

## Conditionals


Tot nu toe hebben we eenvoudige programma's gezien, waarbij alle instructies een voor een worden uitgevoerd. Dat hoeft echter niet zo te zijn. Python-programma's kunnen in verschillende richtingen aftakken, afhankelijk van een voorwaarde of conditie. Dat doe je met het if-statement:

```
getal = int(input("Geef een getal: "))
if getal < 0:
    print("Negatief getal")
elif getal == 0:
    print("Nul")
    print("Is nul positief of negatief? Hmm...")
else:
    print("Positief getal")
```

Python deed misschien al moeilijk over syntax, maar vanaf nu moet je echt beginnen opletten! De dubbele punten zijn belangrijk, omdat ze het begin van een blok code aanduiden. Elke lijn in een blok gaat vooraf door een gelijk aantal spaties. De meerderheid van de Pythonprogrammeurs gebruikt 4 spaties om een blok code te indenteren.

Je kan meerde elif-takken hebben:

```
seizoen = input("Wat is je favoriete seizoen? ")
if seizoen == "Lente":
    print("Ontluikende bloemetjes en chocolade met Pasen!")
elif seizoen == "Zomer":
    print("Warme zon en vakantie!")
elif seizoen == "Herfst":
    print("Vallende bladeren en kastanjes!")
elif seizoen == "Winter":
    print("Sneeuwballen en eindejaarsfeesten!")
else:
    print(f"Het seizoen '{seizoen}' ken ik niet..")
```

 Wat gebeurt er als je je favoriete seizoen zonder hoofdletter ingeeft? Zoals je vast al wel ondervonden hebt nemen computers alles letterlijk op! We zullen later een paar manieren zien om dit programma gebruiksvriendelijker te maken.

Het is niet verplicht om elif of else-takken te hebben:

```
from math import sqrt

getal = int(input("Geef een getal: "))
if getal < 0:
    print("Ik kan geen vierkantswortel van negatieve getallen berekenen, sorry!")
    print("Ik zal het getal positief maken.")
    getal = -getal
print(f"De vierkantswortel van {getal} is {sqrt(getal)}")
```

 Wat gebeurt er als je het hele if-blok (regel 4 tot 7) weglaat, en je een negatief getal ingeeft? Wat is er fijner voor de gebruiker van je programma?

## Vergelijkingen en booleaanse waarden

De stukken code die gebruikt worden om een if-conditie te evalueren zijn vergelijkingen. Deze vergelijkingen hebben een waarde als uitkomst: True (waar) of False (onwaar). Deze waarden zijn geen strings of ints, maar booleans. Je kan rechtstreeks met booleans werken in Python:

```
getal = 5
is_nul = getal == 0
print(is_nul)
print(type(is_nul))
```

Let op het grote verschil in betekenis tussen de assignment-operator (=) en de gelijkheids-vergelijkingsoperator (==). Een veelvoorkomende fout is:

```
getal = 5
if getal = 0:
    print("Dit getal is nul")
```

Python valt hierover omdat je waarschijnlijk `==` bedoelt, en niet `=`. In sommige andere programmertalen is dit toegelaten, wat tot verwarring en fouten kan leiden!

## Booleaanse operatoren

Je kan rekenen met booleaanse waardes! Dat is minder vreemd dan je zou denken:

TODO: boolean examples