



Lógica de Programação

Introdução ao Python

Checando o Aprendizado


- O que é e para que serve o Visual Studio Code?
- O que são estruturas de controle?
- O que é e para que serve o if e o else?
- O que é e para que serve os operadores lógicos AND, OR e NOT?
- Faça um algoritmo que utilize o if e o else:



Vamos Entender Melhor o IF

A solid green horizontal bar.

E aí, futuros mestres da programação! Hoje, embarcamos em uma jornada épica para desvendar os segredos do if em Python, a chave para controlar o destino do seu código e torná-lo mais inteligente do que um mago! Preparem seus cajados de código e vamos nessa!

Abstract geometric shapes in teal, yellow, and green in the bottom-left corner.

O que é o **if**?

Imaginem o **if** como um guardião sábio que decide qual caminho o seu programa deve seguir. Ele analisa uma situação, como se fosse um enigma, e só permite que o código prossiga se a resposta for a certa. Incrível, né?

Mas como funciona essa magia?



O guardião **if** recebe uma pergunta, como "É dia?" ou "O usuário digitou um número maior que 10?". Se a resposta for "sim", ele abre as portas para o código que deve ser executado, como acender a luz do dia ou mostrar uma mensagem especial. Mas se a resposta for "não", ele tece um caminho diferente, talvez exibindo uma mensagem de boa noite ou validando o número digitado.

Guardião da Floresta



Imaginem uma floresta mágica protegida por um sábio guardião. Vocês querem entrar para colher frutas, mas ele só permite se vocês responderem "**Qual a cor da flor mais rara?**". Se responderem certo, ele abre a passagem. Se errarem, ele indica outro caminho.

Estrutura do **if**

O **if** é como um castelo com três partes:

- **A porta:** A pergunta que o guardião faz, como "Está chovendo?".
- **O salão principal:** O código que será executado se a resposta for "sim", como "Pegar o guarda-chuva".
- **A torre:** O código que será executado se a resposta for "não", como "Levar óculos de sol".

Exemplos

Verificando se está chovendo

```
está_chovendo = True

if está_chovendo:
    print("Levar o guarda-chuva!")
else:
    print("Levar óculos de sol!")
```


Exemplos

Validando a idade para entrar no cinema

```
idade = 15

if idade >= 18:
    print("Pode entrar no cinema!")
else:
    print("Desculpe, menores de 18 anos não são permitidos.")
```

Lembre-se

- O **if** precisa de uma pergunta clara e uma resposta definitiva (verdadeiro ou falso).
- Usem recuo para organizar as partes do **if** como um castelo.
- O **else** é opcional, mas é útil para indicar o que fazer quando a resposta for "não".
- Vocês podem ter vários **if** aninhados, como castelos dentro de castelos!

Dicas para Dominar o **if**:

- Comecem com perguntas simples.
- Usem comentários para explicar o que cada parte do código faz.
- Testem diferentes valores para ver como o código se comporta.
- Errem e aprendam com seus erros. Afinal, errar é humano, e programar também!

Dominando o **else**

Agora que desvendamos os segredos do **if**, o guardião sábio que controla o destino do nosso código. Agora, embarcaremos em uma nova jornada para desvendar os mistérios do **else**, seu fiel companheiro que torna o código ainda mais poderoso e versátil!



O que é o **else**?

O **else** é como um mago bondoso que surge quando o **if** decide que o código deve seguir um caminho diferente. Ele oferece uma nova oportunidade para executar instruções, mesmo que a resposta à pergunta do **if** seja "não". Incrível, né?

Como o **else** funciona?

Imaginem que o **if** é como um juiz que decide se um réu é culpado ou inocente. Se o réu for considerado culpado, o **if** abre as portas para o código que o condena. Mas se o réu for inocente, o **else** surge como um advogado de defesa e oferece uma chance de redenção, executando um código que o absolve ou o envia para um novo julgamento.

Juiz e o Advogado

Imaginem um tribunal mágico onde um juiz decide o destino de um réu acusado de bruxaria. Se o juiz o considerar culpado, o **if** condena o réu à fogueira. Mas se o juiz o considerar inocente, o **else** oferece uma chance de defesa, permitindo que o réu prove sua inocência ou seja enviado para um novo julgamento.



Estrutura do **else**

O **else** é como uma torre mágica dentro do castelo do **if**. Ele só aparece quando o guardião **if** decide que o código deve seguir um caminho diferente.

