

Comentários em Python

In []: *#Utiliza-se o # para inserir comentários no código. Após usar #, a linha #É fundamental que o código seja inteiramente comentado, pois facilita...*

In []: *#para comentários multilinha podemos usar três aspas simples ou duplas (" #Na verdade, aspas triplas é usada para fazer documentação oficial do código*

```
"""
Testando
Comentário
Multilinha
"""
```

Imprimir na Tela

In [1]: *#O comando print serve para imprimir na tela*

```
print("Python é TOP")
```

Python é TOP

In [2]: *#É possível imprimir varios valores (por padrão separa com espaço e quebra*

```
print(12, 35)
print(23, 56)
```

12 35
23 56

In [3]: *#É possível definir o separador com o atributo sep*

```
print(23, 56, sep="-")
```

23-56

In [5]: *#Por padrão o Python quebra a linha (\n) em cada print*

#É possível mudar este padrão com o atributo end

#observe que agora não tem mais quebra de linha

#por padrão o end="/n"

```
print(23, 56, sep="-", end="yy")
```

23-56yy

In [6]: *#Barra invertida é caractere de escape, quando se deseja imprimir " por exemplo*

```
print('Python é \"TOP\"')
```

Python é "TOP"

Tipos de Dados

In []: *#inteiro*

```
print(1)
```

In []: *#float*

```
print(1.3)
```

In [7]: *#Com type é retornado o tipo do dado*

```
print(type(5))
```

<class 'int'>

```
In [8]: #booleano
#neste caso retorna False, pois 10 não é menor que 5
print(10<5)
```

False

Casting ou Conversão de Tipos

```
In [9]: #Com o uso da função float é convertido 1 de int para float
print(type(float('1')))
```

<class 'float'>

```
In [12]: #neste caso convertemos 11 para string e concatenamos com caractere 'b'
print(str(1) + 'b')
```

1b

Variáveis

```
In [ ]: #As variáveis podem ser definidas como espaços de memória capazes de armazenar
#Por padrão, é recomendado que o nome das variáveis sejam minúsculos e sejam significativos
#É muito importante que o nome da variável seja relacionado ao valor que ela armazena
```

```
In [18]: #É criada a variável idade e atribuído o valor de 30
idade=30
print(idade)
```

30

```
In [20]: #Como idade é igual a 30, a comparação idade>=18 é true. Logo a variável maior recebe True
maior=idade>=18
print(maior)
```

False

Operadores Matemáticos

```
In [22]: adicao = 10 + 10
print('Adição', adicao)

subtracao = 10 - 5
print('Subtração', subtracao)

multiplicacao = 10 * 10
print('Multiplicação', multiplicacao)

#Divisão sempre retorna um float
divisao = 10 / 3
print('Divisão', divisao)
```

Adição 20

Subtração 5

Multiplicação 100

Divisão 3.3333333333333335

```
In [23]: #Utilizando // sempre retorna valor truncado sem decimais
divisao_inteira = 10 // 3
print('Divisão inteira', divisao_inteira)
```

```
exponenciacao = 2 ** 10
print('Exponenciação', exponenciacao)

#Retorna o resto da divisão. Muito útil para calcular paridade
modulo = 55 % 2
print('Módulo', modulo)
```

Divisão inteira 3
Exponenciação 1024
Módulo 1

In []: *#Imprimindo um expressão, o Python retorna verdadeiro ou falso*

```
print(10 % 8 == 0)
print(16 % 8 == 0)
```

In []: *#Com * da pra repetir caracteres*

```
repetindo='P' * 3
print(repetindo)
```

Precedência de Operadores

In []: *#Em uma operação, a precedência define quais cálculos serão realizados primeiro. No Python, temos a seguinte ordem de precedência:*

```
#primeiro será executado o que tiver dentro de parenteses
# 1. (n + n) - 

# Depois serão executadas as potências
# 2. **

# Depois as operações de multiplicação, divisão e resto.
# 3. * / // %

#E por fim operações de soma e subtração
# 4. + -
```

