

# Condições e Ramificações

Tempo estimado necessário: 10 minutos

## Objetivo:

Nesta leitura, você aprenderá sobre:

1. Operadores de comparação
2. Ramificação
3. Operadores lógicos

## 1. Operações de comparação

As operações de comparação são essenciais na programação. Elas ajudam a comparar valores e a tomar decisões com base nos resultados.

### Operador de igualdade

O operador de igualdade `==` verifica se dois valores são iguais. Por exemplo, em Python:

```
age = 25
if age == 25:
    print("You are 25 years old.")
```

Aqui, o código verifica se a variável `age` é igual a 25 e imprime uma mensagem de acordo.

### Operador de desigualdade

O operador de desigualdade `!=` verifica se dois valores não são iguais:

```
if age != 30:
    print("You are not 30 years old.")
```

Aqui, o código verifica se a variável `age` é diferente de 30 e imprime uma mensagem de acordo.

### Maior que e menor que

Você também pode comparar se um valor é maior que outro.

```
if age >= 20:
    Print("Yes, the Age is greater than 20")
```

Aqui, o código verifica se a variável `age` é maior ou igual a 20 e imprime uma mensagem de acordo.

## 2. Ramificação

A ramificação é como tomar decisões em seu programa com base em condições. Pense nisso como escolhas da vida real.

### A declaração IF

Considere um cenário da vida real ao entrar em um bar. Se você tiver acima de uma certa idade, pode entrar; caso contrário, não pode.

```
age = 20
```

```
if age >= 21:
    print("You can enter the bar.")
else:
    print("Sorry, you cannot enter.")
```

Aqui, você está usando a instrução `if` para tomar uma decisão com base na variável `idade`.

### A Declaração `ELIF`

Às vezes, há várias condições a serem verificadas. Por exemplo, se você não tiver idade suficiente para o bar, pode ir ao cinema em vez disso.

```
if age >= 21:
    print("You can enter the bar.")
elif age >= 18:
    print("You can watch a movie.")
else:
    print("Sorry, you cannot do either.")
```

### Exemplo da vida real: Caixa Eletrônico (ATM)

Quando um usuário interage com um caixa eletrônico, o software no caixa pode usar ramificações para tomar decisões com base na entrada do usuário. Por exemplo, se o usuário selecionar “Sacar Dinheiro”, o caixa pode ramificar em diferentes denominações de cédulas a serem dispensadas com base no valor solicitado.

```
user_choice = "Withdraw Cash"
if user_choice == "Withdraw Cash":
    amount = input("Enter the amount to withdraw: ")
    if amount % 10 == 0:
        dispense_cash(amount)
    else:
        print("Please enter a multiple of 10.")
else:
    print("Thank you for using the ATM.")
```

## 3. Operadores lógicos

Os operadores lógicos ajudam a combinar e manipular condições.

### O operador `NOT`

#### Exemplo da vida real: Configurações de notificação

Nas configurações de notificação de um smartphone, você pode usar o operador `NOT` para controlar quando enviar notificações. Por exemplo, você pode querer receber notificações apenas quando seu telefone não estiver no modo “Não Perturbe”.

O operador `not` nega uma condição.

```
is_do_not_disturb = True
if not is_do_not_disturb:
    send_notification("New message received")
```

### O operador `AND`

#### Exemplo da vida real: Controle de acesso

Em uma instalação segura, você pode usar o operador `AND` para verificar múltiplas condições de acesso. Para abrir uma porta de alta segurança, uma pessoa pode precisar tanto de um cartão de identificação válido quanto de uma impressão digital correspondente.

O operador AND verifica se todas as condições necessárias são verdadeiras, como precisar de ambas as chaves para abrir um cofre.

```
has_valid_id_card = True
has_matching_fingerprint = True
if has_valid_id_card and has_matching_fingerprint:
    open_high_security_door()
```

O operador OR

Exemplo da vida real: Decisão sobre a noite de filme

Ao planejar uma noite de filme com amigos, você pode usar o operador OR para decidir um gênero de filme. Você escolherá um filme se pelo menos uma pessoa estiver interessada.

O operador OR verifica se pelo menos uma condição é verdadeira. É como escolher entre diferentes filmes para assistir.

```
friend1_likes_comedy = True
friend2_likes_action = False
friend3_likes_drama = False
if friend1_likes_comedy or friend2_likes_action or friend3_likes_drama:
    choose_a_movie()
```

Resumo

Nesta leitura, você explorou o operador mais utilizado e o conceito de ramificação condicional, que abrange a utilização de if statements e if-else statements.

Autor

[Akansha Yadav](#)



Changelog

Data	Versão	Alterado por	Descrição da Mudança
2023-11-03	1.0	Akansha Yadav	Criado um arquivo de leitura
2024-02-19	2.0	Gagandeep	Revisão de ID