

# Sintaxe do if

```
if <condição>:  
    <comando1>  
    <comando2>
```

Observações:

- <condição> é uma expressão booleana, ou seja, uma expressão que retorna True ou False
- <comando1> e <comando2> só serão executados se a <condição> for True (verdadeira)

Praticar



Digite no IDLE:

```
x = 33
```

```
if x % 5 != 0:
```

```
    print("x não é multiplo de 5")
```

Observação: Se você redigitar o exercício acima, porém, com x valendo 30, ele não imprimirá a mensagem.

## Praticar



"""v1"""

```
a = int(input("Digite um numero: "))
```

```
if (a > 0) and (a < 10):
```

```
    print ("Entre zero e dez")
```

"""v2"""

```
a = int(input("Digite um numero: "))
```

```
if (0 < a < 10):
```

```
    print ("Entre zero e dez")
```

# Sintaxe do if..else

```
if <condição>:  
    <comando1>  
    <comando2>  
else:  
    <comando3>  
    <comando4>
```

Observações:

- <condição> é uma expressão booleana, ou seja, uma expressão que retorna True ou False
- <comando1> e <comando2> só serão executados se a <condição> for True (verdadeira)
- <comando3> e <comando4> só serão executados se a <condição> for False (falsa)

# Sintaxe do if..else

Praticar



Digite no IDLE e veja qual mensagem ele irá imprimir:

```
x = 30
```

```
if x % 5 != 0:
```

```
    print("x não é múltiplo de 5")
```

```
else:
```

```
    print("x é múltiplo de 5")
```

Observação: Teste com x recebendo outros valores

# Sintaxe do if..else



Faça um programa em Python que calcule 20% de aumento caso o salário informado pelo usuário seja menor que R\$ 2000,00 e, 10% se o salário for maior ou igual a R\$ 2000,00. Mostre o resultado ao usuário.

```
salario = float(input("Digite o valor salario: "))
```

```
if salario < 2000:
```

```
    salario = salario * 1.20
```

```
else:
```

```
    salario = salario * 1.10
```

```
print ("Salario = R$", salario)
```

Praticar



```
'''par ou impar'''
```

```
a = int(input("Digite um numero: "))
```

```
if (a%2==0):
```

```
    print ("PAR")
```

```
else :
```

```
    print ("IMPAR")
```

# If / elif / else

Praticar



Escreva um programa que determina se o valor inteiro informado pelo usuário é negativo, zero ou positivo.

```
num = int(input("Digite um valor inteiro:"))
```

```
if num < 0:
```

```
    print("Negativo")
```

```
elif num == 0:
```

```
    print ("Zero")
```

```
else:
```

```
    print ("Positivo")
```

## Praticar



```
'''notas'''
```

```
nota = float(input("Digite uma nota: "))
```

```
if nota < 3:
```

```
    print("Reprovado")
```

```
elif nota < 5:
```

```
    print("Recuperacao")
```

```
elif nota < 9:
```

```
    print("Aprovado")
```

```
else:
```

```
    print("Aprovado com louvor!")
```

**Melhoria:** Caso a entrada da nota seja Negativa OU Maior que 10, exiba a mensagem “ENTRADA INVALIDA”



## Praticar



```
'''calc'''
```

```
print ('Entre com numero / operador / numero')
```

```
num1 = float (input())
```

```
op = input()
```

```
num2 = float (input())
```

```
print (num1,op,num2,'=')
```

```
if op == '+' :
```

```
    print (num1 + num2)
```

```
elif op == '-' :
```

```
    print (num1 - num2)
```

```
elif op == '/' :
```

```
    print (num1 / num2)
```

```
elif op == '*' :
```

```
    print (num1 * num2)
```

```
elif op == '%' :
```

```
    print (num1 % num2)
```

```
elif op == '^' :
```

```
    print (num1 ** num2)
```

```
else :
```

```
    print ('Operador invalido.')
```

**Melhoria:** No caso da divisão real, se num2 for zero, ocorrerá o erro de Divisão por Zero. Neste caso o programa deverá exibir a mensagem “DIVISÃO POR ZERO”



```
'''baskara'''
```

```
import math
```

```
a = float(input("Digite A: "))
```

```
b = float(input("Digite B: "))
```

```
c = float(input("Digite C: "))
```

```
delta = b**2-4*a*c
```

```
if delta < 0:
```

```
    print ("Não tem raiz Real")
```

```
elif delta == 0:
```

```
    raiz1 = (-b)/(2*a)
```

```
    print ("Raiz x = ",raiz1)
```

```
else:
```

```
    raiz1 = (-b+math.sqrt(delta))/(2*a)
```

```
    raiz2 = (-b-math.sqrt(delta))/(2*a)
```

```
    print ("Raizes x1 = ",raiz1," x2= ",raiz2)
```

Uma empresa concederá um aumento de salário aos seus funcionários, variável de acordo com o cargo, conforme a tabela abaixo. Faça um programa que leia o salário e o código do cargo de um funcionário e calcule o novo salário. Se o cargo do funcionário não estiver na tabela, ele deverá, então, receber 40% de aumento. Mostre o salário antigo, o novo salário e a diferença.