In [24]: *#Abaixo é calculado o IMC como exemplo. Observe que primeiro será calculado a divisão entre peso e altura ao quadrado.*

```
nome = 'Paulo'
altura = 1.86
peso = 77
imc = peso / altura ** 2
print('O IMC do ', nome, ' é ', imc)
```

O IMC do Paulo é 22.25690831309978

Formatando String

In [25]: *#As "f-strings" permitem formatar strings. Para usar basta inserir 'f' no começo da string e inserir {} para inserir variáveis.*

```
nome = 'Paulo'
altura = 1.86
peso = 77
imc = peso / altura ** 2

#As "f-strings" permitem formatar strings. Para usar basta inserir 'f'
#Inserindo '.2f' formata com duas casas decimais e faz o arredondamento
linha_1 = f'{nome} tem {altura:.2f} de altura,'
linha_2 = f'{peso} quilos e seu imc é '
linha_3 = f'{imc:.2f}'

print(linha_1)
```

```
print(linha_2)
print(linha_3)
```

Paulo tem 1.86 de altura,
77 quilos e seu imc é
22.26

Inserindo dados com INPUT

```
In [26]: #input permite que seja inserido um valor
nome = input('Qual o seu nome? ')
print(f'O seu nome é {nome}')

numero_1 = input('Digite um número: ')
numero_2 = input('Digite outro número: ')

int_numero_1 = int(numero_1)
int_numero_2 = int(numero_2)

print(f'A soma dos números é: {int_numero_1 + int_numero_2}')
```

O seu nome é Paulo
A soma dos números é: 6

Condisional IF / ELIF / ELSE

```
In [29]: #O Python possui uma estrutura condicional formada pelo IF (SE), ELIF (se
#o exemplo imprime a situação do aluno estar aprovado ou não
media=30
if (media>=60):
    print("Recuperação")
    print("Na próxima semana")
else:
    print("Reprovou")
```

Reprovou

```
In [ ]: #Em algumas situações é necessário encadear condições, neste caso precisa
#o exemplo imprime a situação do aluno de acordo com a média
media=30
if (media>=90): print("Aluno Excelente")
elif (media>=70 and media<=89): print("Aluno Bom")
elif (media>=60 and media<=69): print("Desempenho Mediano")
else: print("Reprovou")
```

Operadores de Comparação e Lógicos

```
In [ ]: #Operadores de Comparação
# OP      Significado          Exemplo (True)
# >      maior                2 > 1
# >=     maior ou igual        2 >= 2
# <      menor                1 < 2
# <=     menor ou igual        2 <= 2
# ==     igual                'a' == 'a'
# !=     diferente            'a' != 'b'
```

```
In [ ]: # Operadores lógicos

#and (e): todas as condições precisam ser verdadeiras para o retorno ser
```

```
#or (ou): apenas uma das condições precisa ser verdadeira para o retorno
#not (não): inverte o retorno da expressão
```

```
In [ ]: #o exemplo abaixo pede para entrar no programa e digitar uma senha
#Se o usuário digitar "E" e a senha correta, imprimir "Entrar", senão "Sair"

entrada = input('[E]ntrar [S]air: ')
senha_digitada = input('Senha: ')

senha_permitida = '123456'

if entrada == 'E' and senha_digitada == senha_permitida:
    print('Entrar')
else:
    print('Sair')
```

```
In [ ]: #Agora vamos tratar para que o usuário também possa digitar "e" minúsculo
entrada = input('[E]ntrar [S]air: ')
senha_digitada = input('Senha: ')

senha_permitida = '123456'
if (entrada == 'E' or entrada == 'e') and senha_digitada == senha_permitida:
    print('Entrar')
else:
    print('Sair')
```

```
In [32]: #O operador NOT inverte o resultado de uma expressão
aprovado=True
print(not aprovado)
```

