



Aula 3: Estruturas Condicionais em Python

FUNDAMENTOS DE PROGRAMAÇÃO

CodeRoots

Aprendendo a usar “if”, “elif” e “else” para tomar decisões no código

Assim como nós tomamos decisões no dia a dia, os programas também precisam escolher o que fazer em determinadas situações.

Exemplo do dia a dia:

“Se estiver frio, coloco um casaco.
Senão, saio de camiseta.”

O que é uma estrutura condicional

- Uma estrutura condicional permite que o programa escolha um caminho diferente dependendo de uma condição.
- Essa condição sempre resulta em verdadeiro (True) ou falso (False).
- Isso é o que chamamos de tomar decisões no código.

Em resumo:

O programa verifica uma condição →
se for verdadeira, faz algo →
se for falsa, faz outra coisa (ou nada).

O comando if

O comando **if** significa “se”.

Ele é usado para verificar uma condição e executar um bloco de código se ela for verdadeira.

Sintaxe (Como deve ser escrito):

```
● ● ●  
if condição:  
    # código executado se a condição for verdadeira
```

Exemplo:

```
● ● ●  
idade = 20  
  
if idade >= 18:  
    print("Você é maior de idade.")
```



- O programa testa se `idade >= 18`.
- Como é verdadeiro, ele executa o `print()`.
- Se fosse falso, não faria nada.

O comando `else`

O `else` significa “senão”.

Ele é usado quando queremos executar algo diferente caso a condição seja falsa.

Sintaxe (Como deve ser escrito):

```
● ● ●  
if condição:  
    # se for verdadeiro  
else:  
    # se for falso
```

Exemplo:

```
● ● ●  
  
idade = 16  
  
if idade >= 18:  
    print("Maior de idade.")  
else:  
    print("Menor de idade.")
```



- Como `idade >= 18` é falsa, o Python ignora o primeiro bloco e executa o do `else`.

O comando **elif**

O **elif** é a abreviação de “else if” (senão se).

Usamos ele quando queremos verificar mais de uma condição.

Sintaxe (Como deve ser escrito):

```
● ● ●  
if condição1:  
...  
elif condição2:  
...  
else:  
...
```

Exemplo:

```
● ● ●  
nota = 7  
  
if nota >= 9:  
    print("Excelente!")  
elif nota >= 6:  
    print("Aprovado.")  
else:  
    print("Reprovado.")
```



- O Python verifica a primeira condição (`nota >= 9`).
- Como é falsa, ele vai pro `elif`.
- `nota >= 6` é verdadeira, então ele executa o bloco “Aprovado”.
- Se nenhuma fosse verdadeira, cairia no `else`.

Operadores de comparação

As condições dependem de operações lógicas.

Essas são as mais usadas:

Operador	Significado	Exemplo	Resultado
<code>==</code>	Igual	<code>5 == 5</code>	VERDADEIRO
<code>!=</code>	Diferente	<code>5 != 3</code>	VERDADEIRO
<code>></code>	Maior que	<code>8 > 6</code>	VERDADEIRO
<code><</code>	Menor que	<code>4 < 2</code>	FALSO
<code>>=</code>	Maior ou igual	<code>10 >= 10</code>	VERDADEIRO
<code><=</code>	Menor ou igual	<code>7 <= 5</code>	FALSO

- O operador `=` atribui valor, e `==` compara valores.

Condições compostas

Podemos combinar mais de uma condição usando operadores lógicos:

Operador	Significado	Exemplo	Resultado
and	As duas devem ser verdadeiras	(idade >= 18 and idade < 60)	VERDADEIRO
or	Basta uma ser verdadeira	(nota >= 9 or nota == 10)	VERDADEIRO
not	Inverte o resultado	not (5 > 2)	FALSO

Exemplo:

```
● ● ●  
idade = 25  
if idade >= 18 and idade < 60:  
    print("Adulto")
```

Importância da indentação

- No Python, a indentação (espaço antes do código) é obrigatória.
- É assim que o Python entende quais linhas fazem parte de qual bloco.

Exemplo correto:

```
● ● ●  
if True:  
    print("Dentro do if")  
print("Fora do if")
```

Exemplo incorreto:

```
● ● ●  
if True:  
print("Dentro do if") # Erro de indentação
```

Boas práticas

- Use condições simples e claras.
- Sempre comente trechos importantes do código.
- Mantenha indentação correta.
- Teste seu código com vários valores diferentes.



Obrigado por participar
da aula de hoje.

CodeRoots