
Inhoudsopgave

Voorwoord	5
Voordat je begint...	6
Wat heb je nodig?	7
De website bij het boek	7
Voor ouders, verzorgers en leraren	8
Vervolgboeken over programmeren	8
1. Aan de slag met Python	9
1.1 Wat is programmeren?	10
1.2 Welkom bij <i>Python</i>	11
1.3 Een commando typen	12
1.4 Een fout maken	15
1.5 Het <i>Codevenster</i>	15
1.6 Een programma maken	16
1.7 Je programma opslaan	17
1.8 Je programma draaien	18
1.9 <i>Python</i> afsluiten	19
1.10 Een programma openen	20
1.11 Oefeningen	22
2. Goochelen met variabelen	23
2.1 Rekenen in <i>Python</i>	24
2.2 Het <i>Shell</i> -venster opruimen	26
2.3 Wat is een variabele?	27
2.4 Een numerieke variabele maken	27
2.5 Kilometers omrekenen met numerieke variabelen	30
2.6 Input vragen	33
2.7 Tekst en getallen samen afbeelden	37
2.8 Tekstvariabelen	39
2.9 Commando's kopiëren	40
2.10 Een programma maken dat letters telt	41
2.11 Oefeningen	46
3. Beslissingen nemen	47
3.1 Waar of niet waar	48
3.2 Beslissen met het <i>if</i> -commando	50
3.3 Het <i>else</i> -commando gebruiken	53
3.4 Modules gebruiken	56
3.5 Tussendoor testen	57
3.6 Raadgetal afmaken	59
3.7 Het programma stoppen	60
3.8 Het <i>elif</i> -commando gebruiken	61
3.9 Opnieuw de commando's gebruiken	65
3.10 Oefeningen	72

4. Lussen maken	73
4.1 Een <i>for</i> -lus maken	74
4.2 Een <i>while</i> -lus maken	85
4.3 Oefeningen	90
5. Codebreker maken	91
5.1 <i>Codebreker</i>	92
5.2 Input maken en controleren	93
5.3 Commentaar leveren	95
5.4 Controleren op cijfers	96
5.5 Meerdere ifs tegelijk	104
5.6 Controleren met een geneste lus	110
5.7 De inputcontrole afronden	115
5.8 De geheime code maken	116
5.9 De codes vergelijken	117
5.10 Het programma in elkaar zetten	118
5.11 Oefeningen	122
6. Alles op een rijtje met lijsten	123
6.1 Een lijst maken	124
6.2 Een woordenboek maken	129
6.3 Je woordenboek veranderen	132
6.4 Oefeningen	134
7. Teken en functies maken	135
7.1 Teken met een <i>turtle</i>	136
7.2 Maak het jezelf makkelijk met je eigen functies	140
7.3 Oefeningen	146
Bijlagen	
A. Python downloaden en installeren	147
B. Woordenlijst en commando's	153
C. Index	157

Voorwoord

Ik hoop dat je er door dit boek achter komt hoe leuk programmeren is. Het leuke aan programmeren vind ik dat je creatief bezig bent, het is voor mij echt een hobby. Wat je bedenkt in je fantasie kan je in beeld en geluid, spellen en handige programma's omzetten. Hoe leuk is dat? En bovendien ook best spannend. Zal het programma doen wat jij wilt? Als het je lukt, voelt dat als een overwinning. Het zorgt ervoor dat je nog grotere en betere programma's wilt maken.

En wie weet: als je genoeg leert over programmeren, kun je van je hobby je werk maken. Net als ik ooit gedaan heb. In de laatste klas van de middelbare school moest ik kiezen welke studie ik wilde doen? In die tijd had iets mijn aandacht getrokken: computers. Nu zijn computers overal, toen waren het voor de meeste mensen nog mysterieuze apparaten. Dat veranderde in de jaren erna.

Voor het eerst kon je een computer in huis halen en hem zelf programmeren. Je kon het apparaat laten doen wat je wilde. Geluid afspelen, poppetjes laten bewegen, spellen spelen. Het beeld was blokkerig en het geluid erg matig. Maar voor die tijd was het magisch!