False

```
In [ ]: #Operadores IN e NOT IN

# As Strings são iteráveis (pode ser navegável pela posição)
# 0 1 2 3 4
# P a u l o
# -5-4-3-2-1
```

```
In [33]: #pegando elemento da posição 2
nome = 'Paulo'
print(nome[2])
```

u

```
In [34]: #pegando elemento da posição -3 (sentido direita para esquerda)
nome = 'Paulo'
print(nome[-3])
```

u

```
In [37]: # Operador in verifica se existe 'ulo' na string
nome = 'Paulo'
print('ulo' in nome)
```

True

```
In [ ]: # Operador not in verifica se não existe 'ulo' na string. Retorna false p
nome = 'Paulo'
print('ulo' not in nome)
```

Interpolação

```
In [ ]: #Interpolação é uma forma de formatar a impressão de valores na tela.

#Os simbolos abaixo definem o tipo de dado
# s - string
# d e i - int
# f - float
# x e X - Hexadecimal (ABCDEF0123456789)

nome = 'Paulo'
preco = 1000.95897643
#vamos formatar com duas casas decimais %2.f
variavel = '%s, o preço é R$%.2f' % (nome, preco)
print(variavel)

#X converte um valor para hexadecimal. O valor 08 estabelece 8 casas
valor=1000
print('0 hexadecimal de %d é %08X' % (valor, valor))
```

f-string

```
In [38]: #f-string é uma forma mais moderna de fazer formatação nos textos
variavel = 'ABC'
print(f'{variavel}')

#completa o que faltar a esquerda para completar 10 dígitos
print(f'{variavel: >10}')

#coloca a direita e completa com espaços para ter 10 dígitos
print(f'{variavel: <10}')

#deixa a variável no meio e completa para completar 10 dígitos
print(f'{variavel: ^10}')

#converte para hexadecimal com 8 casas
print(f'0 hexadecimal de 1000 é {1000:08X}')

#formata com duas casas decimais (arredonda se necessário)
valor=1005.4556789
print(f'0 valor do produto é {valor:.2f}')
```

ABC
ABC
ABC
ABC
0 hexadecimal de 1000 é 000003E8
0 valor do produto é 1005.46

Fatiando Strings

```
In [39]: #Fatiamento (slice) é pegar parte de uma string maior
#Para fatiar, temos a seguinte sintaxe: nome_string[posição inicial : posição final]

variavel = 'Palmeiras'

#posição 1 até 5 (mas 5 não entra)
print(variavel[1:5])
```

```

#retorna posição 1 até o fim
print(variavel[1:])

#retorna do inicio até posicao 4 (5 não entra)
print(variavel[:5])

#len() retorna o número de caracteres
print(len(variavel[1:]))

#retorna de 0 até o tamanho da variavel pulando de 2 em 2 (step)
print(variavel[0:len(variavel):2])

#o índice negativo pega de traz para frente
print(variavel[-1:-10:-1])

#Também retorna todos os valores invertidos
print(variavel[::-1])

```

```

alme
almeiras
Palme
8
Plers
sariemlaP
sariemlaP

```

In [40]: *#Strings são imutáveis, ou seja, não é possível alterar uma parte da string*
#Por exemplo, ao tentar trocar a letra 'o' por 'a' gera erro

```

nome = 'Paulo'
nome[4]='a'

```

```

-----
-
TypeError                                     Traceback (most recent call last)
t)
Cell In[40], line 4
    1 #Strings são imutáveis, ou seja, não é possível alterar uma parte
    2 da string
    3 #Por exemplo, ao tentar trocar a letra 'o' por 'a' gera erro
    4 nome = 'Paulo'
----> 4 nome[4]='a'

TypeError: 'str' object does not support item assignment

```

In []: *#É preciso criar uma nova string e fazer a alteração na atribuição do valor*

```

nome='Paulo'
novo_nome = f'{nome[0:4]}a'
print(novo_nome)

