python do exercício anterior de forma a considerar mais categorias de peso, conforme a relação abaixo:

IMC	Classificação
(0,0 - 18,5)	Abaixo do peso
[18,5 - 25,0)	Peso saudável
[25,0 - 30,0)	Sobrepeso
[30,0 - 35,0)	Obesidade grau I
[35,0 - 40,0)	Obesidade grau II (severa)
[40,0 - $\infty$ )	Obesidade grau III (mórbida)

parênteses representam *intervalo aberto* e colchetes representam *intervalo fechado*.

- 3) O programa em Python fornecido abaixo lê do usuário o peso (em kg), a altura (em m) e o sexo (m/f) de um paciente qualquer. Depois, por meio de funções, o programa calcula/obtem: o IMC do paciente, sua categoria de peso, seu peso ideal e sua situação, ou seja, quantos kg está acima ou abaixo do peso ideal. Por fim, o programa apresenta na tela essas quatro informações.

### Programa em Python

```
# -- definição das funções --
# ** os estudantes devem fazer **

# -- programa principal --
# entrada de dados (validar as entradas)
# ** os estudantes devem fazer **

# obtenção dos resultados
imc = calcula_imc(p,h)
cat = categoria_peso(imc)
pi = peso_ideal(h,s)
sit = situacao(p,pi)

# impressão dos resultados
print('\n*****')
print('IMC.....', round(imc,1))
print('CLASSIFICAÇÃO.....', cat)
print('PESO IDEAL.....', round(pi,1), 'kg')
print('SITUAÇÃO.....', sit)
```

Faça o que se pede:

- Copie o código fornecido e cole em um editor de sua preferência (Spyder, Pycharm, Idle, etc.).
- Crie a função `calcula_imc(p,h)`, que calcula e retorna o IMC do paciente em função de seu peso e altura, sabendo-se que  $IMC = \text{peso}/\text{altura}^2$ .
- Crie a função `categoria_peso(imc)`, que retorna a classificação de peso do paciente em função de seu IMC, sabendo-se que:
- Crie a função `peso_ideal(h,s)`, que calcula e retorna o peso ideal (em kg) do paciente em função de sua altura e sexo, sabendo-se que:

Sexo	Peso ideal
Masculino	$(72,7 \times \text{altura}) - 58,0$
Feminino	$(62,1 \times \text{altura}) - 44,7$

- Crie a função `situacao(p,pi)`, que calcula e retorna quantos kg o paciente está acima ou abaixo do seu peso ideal.

Veja abaixo um exemplo de execução do programa:

```
Digite o peso (em kg) do paciente: 70
Digite a altura (em m) do paciente: 1.95
Digite o sexo do paciente (m/f): f
*****
IMC..... 18.4
CLASSIFICAÇÃO..... abaixo do peso
PESO IDEAL..... 76.4 kg
SITUAÇÃO..... Você está 6.4 kg abaixo do seu peso ideal
```

- 4) Criar um programa em Python que, a partir de dois números quaisquer e também de uma operação aritmética (+, -, \*, /) digitados pelo usuário, apresenta na tela o resultado do cálculo solicitado. Utilize quatro funções, uma para cada operação matemática elementar. Exemplos de execução:

```
Digite o 1º valor:10
Operação desejada (+,-,*,/):+
Digite o 2º valor:30

*****
10.0 + 30.0 = 40.0
*****
```

```
Digite o 1º valor:20
Operação desejada (+,-,*,/):-
Digite o 2º valor:8

*****
20.0 - 8.0 = 12.0
*****
```

```
Digite o 1º valor:5
Operação desejada (+,-,*,/):*
Digite o 2º valor:3

*****
5.0 * 3.0 = 15.0
*****
```

```
Digite o 1º valor:10
Operação desejada (+,-,*,/):/
Digite o 2º valor:0
Erro! Digite o 2º valor:5

*****
10.0 / 5.0 = 2.0
*****
```

- 5) Criar um programa em Python que, a partir dos coeficientes 'a', 'b' e 'c' de uma equação do 2º grau digitados pelo usuário, apresenta na tela quantas raízes reais existem e quais são seus valores. As saídas possíveis são: "Não existem raízes reais", "Há uma única raiz real que vale:\_\_\_" e "Há duas raízes reais que valem \_\_\_ e \_\_\_". Utilizar uma única função para resolver o problema. Obs.: se necessário utilize a função **str( )** para converter um número em texto.
- 6) Criar um programa em Python que imprime se um triângulo é "equilátero", "isósceles" ou "escaleno" a partir das medidas dos três lados informadas pelo usuário.
- 7) Criar um programa em Python que imprime se um valor inteiro qualquer informado pelo usuário é "par" ou "ímpar".
- 8) Criar um programa em Python que imprime se um visitante do Hopi Hari está "liberado" ou "proibido" de andar na montanha russa do parque com base na idade (em anos) e no peso (em kg) informados pelo usuário. Para estar liberado, é preciso ser maior de 15 anos e não pesar mais que 120 kg.
- 9) Crie um programa em Python que, a partir do tempo (em minutos) de permanência de um veículo no estacionamento de um shopping informado pelo usuário, imprime o valor a ser pago para retirar o veículo. A cobrança é feita da seguinte forma:

Até 15 minutos, R\$ 3,00	De 15 a 30 minutos, R\$ 6,00	De 30 a 60 minutos, R\$ 10,00	Acima de 60 minutos, R\$10,00 + R\$ 1,00 por minuto excedente aos 60 iniciais.
--------------------------	------------------------------	-------------------------------	--