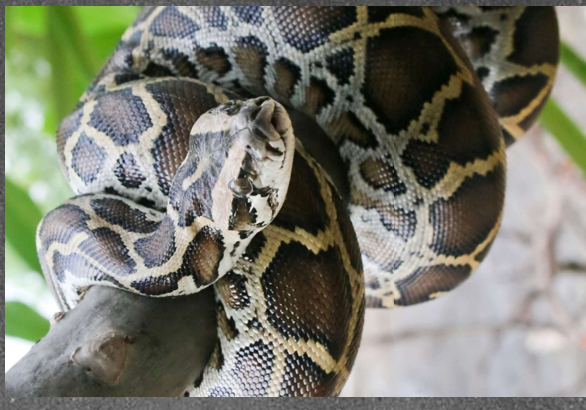
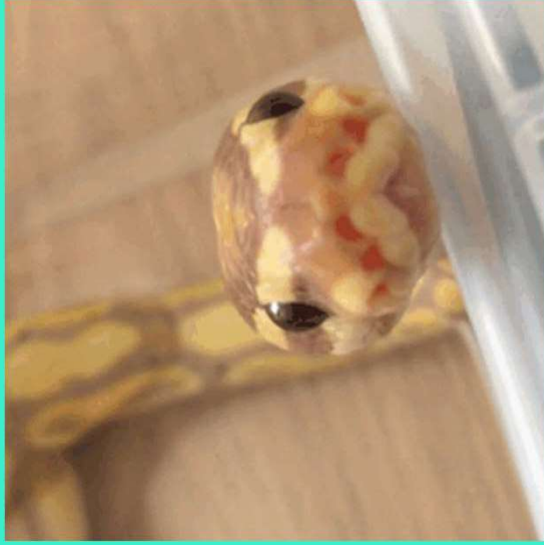
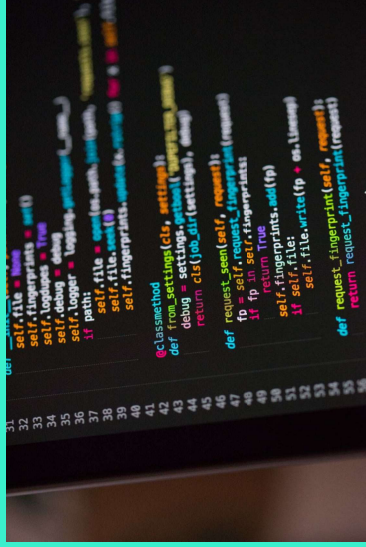


# INTRODUÇÃO A PROGRAMAÇÃO EM PYTHON



# INTRODUÇÃO A PROGRAMAÇÃO EM PYTHON





## O que é Python?

**Definição:** Python é uma linguagem de programação de alto nível, interpretada e de propósito geral.

**Criador e História:** Foi criada por Guido van Rossum e lançada pela primeira vez em 1991.

**Nome:** O nome "Python" vem do grupo de comédia britânico "Monty Python", e não da cobra.





## O que é Python?

Python é uma linguagem de programação de alto nível e de propósito geral . Sua filosofia de design enfatiza a legibilidade do código com o uso de recuo significativo.

Python suporta vários paradigmas de programação , incluindo programação estruturada (particularmente procedural), orientada a objetos e funcional . É frequentemente descrito como uma linguagem "baterias incluídas" devido à sua biblioteca padrão abrangente .

