



Aula 06 — Condições em Python

1. O que são condições?

Condições permitem **tomar decisões** no programa.

Em Python, usamos as palavras-chave:

- `if` → se;
 - `elif` → senão se;
 - `else` → senão.
-

2. Estrutura básica

```
if condição:
    # bloco de código se a condição for verdadeira
else:
    # bloco de código se a condição for falsa
```

3. Exemplos práticos

Condicional simples

```
nome = input("Digite seu nome: ")

if nome == "Marcelo":
    print("Que nome lindo você tem!")

print(f"Bom dia, {nome}!")
```

➔ O bloco `if` só roda se a condição for verdadeira.

➔ Não tem `else`, então pode executar ou não.

Condicional composta

```
nome = input("Digite seu nome: ")

if nome == "Marcelo":
    print("Que nome lindo você tem!")
else:
    print("Seu nome é tão normal...")

print(f"Bom dia, {nome}!")
```

➔ Aqui o programa tem dois caminhos: se for verdadeiro executa o `if`, senão executa o `else`.

Condicional encadeada (elif)

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

if nota >= 9:
    print("Excelente!")
elif nota >= 7:
    print("Boa nota!")
elif nota >= 5:
    print("Recuperação")
else:
    print("Reprovado")
```

➔ Usa-se `elif` quando temos **mais de duas condições**.

Condição em uma linha (operador ternário)

```
tempo = int(input("Quantos anos tem seu carro? "))

print("Carro novo" if tempo <= 3 else "Carro velho")
```

➔ Esse formato é útil quando a condição é simples.

4. Operadores relacionais

- `==` → igual
- `!=` → diferente
- `>` → maior
- `<` → menor
- `>=` → maior ou igual
- `<=` → menor ou igual

Exemplo:

```
x = 10
y = 20
print(x == y)  # False
print(x < y)   # True
```

5. Operadores lógicos

- `and` → todas condições precisam ser verdadeiras.
- `or` → basta uma ser verdadeira.
- `not` → inverte o valor lógico.

Exemplo:

```
idade = 18
tem_carteira = True

if idade >= 18 and tem_carteira:
    print("Pode dirigir")
else:
    print("Não pode dirigir")
```

6. Boas práticas

- ✓ Use **indentação de 4 espaços** (obrigatório no Python).
- ✓ Use nomes de variáveis claros (*nota*, *media*) em vez de *n1*, *n2* sempre que possível.
- ✓ Prefira `elif` em vez de `if` aninhados desnecessários.
- ✓ Quando a condição for curta, pode usar `if ... else` em uma linha.