

# Online Proefstuderen

## HBO-ICT

### Theorie – module Programmeren

#### Condities

Een van de eerste concepten die we nodig hebben bij het programmeren is het **conditioneel statement**, dat ons in staat stelt om onze computer te vertellen dat het een set instructies moet volgen als aan een bepaalde voorwaarde is voldaan. Elke keer dat we controleren of aan een voorwaarde wordt voldaan wordt een **test** genoemd. Deze tests slagen als aan de voorwaarde wordt voldaan (waar, of **True** zijn), en falen als de uitkomst van de test **False** is.



Een keus maken in Python kan heel eenvoudig zijn...

```
if [conditie]:  
    [doe iets]
```

#### Drie mogelijkheden

Er zijn drie manieren om een conditionele statement te schrijven. **if** is verplicht en kan gecombineerd worden met **elif** en/of **else**.

##### If

Als aan de conditie is voldaan dan worden de instructies uitgevoerd die ingesprongen onder **if** staan. **if** is natuurlijk Engels voor “als”, en zo zal je meer termen in het Engels gaan tegenkomen. En let op, het gebruik van **if** is verplicht, een **elif** of **else** zonder **if** is niet geldig!

```
if cijfer > 5.5:  
    print("Voldoende")
```

##### elif

**elif** is een afkorting van “else if”, of “anders als”. Dit geeft aan dat in het geval de test boven deze faalt (de **if**, in dit geval), een nieuwe conditie zal moeten worden gecontroleerd op dezelfde manier als het **if** statement. Als aan de voorwaarde erboven (**if**) al is voldaan, dan zal de **elif** niet worden uitgevoerd.

```
if cijfer > 7.5:  
    print("Goed")  
elif cijfer > 5.5:  
    print("Voldoende")
```

## Else

**else** is de laatste optie: als alle voorgaande tests mislukken, voer dan de volgende instructie(s) uit. Let op: Python is zeer gevoelig voor de manier waarop je jouw code schrijft, zowel de *syntaxis* (hoe je de woorden van de taal samenvoegt tot samenhangende instructies) als de inspringing (hoe ver de tekst van de linkerkant van het scherm af staat) zijn belangrijk.

```
if cijfer > 7.5:
    print("Goed")
elif cijfer > 5.5:
    print("Voldoende")
else:
    print("Onvoldoende")
```

Voor conditionele statements moet je ervoor zorgen dat de instructies die bij elke test horen, daar direct onder staan en vier spaties onder het **if**, **elif** of **anders** statement ingesprongen staan.

We zullen het nu hebben over waarom je misschien conditionele statements wilt gebruiken, en je zult zien hoe je ze schrijft. Vervolgens nemen we enkele voorbeelden door en leggen we uit waarom ze werken. Na deze lezing zal je voorbereid zijn om het steen, papier en schaar lezing te volgen en aan het huiswerk te beginnen.

## Waar of niet waar

True of False

Een conditioneel statement maakt het mogelijk dat onze code verschillende dingen kan doen, afhankelijk van de vraag of een conditie waar of niet waar was. Dit is waarschijnlijk de meest nuttige uitdrukking die we kunnen schrijven. Als we geen condities hadden dan zou onze code elke keer weer hetzelfde doen. Dat zou een niet heel erg nuttig programma zijn.

Stel je voor dat jouw e-mailprovider je altijd laat inloggen, ongeacht het wachtwoord dat je gebruikt? Conditionele statements stellen ons programma in staat om het juiste antwoord te geven, zelfs als de invoer verandert. In dit geval is het *conditionele statement*: is het ingevoerde wachtwoord gelijk aan het eerder bewaarde wachtwoord van een gebruiker?

```
temp = 22.0

if temp > 35.0:
    print("Heet!")
```

Conditionele statements stellen ons programma in staat om het juiste antwoord te geven, zelfs als de invoer verandert. Stel je bijvoorbeeld voor dat we proberen te bepalen of het buiten heet is. Als het warmer is dan 35 graden Celsius zullen we zeggen dat het heet is.

Een **if**-statement is de juiste keuze voor deze code omdat we alleen willen zeggen dat het heet is als de temperatuur hoger is dan 35 graden Celsius. Op dit moment hebben we de temperatuur gedefinieerd op 22. We weten dat 22 niet groter is dan 35, dus deze test zal mislukken. Wanneer een test faalt, dan zal zijn *body* niet worden uitgevoerd.

Zie [www.hanze.nl/onlineproefstuderen](http://www.hanze.nl/onlineproefstuderen) voor alle modules!

**share your talent. move the world.**

Dus deze code zal *niets* afdrukken. Onthoud, dat is wat we willen! Merk op dat de body altijd één niveau dieper ingesprongen is dan de test en dat er een dubbele punt is geschreven na de test.

```
temp = 36.0

if temp > 35.0:
    print("Heet!")
```

Heet!