Het voelde als iets dat erg groot en belangrijk zou worden in de toekomst en dat werd het. Van mijn vader kreeg ik een Commodore 64-computer en ik kocht een dik programmeerboek om te leren programmeren. Al snel daarna begon ik ook aan een beroepsopleiding voor programmeren. Een paar jaar later haalde ik mijn diploma en binnen een paar maanden werkte ik als programmeur bij een chocoladefabriek.

Na jaren met plezier programmeren voor mijn werk en voor de lol, kwam ik uiteindelijk terecht in iets anders dat ik erg leuk vond: schrijven. Nu schrijf ik alweer heel wat jaren over computers en alles wat daarmee te maken heeft. Met dit boek over programmeren ben ik eigenlijk weer terug waar ik zelf ooit begon.

Ik wens je veel succes en plezier met leren programmeren in *Python*!



Alex Wit
Studio Visual Steps

Voordat je begint...

Voordat je begint, lees je eerst deze bladzijden. Daarna weet je precies hoe je dit boek gebruikt.

In dit boek zie je verschillende plaatjes:



Bij de muis staat altijd een opdracht die je moet uitvoeren. Bijvoorbeeld ergens op klikken, rechtsklikken of dubbelklikken.



Bij het toetsenbord staat een opdracht om iets te typen.



Bij het handje staat een opdracht die je moet uitvoeren. Bijvoorbeeld een computerprogramma openen.



Bij de pijl moet je ergens speciaal op letten. Lees daarom altijd wat erbij staat.



Bij de pleister vind je hulp bij bepaalde problemen.



Bij de lamp vind je een handige tip over het werken met de computer.



Bij het boek krijg je extra informatie. Die kun je op je gemak doorlezen.

Dit boek is een doe-boek. Als je begint met het eerste hoofdstuk leg je het boek naast je toetsenbord neer.

Daarna ga je alle opdrachten die je leest direct uitvoeren op je eigen computer. Zo leer je het snelste werken met het programma waarin je leert programmeren. Gewoon door het te doen!

In dit boek kom je deze opdrachten tegen die je met de computermuis uitvoert:

Klikken = 1 keer kort op de linkermuisknop drukken.

Dubbelklikken = 2 keer heel snel na elkaar op de linkermuisknop drukken.

Rechtsklikken = 1 keer kort op de rechtermuisknop drukken.

3. Beslissingen nemen



Heel belangrijk bij het maken van programma's is het nemen van beslissingen. Daarmee bepaal je wat een programma moet doen. Dat werkt hetzelfde als wanneer je zelf een beslissing neemt. Bijvoorbeeld: je kijkt of het buiten koud is. Ja? Dan trek je een dikke jas aan.

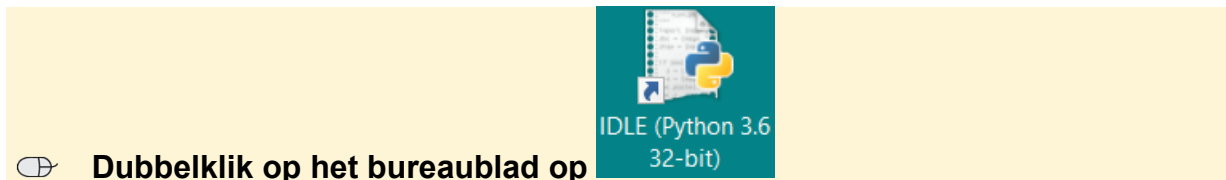
Voor het nemen van beslissingen in *Python* gebruik je het *if*-commando. Dat beslist wat het programma moet doen. Zo'n beslissing wordt altijd genomen aan de hand van een *vergelijking* of *voorwaarde*. Dat kan bijvoorbeeld een vergelijking tussen twee getallen zijn. Als het eerste getal groter is dan het tweede, wordt een bepaald deel van het programma uitgevoerd. Je ziet in dit hoofdstuk precies hoe dat werkt.