```

Try - Exception

In [41]: *#Para garantir que o código não trave ou gere erros ilegíveis ao usuário,*
#try -> tentar executar o código
#except -> executa se ocorreu algum erro ao tentar executar

```

numero_str = input('Vou dobrar o número que vc digitar: ')
try:
    numero_float = float(numero_str)
    print('FLOAT:', numero_float)

```

```

        print(f'0 dobro de {numero_str} é {numero_float * 2:.2f}')
except:
    print('Isso não é um número')

```

FLOAT: 4.0
0 dobro de 4 é 8.00

Variáveis e Constantes

```

In [ ]: #As variáveis costumam trocar de valores constantemente
#Uma constante, em teoria, não deve trocar de valores.
#no Python, é um padrão deixar as variáveis minúsculas e constantes maiúsculas

#variável
raio=2

#constante
PI=3.14

```

```

In [ ]: #Exemplo: Faça um programa que peça uma temperatura em celcius e converta para fahrenheit
#Fórmula: C*1.8+32

```

```

In [42]: valor = input("Digite a temperatura em graus Celsius: ")

try:
    celsius = float(valor) # tenta converter para número
    fahrenheit = (celsius * 1.8) + 32
    print(f"{celsius:.2f}°C equivalem a {fahrenheit:.2f}°F")
except ValueError:
    print("Erro: você deve digitar um valor numérico.")

```

45.00°C equivalem a 113.00°F

In []:

```

In [ ]: # 1) Faça um programa que peça a entrada de um número ao usuário, e retorna se é par ou ímpar.
#Caso o usuário não digite um número válido, o programa deve retornar uma mensagem de erro.

# 2) Faça um programa que pergunte a hora ao usuário e, baseando-se no horário, informe se é manhã, tarde ou noite.
#Ex: Bom dia 0-11, Boa tarde 12-17 e Boa noite 18-23.

# 3) Faça um programa que peça o primeiro nome do usuário. Se o nome tiver 4 letras ou menos, imprima "Seu nome é curto";
#Se o nome tiver entre 5 e 6 letras, imprima "Seu nome é normal";
#Caso tenha mais de 6 letras, escreva "Seu nome é grande", e imprima "Parabéns, seu nome é grande".
#Para verificar o tamanho do nome use a função len()

# 4) Faça um programa que calcule as raízes de uma equação de segundo grau.
#O usuário deve digitar os valores de 'a', 'b' e 'c'.
#Considere a equação como ax²+bx+c

```

```

In [4]: try:
    entrada=input()
    numero=float(entrada)
    if (numero % 2==0):
        print("Par")
    else:
        print("Impar")
except:
    print("Você não digitou um número")

```

Impar

```
In [7]: try:
    entrada=input()
    hora=int(entrada)
    if hora>=0 and hora<=11:
        print("Bom dia")
    elif hora>=12 and hora<=17:
        print("Boa tarde")
    elif hora>=18 and hora<=23:
        print("Boa noite")
    else:
        print("Não existe esta hora")
except:
    print("Não digitou um número")
```

Não existe esta hora

```
In [9]: nome=input()
tamanho=len(nome)
if tamanho>=1 and tamanho<=4:
    print("Seu nome é curto")
elif tamanho>=5 and tamanho<=6:
    print("Seu nome é normal")
else:
    print("Seu nome é grande")
    print(nome[0:6])
```

Seu nome é grande

Franci

```
In [11]: import math

a=float(input("Digite A"))
b=float(input("Digite B"))
c=float(input("Digite C"))

if a==0:
    print("Não é equação do segundo grau")
else:
    delta=b**2-4*a*c
    if delta>0:
        x1=(-b + math.sqrt(delta))/(2*a)
        x2=(-b - math.sqrt(delta))/(2*a)
        print(f"x1={x1:.2f}, x2={x2:.2f}")
    else:
        print("Não há raízes reais")
```

Não há raízes reais

```
In [ ]:
```