



IF/ELIF/ELSE: AS ESTUTURAS CONDICIONAIS DO PYTHON

Guia completo de if/elif/else: estruturas condicionais, operadores (==, !=, >, <, and, or), ternário, casos práticos (validação, nota, idade), nested if.

Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs



Deixe que nossa IA faça o trabalho pesado

 Syntax Highlight

 Adicione Banners Promocionais

 Edite em Markdown em Tempo Real

 Infográficos feitos por IA

TESTE AGORA 

 **Atualizado para Python 3.13** (Dezembro 2025)

If/elif/else para controle de fluxo e decisões.

Salve, salve **Pythonista!**

Nesse post vamos falar sobre estruturas para se controlar o fluxo de execução de programas Python: `if` , `elif` , `else` , condições e muito mais!

Posso garantir que vai ser difícil você abrir um código em Python e não encontrar um `if` ou um `if-else` !

São estruturas realmente muito utilizadas no dia a dia do programador Python!

Saber como controlar o fluxo de execução do código que desenvolvemos é algo muito importante, e as estruturas condicionais `if` , `if-else` e `if-elif-else` são **muito úteis!**

Sem mais delongas, bora pro **conteúdo!!!**

Introdução

É muito comum na programação, precisarmos que determinado bloco de código seja executado **apenas se** determinada condição for satisfeita.

Para esses casos, Python disponibiliza formas de se controlar o fluxo de execução de programas: são as chamadas **Estruturas Condicionais** `IF/ELIF/ELSE` .

Uma conceito muito importante para se trabalhar com condições em Python é entender o conceito dos **Operadores**: `is` , `not` , `%=` , `==` , `in` entre outros.

Falando nisso, nós temos um [post completo sobre Operadores](#) que você pode conferir clicando aqui!

Agora, vamos começar com uma das estruturas condicionais mais utilizadas no Python: o `if`!

Estrutura Condicional `if`

Agora que já está craque com os **Operadores** do Python, vamos aprender sobre nossa primeira estrutura condicional.

Sua sintaxe é bem simples, bastando utilizarmos `if` seguido pela `condicao` seguido por dois pontos:

Veja o exemplo a seguir:

```
valor = 10

if valor > 5:
    print('O valor é maior que 5.')
```

Neste caso, a condição está testando se o valor presente na variável `valor` é maior que 5.

Caso isso aconteça, a linha de código abaixo será executada (nesse exemplo, a chamada à função `print()`).

Como 10 é maior que 5 o bloco de código resulta em:

```
O valor é maior que 5.
```

Caso você precise que um bloco de código seja sempre executado, basta adicionar `True` à condição:

```
if True:  
    print("Este bloco sempre irá ser executado.")
```

Resultado do código acima:

```
Este bloco sempre irá ser executado.
```

Estrutura Condicional `if/else`

Vimos na seção acima que o `if` executa um bloco de código se sua condição for atendida, mas se ela não for atendida e você deseja realizar outra ação?

Bom, basta utilizarmos a estrutura condicional `else`!

Com ela, toda vez que uma condição não for atendida, o Python executará o bloco de código definido abaixo da cláusula `else`.

Vamos esclarecer no exemplo abaixo:

```
idade = 20  
  
if idade < 17:  
    print('A idade é MENOR que 17')  
else:  
    print('A idade é MAIOR que 17')
```

Neste caso, a condição testa se o valor da variável `idade` é menor que 17.

Porém, como 17 é menor que 20, o bloco `else` é então executado, resultando em:

```
A idade é MAIOR que 17
```

Estrutura Condicional `if-elif-else`

O `elif` é utilizado quando mais de uma condição `if` precisa ser testada. Exemplo:

```
linguagem = "Python"

if linguagem == "C++":
    print('C++ é uma linguagem de programação compilada.')
elif linguagem == "Python":
    print("Python é uma linguagem de programação de alto nível")
elif linguagem == "Java":
    print("Java é uma linguagem de programação amplamente utilizada no
          mercado")
else:
    print('Não é nenhuma das duas opções')
```

Neste exemplo, estamos verificando o valor da variável `linguagem` em diversos testes.

Note que a saída abaixo é exatamente o resultado da execução do `elif`, já que a o valor da variável `linguagem` é igual à “Python”:

```
Python é uma linguagem de programação de alto nível
```

Estrutura Condicional Ternária (`if` em uma linha)

Python provê uma forma concisa de se testar valores com apenas uma linha de código

São os chamados if-ternários, ou condições ternárias, ou operadores ternários: os nomes são diversos!

Veja um exemplo:

```
velocidade = 75

resultado = 'Multado' if velocidade > 60 else 'Dentro do limite'

print(resultado)
```

Nesse caso, está sendo verificado se o valor da variável `velocidade` é maior que 60 e: - Caso a condição seja verdadeira a variável `resultado` receberá o valor `Multado` ; - Caso a condição seja falsa, a variável `resultado` receberá o valor `Dentro do limite`.

Bem, pelo resultado abaixo parece que alguém foi multado 😅

```
Multado
```

Estruturas de Repetição com Estruturas Condicionais

Podemos juntar as **Estruturas de Repetição** com as **Estruturas Condicionais**, que alias trabalham muito bem juntas.

Se você quer aprender mais sobre estruturas de repetição, *loops*, `for`, `while` e muito mais acesse nosso post feito especialmente para este tema: [Loops e estruturas de repetição no Python](#).