# MULTIPLICAÇÃO ASSEMBLY

```
assembly
Copy code

section .data
    num1 dw 5          ; Primeiro número (5)
    num2 dw 10         ; Segundo número (10)
    result dw 0        ; Variável para armazenar o resultado

section .text
    global _start

_start:
    ; Carrega os valores de num1 e num2 nos registradores
    mov ax, [num1]     ; Carrega num1 no registrador AX
    mov bx, [num2]     ; Carrega num2 no registrador BX

    ; Multiplica AX por BX, o resultado fica em AX
    mul bx             ; AX = AX * BX

    ; Armazena o resultado em result
    mov [result], ax   ; Armazena o resultado em result

    ; Termina o programa
    mov eax, 60        ; Código do syscall para exit
    xor edi, edi       ; Código de retorno 0
    syscall
```

# MULTIPLICAÇÃO EM PYTHON

```
python
# Definindo os números
numero1 = 6
numero2 = 7

# Multiplicando os números
resultado = numero1 * numero2

# Exibindo o resultado
print(f"O resultado da multiplicação de {numero1} por {numero2} é: {resultado}")
```

Copy code



# IDENTIFICAÇÃO



```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

# VARIÁVEIS

```
C: > Users > Wagner Coutinho > Documents > Raro Academy > Aula 2 > aula.py > ...
1  # Definindo variáveis
2  nome = "Wagner Coutinho"
3  altura = 1.70
4  idade = 30
5  a = 5 + 2j
6  gosta_anime = False
7
8  # Exibindo os valores das variáveis e seus tipos
9  print(f"Nome: {nome}")
10 print(type(nome)) # Tipo da variável 'nome'
11
12 print(f"Altura: {altura}")
13 print(type(altura)) # Tipo da variável 'altura'
14
15 print(f"Idade: {idade}")
16 print(type(idade)) # Tipo da variável 'idade'
17
18 print(f"Gosta de novela: {gosta_anime}")
19 print(type(gosta_anime)) # Tipo da variável 'gosta_anime'
20
21 print(f"A: {a}")
22 print(type(a)) # Tipo da variável 'a'
```



# OPERADORES ARITMÉTICOS

Operador	Descrição	Sintaxe
+	Adição: adiciona dois operandos	$x + y$
-	Subtração: subtrai dois operandos	$x - y$
*	Multiplicação: multiplica dois operandos	$x * y$
/	Divisão (float): divide o primeiro operando pelo segundo	$x / y$
//	Divisão (piso): divide o primeiro operando pelo segundo	$x // y$
%	Módulo: retorna o resto quando o primeiro operando é dividido pelo segundo	$x \% y$
**	Potência: Retorna primeiro elevado a potência segundo	$x ** y$

# OPERADORES COMPARAÇÃO

Operador	Descrição	Sintaxe
>	Maior que: Verdadeiro se o operando esquerdo for maior que o direito	<code>x &gt; y</code>
<	Menor que: Verdadeiro se o operando esquerdo for menor que o direito	<code>x &lt; y</code>
==	Igual a: Verdadeiro se ambos os operandos forem iguais	<code>x == y</code>
!=	Diferente de - Verdadeiro se os operandos não forem iguais	<code>x != y</code>
>=	Maior ou igual a Verdadeiro se o operando esquerdo for maior ou igual ao direito	<code>x &gt;= y</code>
<=	Menor ou igual a Verdadeiro se o operando esquerdo for menor ou igual ao direito	<code>x &lt;= y</code>
é	<code>x</code> é igual a <code>y</code>	<code>x é y</code>
não é	<code>x</code> não é igual a <code>y</code>	<code>x não é y</code>
== é um operador de atribuição e != operador de comparação		

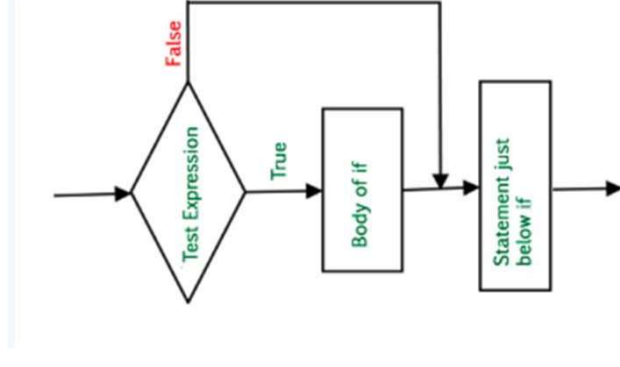
# OPERADORES BOOLEANOS

Operador	Descrição	Sintaxe
e	E Lógico: Verdadeiro se ambos os operandos forem verdadeiros	<code>x e y</code>
ou	OR Lógico: Verdadeiro se um dos operandos for verdadeiro	<code>x ou y</code>
não	NOT Lógico: Verdadeiro se o operando for falso	<code>não x</code>

# FLUXO DE CONTROLE IF

If é a declaração mais simples de tomada de decisão. Ele é usado para decidir se uma determinada instrução ou bloco de instruções será executada ou não.

Se uma determinada condição for verdadeira, um bloco de instruções será executado, caso contrário, não.



# FLUXO DE CONTROLE

## IF/ELSE

if (condição):

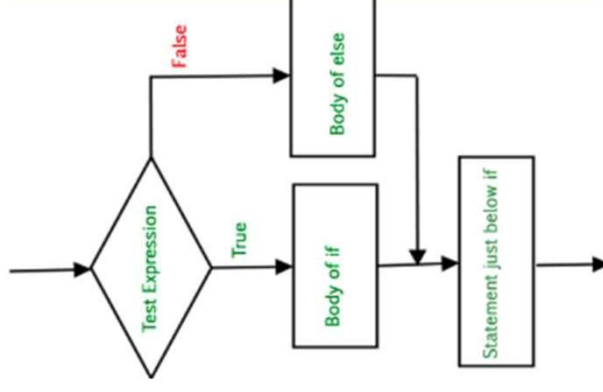
# Executa este bloco do if

# Condição é verdadeira

else:

# Executa este bloco do if

# Condição é falsa