Als we de temperatuur veranderen naar 36 en dan de code uitvoeren, dan wordt er 'Heet!' geprint. Wat als we ook andere temperaturen willen classificeren? Om dit te doen, moeten we twee nieuwe conditionele statements introduceren, **elif** en **else**. Laten we beginnen met **elif**.

```
temp = 36.0

if temp > 35.0:
    print("Heet!")
elif temp > 20.0:
    print("Warm")
```

Heet!

**elif** is een combinatie van de woorden "else" en "if". Het betekent dat als de vorige test is mislukt, doe dan deze test. Om te zien hoe het werkt, laten we een voorbeeld proberen. Hier is de temperatuur nog steeds 36. Net als eerder, de eerste **if** slaagt, dus de body wordt uitgevoerd en het programma drukt 'Heet!' af.

Wordt het *tweede* blok uitgevoerd? Bedenk, **elif** betekent als de eerste test mislukt, voer deze test uit. Sinds de eerste test is geslaagd, voeren we de **elif** niet meer uit, en er wordt niets meer geprint.

```
temp = 23.0

if temp > 35.0:
    print("Heet!")
elif temp > 20.0:
    print("Warm")
```

Warm

Wat gebeurt er als we de temperatuur op 23 zetten? Dan faalt de eerste test, zodat de body niet wordt uitgevoerd. Sinds de eerste test faalde, doen we vervolgens de **elif** test. We zien dat deze test slaagt, dus deze body wordt uitgevoerd en het programma drukt 'Warm' af.

```
temp = 23.0

if temp > 35.0:
    print("Heet!")
elif temp > 20.0:
    print("Warm")
elif temp > 10.0:
    print("Koel")
else:
    print("Brrr!")
```

Warm

Laten we dit programma nu voltooien met een volledig temperatuurbereik. We hebben een classificatie van 'Koel' toegevoegd voor temperaturen tussen 10 graden en 20 graden. We hebben ook een nieuwe conditie toegevoegd, het `else`-statement. Het `else`-statement betekent dat als geen van de `if`'s of `elif`'s geslaagd is, de code in het `else`-statement wordt uitgevoerd.

Hier betekent dat als de temperatuur niet hoger was dan 35, 20 of 10, we 'Brrr!' afdrucken. Bijvoorbeeld, de temperatuur is nu 23. Dus wat drukt onze code af? De eerste test faalt, maar we slagen voor de tweede test. Dat betekent dat de body van de tweede test wordt uitgevoerd, maar geen van de andere `elif`'s of het `else`-statement. Onze code drukt 'Warm' af en dat is wat we willen.

```
temp = 0.0

if temp > 35.0:
    print("Heet!")
elif temp > 20.0:
    print("Warm")
elif temp > 10.0:
    print("Koel")
else:
    print("Brrr!")
```

Brrr!

Wat gebeurt er als de temperatuur 0 is? Dan slagen we niet voor alle `if`- en `elif`-statements, dus geen van de bodies zal worden uitgevoerd. Aangezien alle tests mislukt zijn, wordt het `else`-statement uitgevoerd. Het heeft geen test en in plaats daarvan drukt het 'Brrr' af.

## Mogelijke fouten

We laten nu een paar dingen zien die fout kunnen gaan bij het schrijven van het temperatuurprogramma.

### Volgorde

De code die je hier ziet is een soortgelijk programma dat we eerder hebben geschreven. Maar wat gebeurt er als je de temperatuur op negatief 10 zet? Het eerder geschreven programma zal 'Brrr!' afdrucken. Echter, het programma dat je hier ziet zal 'Heet!' afdrucken! Merk op dat het niet werkt voor negatieve getallen. Maar ook een temperatuur van bijvoorbeeld 15 zal niet het gewenste resultaat geven, en zal 'Brrr!' afdrucken.

Zie [www.hanze.nl/onlineproefstuderen](http://www.hanze.nl/onlineproefstuderen) voor alle modules!

```
temp = -10.0

if temp > 0.0:
    print("Brrr!")
elif temp > 10.0:
    print("Koel")
elif temp > 20.0:
    print("Warm")
else:
    print("Heet!")
```

Heet!

### Uitsluiting

Laten we eens kijken naar een andere manier om dit programma te schrijven die misschien ook niet werkt zoals we zouden willen. In plaats van een heleboel `elif`-statements te schrijven, hadden we drie `if`-statements kunnen schrijven. Maar wat gebeurt er als de temperatuur 40 graden is?

In dit geval zullen alle drie de tests slagen en drie verschillende statements over de temperatuur afdrukken. Dat gebeurt omdat de `if`-statements elkaar niet uitsluiten. Als één slaagt, dan wordt de rest nog steeds getest.

```
temp = 40.0

if temp > 35.0:
    print("Heet!")

if temp > 20.0:
    print("Warm")

if temp > 10.0:
    print("Koel")
```

Heet!  
Warm  
Koel

Dat was veel informatie, dus laten we de belangrijke concepten even kort samenvatten.