Código	Cargo	Percentual
GER	Gerente	10%
ENG	Engenheiro	20%
TEC	Técnico	30%

Exemplo de execução:

Cargo: **ENG**  
Salário (R\$): **1230.50**  
Salário antigo: R\$ 1230.50  
Novo salário: R\$ 1476.60  
Diferença: R\$ 246.10

Outro exemplo de execução:

Cargo: **ASCEN**  
Salário (R\$): **1230.50**

Salário antigo: R\$ 1230.50  
Novo salário: R\$ 1722.70  
Diferença: R\$ 492.20



```
cargo = input('Entre com o código do cargo:').upper()
salario = float(input('Entre com o salario em R$:'))
if (cargo == 'GER'):
    perc = 0.1
elif (cargo == 'ENG'):
    perc = 0.2
elif (cargo == 'TEC'):
    perc = 0.3
else:
    perc = 0.4
aumento = perc*salario
novosal = salario + aumento
print('Salário antigo : R$',salario)
print('Novo Salário : R$',novosal)
print('Diferença : R$',aumento)
```

# Strings

- É um tipo de dado
- É uma coleção sequencial de caracteres
- Não poderá ser usada em cálculos
- Exemplo:
  - 'Python'
  - "True"
  - '523'

- Pode ser tratada como uma lista
- Exemplo: `pais = 'Brasil'`
  - `>>> pais[0]` → 'B'
  - `>>> pais[1]` → 'r'
  - `>>> pais[2]` → 'a'
  - `>>> pais[-1]` → 'l'
  - `>>> pais[-2]` → 'i'

# Comando: upper()

Converte todas as letras para maiúsculas

Sintaxe: `<dado>.upper()`

Observações:

- `<dado>` pode ser um dado ou uma variável do tipo string

Praticar



Digite no IDLE:

```
>>> pais = 'Brasil'
```

```
>>> pais.upper() → 'BRASIL'
```

```
>>> animal = "eLeFaNtE"
```

```
>>> animal.upper() → 'ELEFANTE'
```

# Comando: lower()

Converte todas as letras para minúsculas

Sintaxe: `<dado>.lower()`

Observações:

- `<dado>` pode ser um dado ou uma variável do tipo string

Praticar



Digite no IDLE:

```
>>> pais = 'BRASIL'
```

```
>>> pais.lower()    → 'brasil'
```

```
>>> animal = "eLeFaNtE"
```

```
>>> animal.lower()  → 'elefante'
```

# Comando: capitalize()

Coloca a primeira letra em maiúscula e as demais em minúsculas

Sintaxe: `<dado>.capitalize()`

Observações:

- `<dado>` pode ser um dado ou uma variável do tipo string

## Praticar



Digite no IDLE:

```
>>> pais = 'BRASIL'
```

```
>>> pais.capitalize() → 'Brasil'
```

```
>>> animal = "eLeFaNtE"
```

```
>>> animal.capitalize() → 'Elefante'
```



# Comando: strip()

Remove os espaços em branco existentes no início e no final da string

Sintaxe: `<dado>.strip()`

Observações:

- <dado> pode ser um dado ou uma variável do tipo string

## Praticar



Digite no IDLE:

```
>>> pais = '   BRASIL   '
```

```
>>> pais.strip()      → 'BRASIL'
```

```
>>> animal = '   eLeFaNtE   '
```

```
>>> animal.strip()    → 'eLeFaNtE'
```

# Comando: count()

Conta a quantidade de ocorrências de x na string

Sintaxe: `<dado>.count(x)`

Observações:

- <dado> pode ser um dado ou uma variável do tipo string

## Praticar



Digite no IDLE:

```
>>> "um elefante incomoda muita gente".count("e") → 5
```

```
>>> animal = "orangotango"
```

```
>>> animal.count("o") → 3
```

# Comando: replace()

Substitui o pedaço x do dado por y

Sintaxe: `<dado>.replace(x,y)`

Observações:

- <dado> pode ser um dado ou uma variável do tipo string

## Praticar



Digite no IDLE:

```
>>> "um gatinho incomoda muita gente".replace("gatinho","elefante")
```

→ 'um elefante incomoda muita gente'

```
>>> frase = "eu estou triste"
```

```
>>> frase.replace("triste","feliz")
```

→ 'eu estou feliz'

# Comando: find()

Informa a posição da **primeira** ocorrência de x

Sintaxe: `<dado>.find(x)`

Observações:

- <dado> pode ser um dado ou uma variável do tipo string

## Praticar



Digite no IDLE:

```
>>> "Senha incorreta".find('c') → 8
```

```
>>> 'áGuA'.capitalize().find('Á') → 0
```

```
>>> 'áGuA'.find('Á') → -1
```

# Comando: len()

Informa o comprimento de uma string

Sintaxe: `len(dado)`

Observações:

- <dado> pode ser um dado ou uma variável do tipo string

Praticar



Digite no IDLE:

```
>>> len("Senha incorreta")
```

→ 15

```
>>> len('áGuA')
```

→ 4