Exemplos

Mostrando mensagens diferentes para dias da semana

```
dia_da_semana = input("Digite o dia da semana: ")

if dia_da_semana == "segunda-feira":
    print("Hora de começar a semana com força!")
elif dia_da_semana == "terça-feira":
    print("Ufa, já é terça-feira!")
elif dia_da_semana == "quarta-feira":
    print("Corcunda no lombo? Quarta-feira chegou!")
elif dia_da_semana == "quinta-feira":
    print("Quase chegando no fim de semana!")
elif dia_da_semana == "sexta-feira":
    print("Sextou! Hora de relaxar!")
elif dia_da_semana == "sábado":
    print("Sábado é dia de curtir!")
else:
    print("Dia inválido. Tente novamente.")
```

Lembre-se

- O **else** é opcional, mas é muito útil para indicar o que fazer quando a resposta à pergunta do **if** for "não".
- O **else** sempre estará dentro do castelo do **if**, como uma torre dentro de um castelo.
- Vocês podem ter vários **else** aninhados, como torres dentro de torres!

Dicas para Dominar o **else**

- Pense em todas as possibilidades que podem acontecer.
- Usem o else para mostrar mensagens diferentes ou executar ações diferentes.
- Não se esqueçam de testar diferentes valores para ver como o código se comporta.
- Errem e aprendam com seus erros. Afinal, errar é humano, e programar também!

Dominando o **elif**

Já desvendamos os segredos do **if** e do **else**, os guardiões sábios que controlam o destino do nosso código. Agora, embarcaremos em uma nova jornada para desvendar os mistérios do **elif**, um mago poderoso que expande as possibilidades da nossa programação!

O que é o **elif**?

O **elif** é como um mago multifacetado que surge quando o **if** decide que o código deve seguir um caminho diferente, mas ainda existem outras possibilidades a serem exploradas. Ele permite que você crie vários "**se... senão se...**" dentro do seu código, tornando-o mais versátil e eficiente.

Mas como o **elif** funciona?

Imaginem um reino mágico onde um mago decide qual caminho um viajante deve seguir. Se o viajante responder "**sim**" a uma pergunta, ele o direciona para o caminho do sol. Se responder "**não**", o mago o direciona para o caminho da lua. Mas se o viajante responder "**talvez**", o **elif** surge e o direciona para o caminho das estrelas, revelando um novo destino.

Mago dos Caminhos



Imaginem um mago que guarda os segredos dos caminhos de um reino mágico. Se um viajante perguntar "O caminho do sol é seguro?", o mago responde "sim" e o direciona para esse caminho. Se perguntar "O caminho da lua é rápido?", o mago responde "não" e o direciona para esse caminho. Mas se perguntar "O caminho das estrelas é misterioso?", o **elif** surge e o direciona para esse caminho, revelando um novo destino.

Estrutura do **elif**

O **elif** é como uma série de torres mágicas dentro do castelo do **if**. Cada torre representa uma nova possibilidade de caminho para o código.

Exemplos

Verificando se o usuário digitou um número maior, menor ou igual a 10

```
número = int(input("Digite um número: "))

if número > 10:
    print("O número digitado é maior que 10!")
elif número == 10:
    print("O número digitado é igual a 10!")
else:
    print("O número digitado é menor que 10.")
```

Exemplos

Mostrando mensagens diferentes para diferentes faixas etárias

```
idade = int(input("Digite sua idade: "))

if idade < 18:
    print("Você é menor de idade.")
elif idade >= 18 and idade < 65:
    print("Você é um adulto.")
else:
    print("Você é um idoso.")
```

Lembre-se

- O `elif` só pode ser usado dentro do castelo do `if`, como torres dentro de um castelo.
- Você pode ter vários `elif` aninhados, como torres dentro de torres!
- O `elif` é útil quando você precisa verificar várias possibilidades diferentes.
- Use recuo para organizar as partes do `if`, `elif` e `else` como um castelo.

Dicas para Dominar o **elif**

- Pense em todas as possibilidades que podem acontecer.
- Usem o **elif** para mostrar mensagens diferentes ou executar ações diferentes.
- Não se esqueçam de testar diferentes valores para ver como o código se comporta.
- Errem e aprendam com seus erros. Afinal, errar é humano, e programar também!

break e continue no while: Dominando o Loop

- Assim como no for, os comandos **break** e **continue** podem ser usados para controlar o fluxo do loop **while**. O **break** interrompe o loop completamente, enquanto o **continue** faz com que o loop ignore a iteração atual e siga para a próxima.

break e continue no while: Dominando o Loop

- **Exemplo:** Imprimir números pares de 1 a 20, mas parar quando encontrar o número 10:

```
2  numero = 1
3
4  while numero ≤ 20:
5      if numero % 2 == 0:
6          print(numero)
7      if numero == 10:
8          break # Para o loop quando encontrar o número 10
9      numero += 1
10
```

Por hoje é só...

- Hoje vimos muita coisa, vamos descansar que amanhã tem mais!