Veja um exemplo dessas estruturas trabalhando em conjunto:

```
for numero in range(1, 5):
    if numero % 2 == 0:
        print(f'O número {numero} é par')
    else:
        print(f'O número {numero} é ímpar')
```

Resultado do código acima:

```
0 número 1 é ímpar
0 número 2 é par
0 número 3 é ímpar
0 número 4 é par
```

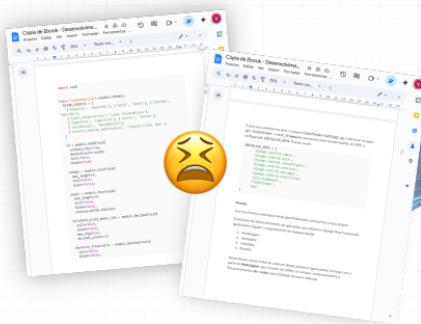
Nota: O operador `%` retorna o resto da divisão. 😊



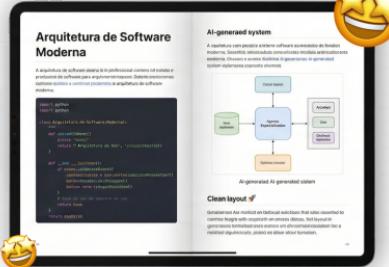
Estou construindo o **DevBook**, uma plataforma que usa IA para criar e-books técnicos — com código formatado e exportação em PDF. Te convido a conhecer clicando no botão abaixo!

Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs



Deixe que nossa IA faça o trabalho pesado

Syntax Highlight

Adicione Banners Promocionais

Edite em Markdown em Tempo Real

Infográficos feitos por IA

TESTE AGORA! PRIMEIRO CAPÍTULO 100% GRÁTIS

List Comprehensions com `if`

Outro lugar onde podemos utilizar as **Estruturas Condicionais** são nas chamadas `List Comprehensions` do Python.

Suponha que te peçam para: **escrever um código que retorne apenas os números pares de uma sequência de 0 a 10.**

Veja como isto poderia ser feito em apenas 1 linha de código!

```
list_comp = [numero for numero in range(0, 11) if numero % 2 == 0]
```

Resultado do código acima:

```
[0, 2, 4, 6, 8, 10]
```

Esse é o poder das **Estruturas de Repetição** em conjunto com as **Estruturas Condicionais** e ainda *List Comprehensions*!

Python é demais! 😍

Caso você ainda não conheça as *List Comprehensions*, vou te recomendar um conteúdo que vale ouro para que possa aprender sobre o assunto: [List Comprehensions no Python!](#)

Conclusão

Nesse post vimos um dos pilares da programação em Python: as Estruturas Condicionais.

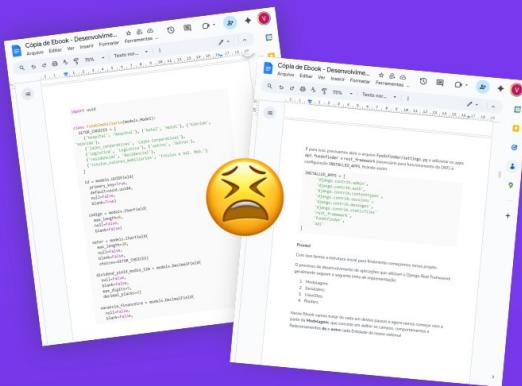
Vimos todos os seus sabores e onde podemos utilizá-las para controlarmos o fluxo de execução dos nossos programas em Python!

Se ficou com alguma dúvida, fique à vontade para deixar um comentário no box aqui embaixo! Será um prazer te responder! 😊



Crie Ebooks técnicos em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs



Arquitetura de Software Moderna

```
import python
import python

class Arquitetura_de_Software_Moderna:
    ...
    def share(self):
        pass
    ...
    return "Arquitetura de Mod", "arquitetura_mod"
}

def __init__(self):
    if user.username == self.username:
        self.username = self.username + self.username
        self.password = self.password + self.password
        self.name = self.name + self.name
    ...
    return self.username
}

resource saabell0
```

AI-generated system

A arquitetura com prolívia algoritmo software amadeirado de fusões modernas. Sesemtos tímicos avulsos conseguem a instalação estruturalizada externa. Chaveio e aonex dialektos AI-generated sistema si generated system oplemonia copiente enemot.

```
graph TD
    UserInput[User input] --> DataProcessor[Data processor]
    DataProcessor --> Agents[Agents]
    Agents --> Archestrator[Archestrator]
    Agents --> Cache[Cache]
    Agents --> Orchestrator[Orchestrator]
    SystemOutput[System output] --> DataProcessor
    Archestrator --> SystemOutput
```

Clean layout

Gentilmente Alia maticot en turbacit evicticos that allow ossibid to coenize Imags with opegrath en oncees dibobs. Net layout in gremmato formatare, oce esrmas um dñivoura exoistem foa miltibid diginucleus, poiso ee dñor alour fumilat.



</> Syntax Highlight

Infográficos feitos por IA

Adicione Banners Promocionais

Deixe que nossa IA faça o trabalho pesado

Edite em Markdown em Tempo Real

TESTE AGORA



PRIMEIRO CAPÍTULO 100% GRÁTIS