```
Python Console x
In [51]: i = 20
...: if (i < 15):
...:     print("é menor que 15")
...: else:
...:     print("é maior que 15")
...:     print("está fora do bloco else")
...:
é maior que 15
está fora do bloco else
In [52]:
```



# FLUXO DE CONTROLE

## IF/ELSE/ELIF

if (condição):

- # Executa este bloco do if

- # Condição é verdadeira

elif (condição): # Executa este bloco do if

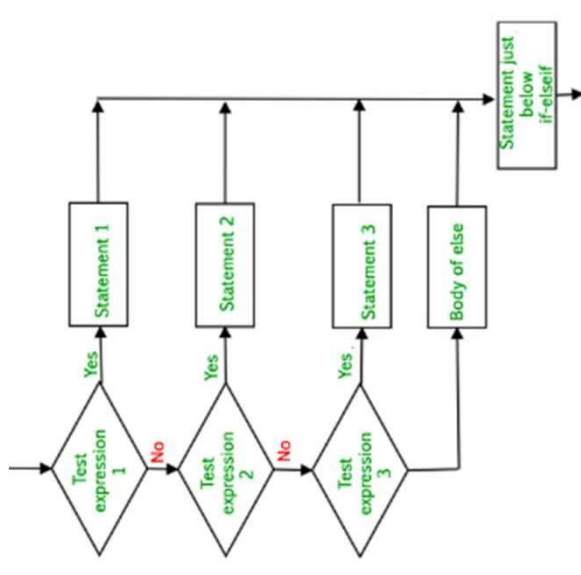
- # Segunda Condição é verdadeira

...

else:

- # Executa este bloco do if

- # Condição é falsa



```
Python Console ×
In [52]: i = 20
...: if (i == 10):
...:     print("i is 10")
...: elif (i == 15):
...:     print("i is 15")
...: elif (i == 20):
...:     print("i is 20")
...: else:
...:     print("nenhuma das condições anteriores")
...:
i is 20
```



# THANK YOU

Wagner Coutinho

<https://www.linkedin.com/in/wagner-coutinho-mf/>

[wagner.filho@rarolabs.com.br](mailto:wagner.filho@rarolabs.com.br)

[www.adatum.com](http://www.adatum.com)