## Conditionele statements

### Blokken

Een `if`-statement laat een programma verschillende code uitvoeren, afhankelijk van het feit of een test slaagt of niet. Het typische `if`-statement heeft één of meer blokken, en elke `if` of `elif` heeft een test. Hier tel je 4 blokken (één `if`, twee `elif`'s en een `else`).

```
if temp > 35.0:
    print("Heet!")
elif temp > 20.0:
    print("Warm")
elif temp > 10.0:
    print("Koel")
else:
    print("Brrr!")
```

### Handelen

Tests worden achtereenvolgens uitgevoerd totdat één slaagt. De code die bij deze test hoort wordt uitgevoerd, en er worden verder *geen* andere tests meer uitgevoerd. Als geen enkele test slaagt, dan wordt de optionele `else` uitgevoerd. Vergeet niet dat `if` tests altijd worden uitgevoerd, terwijl `elif` en `else` tests alleen worden uitgevoerd als de vorige tests zijn mislukt.

```
temp = -10.0

if temp > 35.0:
    print("Heet!")
elif temp > 20.0:
    print("Warm")
elif temp > 10.0:
    print("Koel")
else:
    print("Brrr!")
```

### Geneste blokken

Blokken kunnen worden genest en we gaan een klein experiment doen om dit te laten zien: is `elif` echt nodig?



```
temp = -10.0

if temp > 35.0:
    print("Heet!")
elif temp > 20.0:
    print("Warm")
elif temp > 10.0:
    print("Koel")
else:
    print("Brrr!")
```

```
Brrr!
```

Dit is het voorbeeld dat we al eerder hebben gezien. Wat nu als `elif` niet zou bestaan, kunnen we dit ook met alleen `if` en `else` schrijven? Laten we beginnen met de eerste conditie!

```
temp = -10.0

if temp > 35.0:
    print("Heet!")
else:
    if temp > 20.0:
        print("Warm")
    elif temp > 10.0:
        print("Koel")
    else:
        print("Brrr!")
```

Brrr!

Als de eerste conditie (`temp > 35.0`) faalt, doe dan het volgende (`else`). Dit principe gaan we herhalen voor de volgende condities, tot er geen `elif` meer over is. En hier zie je ook een eerste genest blok!

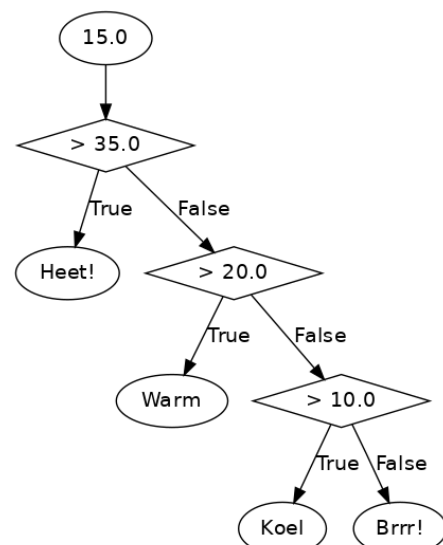
```
temp = 15.0

if temp > 35.0:
    print("Heet!")
else:
    if temp > 20.0:
        print("Warm")
    else:
        if temp > 10.0:
            print("Koel")
        else:
            print("Brrr!")
```

Koel

Het blijkt dat we heel goed zonder `elif` kunnen! En je ziet ook dat we langzaam de diepte ingaan, een vorm die je misschien herkent als een beslissingsboom.

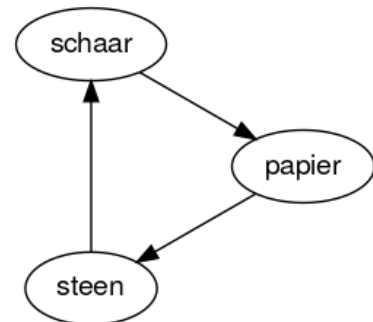
Waarom tóch `elif` gebruiken? `elif` is een syntactisch hulpmiddel dat het voor ons gemakkelijker maakt: met heel veel keuzes zou onze code door alle inspringen van de pagina gaan lopen!



Zie [www.hanze.nl/onlineproefstuderen](http://www.hanze.nl/onlineproefstuderen) voor alle modules!

## Steen, papier en schaar

Een voorbeeld waar geneste conditionele statements kunnen worden gebruikt is bij het steen, papier en schaar spel (RPS, of *rock, paper and scissors*).



```
comp = "steen"
user = "papier"

if comp == "papier" and user == "papier":
    print("Gelijkspel. Nog een keer?")
elif comp == "steen":
    if user == "schaar":
        print("Ik win! *_*")
    else:
        print("Jij wint. Aargh!")
```

We gaan dit spel tegen de computer spelen, waar de variabele `user` jouw keus is en de variabele `comp` de keus van de computer. De vraag is nu of dit programma correct is...

1. Zal dit programma de juiste uitkomst printen?
2. Zal het **altijd** de juiste uitkomst printen, ook als `user` en `comp` andere waarden krijgen?

Let op, je ziet hier een voorbeeld hoe je meerdere condities kan combineren door middel van `and`: de test slaagt alleen als zowel de eerste (`comp == "papier"`) als de tweede test (`user == "papier"`) correct is!