In dit hoofdstuk leer je:

- wat *true* en *false* is;
- beslissen met het *if*-commando;
- vergelijken voor de juiste beslissing;
- waarom je tekst laat inspringen;
- wat je met *else* doet;
- wat modules zijn;
- waarom je tussendoor test;
- hoe je een programma stopt;
- wat *elif* is.

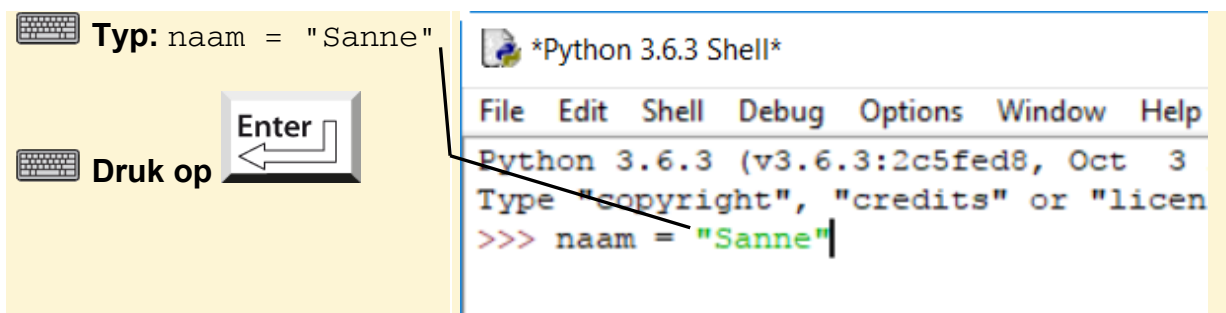
3.1 Waar of niet waar

Python is behalve een programmeertaal ook een soort leugendetector. Als *Python* iets ziet dat niet klopt, zegt hij dat duidelijk. Kijk maar eens. Je start eerst *Python* weer:

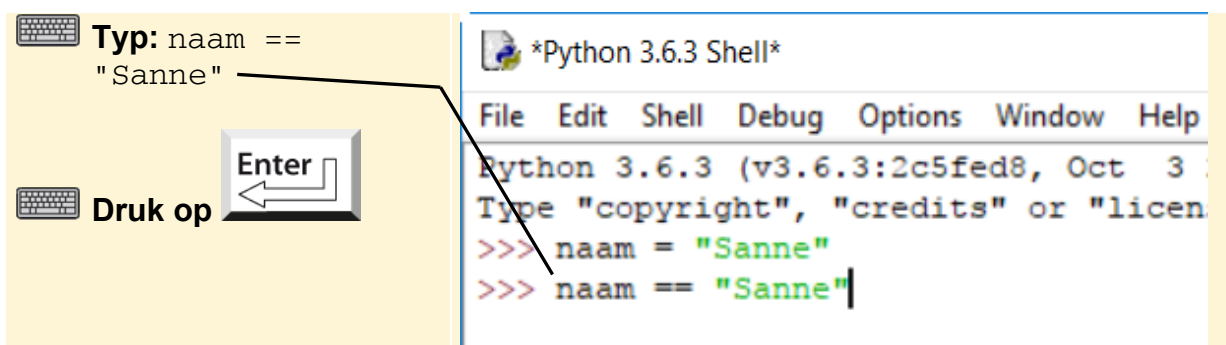


 **Dubbelklik op het bureaublad op**

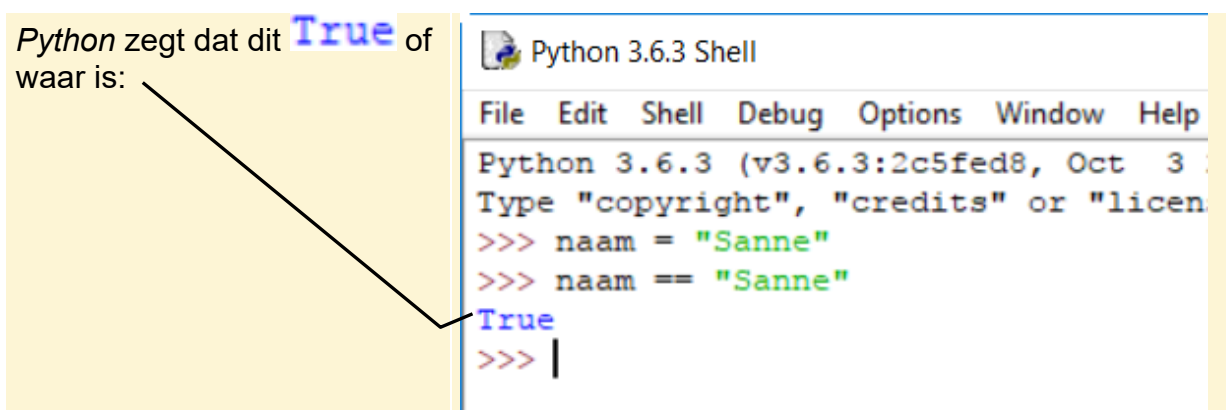
Het *Shell*-venster is open. Je vult de tekstvariabele *naam* met de naam *Sanne*. Daarvoor zet je aanhalingstekens om de naam:



Je vraagt nu aan *Python* of de naam *Sanne* is. Daarvoor gebruik je: `==`. Dit betekent *is gelijk aan*. Er wordt `==` gebruikt zodat je niet in de war raakt met `=`. Met een enkele `=` wijs je een getal of tekst toe aan een variabele, zoals je bij de vorige stap deed.



Python zegt dat dit **True** of waar is:



Typ: naam == "Maan"

Druk op

```

Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3
Type "copyright", "credits" or "licen
>>> naam = "Sanne"
>>> naam == "Sanne"
True
>>> naam == "Maan"

```

Python zegt dat dit **False** of niet waar is:

```

Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3
Type "copyright", "credits" or "licen
>>> naam = "Sanne"
>>> naam == "Sanne"
True
>>> naam == "Maan"
False
>>> |

```

Wat is hier gebeurd? Je hebt eerst *Sanne* in de variabele *naam* gezet. Daarna vroeg je aan *Python* of de inhoud van *naam* hetzelfde was als *Sanne*. Dat was waar. Daarna vroeg je of de inhoud van *naam* hetzelfde was als *Maan*. En dat was niet waar.

Op dezelfde manier vergelijk je getallen met elkaar. Bijvoorbeeld als je wilt testen of een getal hoger is dan een ander getal. Daarvoor gebruik je `>`:

Typ: getal = 34

Druk op

Typ: getal > 33

Druk op

```

Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017,
Type "copyright", "credits" or "license()"
>>> naam = "Sanne"
>>> naam == "Sanne"
True
>>> naam == "Maan"
False
>>> getal = 34
>>> getal > 33

```

Python zegt dat dit **True** of waar is:

En dat klopt natuurlijk, want 34 is groter dan 33.

```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017,
Type "copyright", "credits" or "license()")
>>> naam = "Sanne"
>>> naam == "Sanne"
True
>>> naam == "Maan"
False
>>> getal = 34
>>> getal > 33
True
>>> |
```

De tekens == en > worden *vergelijkingsoperators* genoemd. In *Python* heb je de volgende vergelijkensooperators:

- == is gelijk aan
- != is ongelijk aan
- < is kleiner dan
- > is groter dan
- <= is kleiner dan of gelijk aan
- >= is groter dan of gelijk aan

Op deze manier heb je natuurlijk nog niet veel aan vergelijken. Je weet zelf ook wel dat een naam die je net getypt hebt, hetzelfde is als de naam waarmee je daarna vergelijkt. Maar dit laat wel even goed zien hoe het vergelijken in *Python* werkt.

Door getallen en teksten te vergelijken, beslis je hoe je programma verder moet gaan. En daarmee kun je je programma een heel stuk leuker maken.

3.2 Beslissen met het if-commando

Met het *if*-commando maak je beslissingen. *If* is Engels voor 'als'. Je gebruikt daarvoor de vergelijkensooperators. Als de vergelijking bij het *if*-commando waar is, worden de commando's daarbij uitgevoerd.

Bekijk dat eens in een simpel voorbeeld. Je maakt een programma in het *Codevenster* dat lijkt op wat je geoefend hebt in de vorige paragraaf:

