

ASIMOV

Operadores de comparação em cadeia

Uma característica interessante do Python é a capacidade de *encadear* comparações múltiplas para realizar um teste mais complexo. Você pode usar essas comparações em cadeia como uma abreviatura para expressões booleanas maiores.

Nesta palestra, aprenderemos como encadear operadores de comparação e também apresentamos duas outras declarações importantes em python: **and** e **or**.

Vejamos alguns exemplos de uso de cadeias:

```
In [1]: 1 < 2 < 3
```

```
Out[1]: True
```

A declaração acima verifica se 1 era inferior a 2 e se 2 era inferior a 3. Poderíamos ter escrito isso usando uma instrução **and** em Python:

```
In [2]: 1<2 and 2<3
```

```
Out[2]: True
```

O **and** é usado para garantir que as duas verificações tenham que ser verdadeiras para que a verificação total seja verdadeira. Vamos ver outro exemplo:

```
In [3]: 1 < 3 > 2
```

```
Out[3]: True
```

As verificações acima checam se 3 é maior do que os outros números. Você pode usar **and** para reescrevê-lo como:

```
In [4]: 1<3 and 3>2
```

```
Out[4]: True
```

É importante notar que o Python está verificando ambas as instâncias das comparações. Nós também podemos usar **or** para escrever comparações em Python. Por exemplo:

```
In [5]: 1==2 or 2<3
```

```
Out[5]: True
```

Observe como a expressão retornou True porque com o operador **ou** precisamos apenas um

ou os outros dois sejam verdadeiros. Outro exemplo:

In [6]:

```
1==1 or 100==1
```

Out[6]: True

Ótimo! Vá em frente e vá ao questionário para esta seção para verificar sua compreensão!