

CURSO DE PYTHON



PARA INICIANTE



AULA 03: ESTRUTURAS CONDICIONAIS

Estruturas condicionais Permitem que o código tome decisões com base em condições específicas. O comando mais comum é o **if**, que executa um bloco de código se uma condição for verdadeira. Também é possível usar **elif** para testar outras condições e **else** para executar um código caso nenhuma das condições anteriores seja atendida.

- Permitem que um programa tome decisões com base em condições.
- Elas controlam o fluxo do código com base em valores booleanos (True ou False).

if condição

código executado se a condição for verdadeira



```
33 self.fingerprints = ...
34 self.logdups = True
35 self.debug = debug
36 self.logger = logging
37 if path:
38     self.file = open
39     self.file.seek(
40     self.fingerprin
41
42 @classmethod
43 def from_settings(cls,
44 debug = settings.
45 return cls(job_di
46
47 def request_seen(self
48 fp = self.request
49 if fp in self.fi
50     return True
51 self.fingerprint
52 if self.file:
53     self.file.w
54
55 def request_fingerp
56     return request
```

ESTRUTURA 'IF'

A estrutura if é usada para executar um bloco de código se uma condição for verdadeira. Se a condição for falsa, o código dentro do if é ignorado. É a forma básica de tomada de decisão em Python.

Sintaxe:



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

Exemplo:



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

ESTRUTURA 'ELSE'

A estrutura else é usada em conjunto com o if.

Ela define um bloco de código que será executado caso a condição do if seja falsa. É uma alternativa que cobre todos os outros casos não atendidos pelo if.

Sintaxe:

```
if condição:  
    # bloco de código  
  
else:  
    # bloco de código alternativo
```

Exemplo:

```
idade = 16  
if idade >= 18:  
    print("Você é maior de idade.")  
else:  
    print("Você é menor de idade.")
```

ESTRUTURA 'ELIF'

A estrutura elif (abreviação de "else if") é usada após um if para testar múltiplas condições. Se a condição do if for falsa, o Python verifica a condição do elif. Se for verdadeira, o bloco de código correspondente é executado.

Sintaxe:

```
if condição1:
    # bloco de código 1
elif condição2:
    # bloco de código 2
else:
    # bloco de código 3
```

Exemplo:

```
nota = 7
if nota >= 9:
    print("Excelente!")
elif nota >= 7:
    print("Bom!")
else:
    print("Precisa melhorar.")
```


OPERADORES COMPARATIVOS

- Usados para fazer comparações nas condições.

Operador	Descrição	Exemplo
==	Igual a	x == 5
!=	Diferente de	x != 5
>	Maior que	x > 5
<	Menor que	x < 5
>=	Maior ou igual a	x >= 5
<=	Menor ou igual a	x <= 5

ANINHAMENTO DE CONDICIONAIS

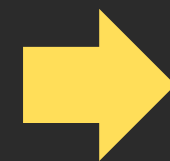


Aninhamento de condicionais em Python ocorre quando uma instrução condicional (if, elif, else) é colocada dentro de outra. Isso permite verificar múltiplas condições de forma hierárquica.

Exemplo:



```
idade = 20
if idade >= 18:
    if idade < 21:
        print("Você é um jovem adulto.")
    else:
        print("Você é um adulto.")
```



O aninhamento é útil para testar várias condições relacionadas, mas pode complicar a leitura do código se usado em excesso

USO COM OPERADORES LÓGICOS

Operadores lógicos em Python permitem combinar múltiplas condições em uma única expressão. Os principais operadores lógicos são:

- ➡ **and**: retorna True se ambas as condições forem verdadeiras.
- ➡ **or**: retorna True se pelo menos uma das condições for verdadeira.
- ➡ **not**: inverte o valor de uma condição (se for True, se torna False).

```
idade = 20
if idade >= 18 and idade < 21:
    print("Você é um jovem adulto.")
```

```
idade = 20
if idade < 18 or idade >= 65:
    print("""Você pode ter acesso a descontos
    especiais.""")
else:
    print("Você não se qualifica
    para descontos especiais.")
```

```
idade = 20
if not (idade >= 18 or idade < 65):
    print("""Você pode ter acesso a descontos
    especiais.""")
else:
    print("Você não se qualifica
    para descontos especiais.")
```


PARABÉNS!

Você concluiu a TERCEIRA aula do curso de introdução a linguagem PYTHON.

Agora um desafio:

Escreva um código que receba uma nota e imprima:

- "Aprovado" se a nota for maior ou igual a 7.
- "Recuperação" se a nota for entre 5 e 6.9.
- "Reprovado" se a nota for menor que 5.

Próxima aula: Estruturas de repetição (Loops)

