Python les-materialen

# Geketende (chained) vergelijkingsoperators

Een interessant kenmerk van Python is de mogelijkheid om meerdere vergelijkingen *aaneen te koppelen* om een complexere test/condities uit te voeren. U kunt deze geketende vergelijkingen gebruiken als afkorting voor grotere Booleaanse expressies.

In deze lezing zullen we leren hoe we vergelijkingsoperatoren kunnen ketenen en we zullen ook twee andere belangrijke operatoren in Python introduceren: **and** en **or**.

Laten we eens kijken naar een paar voorbeelden van het gebruik van kettingen:

1 < 2 < 3

True

Het bovenstaande statement controleert of 1 kleiner is dan 2 **en** 2 kleiner is dan 3. We hadden dit kunnen schrijven met een **and**-statement in Python:

1<2 and 2<3

True

De **and** wordt gebruikt om ervoor te zorgen dat twee controles **True** moeten zijn om de totale controle waar te maken. Laten we nog een voorbeeld bekijken:

1 < 3 > 2

True

Het bovenstaande controleert of 3 groter is dan de andere getallen, dus je zou **and** kunnen gebruiken om het te herschrijven als:

1<3 and 3>2

True

Het is belangrijk op te merken dat Python beide instanties van de vergelijkingen controleert. We kunnen ook **or** gebruiken om vergelijkingen te schrijven in Python. Bijvoorbeeld:

1==2 or 2<3

True

Merk op hoe het **True** was; dit komt omdat we met de operator **or** alleen de ene **of** de andere nodig hebben om True te zijn. Laten we nog een voorbeeld bekijken om dit beter te begrijpen:

1==1 or 100==1

True

Geweldig! Voor een overzicht van deze korte les: U zou een goed begrip moeten hebben van het gebruik van **and** en **or** verklaringen en het lezen van geketende vergelijkingscode.