

Fundamentos de Python

Controle de Fluxo e Operadores básicos

Agora que você já sabe falar com o computador, chegou a hora de ensiná-lo a decidir o que fazer em cada situação. "Se for maior de idade, diga que pode votar" é um exemplo disso!

Decidindo como um robô

Imagine que você está programando um robô para andar, falar ou até brincar com você. Mas o robô precisa de regras claras. Por exemplo:

- Se estiver chovendo, pegue o guarda-chuva.
- Se a comida acabar, peça mais.
- Se o botão for apertado 10 vezes, acenda a luz.

Em Python, essas regras são criadas com comandos de controle de fluxo:

Condições: if, elif, else.

Mas antes, precisamos entender o que são operadores.

Operadores: como manipular os dados

Operadores ajudam a comparar dados e decidir caminhos diferentes para o programa. São muito usados nas condições dos if, nos laços de repetição e nos cálculos em geral. Vamos conhecê-los:

Operadores Aritméticos	
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
%	Resto da Divisão

```
soma = 5 + 3
print(soma) # 8
```

Operadores de Comparação	
==	Igual a
!=	Diferente de
>	Maior que
<	Menor que
>=	Maior ou igual
<=	Menor ou igual

```
idade = 15
print(idade >= 18) # False
```

As **condições** são comandos que dizem ao programa o que fazer dependendo de alguma situação:

```
if idade >= 18:
    print("Você é maior de idade.")
elif idade == 17:
    print("Quase lá!")
else:
    print("Ainda não pode votar.")
```

Repetições: while e for

Eles servem para repetir ações:

- while: repete enquanto a condição for verdadeira
- for: repete para cada item ou número de vezes desejado

```
contador = 1
while contador <= 5:
    print("Repetindo...", contador)
    contador += 1
```

```
for letra in "Python":
    print(letra)
```

range()

range() cria uma sequência de números. Usamos com for para repetir ações:

```
for i in range(3):
    print("Olá", i)
```

break e continue

- break: para tudo e sai do laço.
- continue: pula para a próxima volta sem terminar a atual.

```
for i in range(5):
    if i == 3:
        break
    print(i)
```