

Kennismaking met IDLE

Met IDLE kun je in Python programma's schrijven en uitvoeren. Kijk hoe dit gaat door een eenvoudig programma te maken dat een bericht op het scherm schrijft.

Werken in IDLE

Volg deze stappen om IDLE in een Python-programma te gebruiken. Je leert ermee programma's te beginnen, op te slaan en uit te voeren.

1 IDLE starten

Start IDLE op via de instructies voor jouw computersysteem (zie p. 88-91). Het Shell-venster verschijnt en toont de programma-output (elke informatie die het programma produceert) en eventuele foutmeldingen.

Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 16 2013, 23:39:35)
 [GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
 Type "copyright", "credits" or "license()" for more information.
 >>>

Hier verschijnen berichten van Python

Wat er verschijnt, hangt af van je besturingssysteem

2 Open een nieuw venster

Klik op het menu 'File' boven in het Shell-venster en kies 'New File'. Hiermee open je het Code-venster.

Dit is het Shell-venster

Klik hier om het Code-venster te openen

ZIE OOK

(88-91) Python installeren

Welk 106-107 venster?

TIPS VAN EXPERTS

Verschillende vensters

Python kent twee verschillende vensters, het 'Shell-venster' en het 'Code-venster' (zie p. 106-107). We hebben ze voor het onderscheid een eigen kleur gegeven.

Shell-venster

Code-venster

3 Voer de code in

Typ de volgende tekst in in het nieuwe Code-venster. Het is een instructie om de woorden 'Hello World' te schrijven.

```
print('Hello World!')
```

Gebruik enkele aanhalingstekens

4 Sla het Code-venster op

Klik op het menu 'File' en kies 'Save as'. Voer de bestandsnaam 'Hello World' in en klik op 'save'.



Klik hier om het bestand op te slaan

5 Voer het programma uit

Klik in het Code-venster op 'Run' en kies 'Run module'. Hiermee zal het programma worden uitgevoerd in het Shell-venster.

Klik hier om het programma uit te voeren

6 Output in het Shell-venster

Kijk naar het Shell-venster. Als het programma wordt uitgevoerd, moet het bericht 'Hello World' verschijnen. Je hebt nu je eerste coderingsbit in Python gemaakt!

```
>>>
Hello World!
>>>
```

Het bericht zal verschijnen zonder aanhalingstekens

ONTHOUD

Hoe IDLE werkt

Volg altijd deze drie stappen in IDLE: schrijf de codering, sla hem op en voer hem uit. Onthoud: een codering die niet is opgeslagen, wordt ook niet uitgevoerd. Er verschijnt een waarschuwing als je dit probeert.

Tekst invoeren → Opslaan → Uitvoeren

Fouten

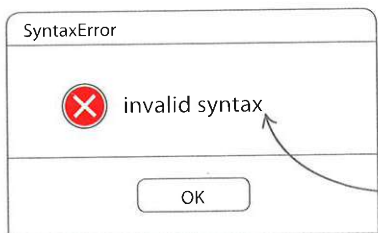
Soms werken programma's niet meteen, maar ze kunnen altijd worden gerepareerd. Als de codering voor een programma niet correct is ingevoerd, zal Python een foutmelding geven waarin staat wat er mis is gegaan.

Fouten in het Code-venster

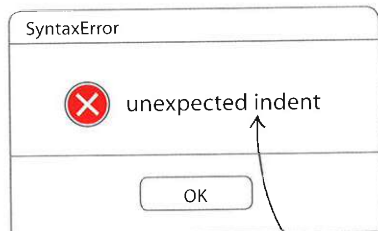
Als je probeert een programma in het Code-venster uit te voeren, kan er een pop-upvenster verschijnen met een foutmelding, zoals 'SyntaxError'. Deze fouten onderbreken het programma en moeten worden hersteld.

1 Syntaxfout

Als er een pop-upvenster verschijnt met de melding 'SyntaxError', betekent dat vaak dat er een spel- of typfout in de codering zit.



Deze codering bevat een typfout

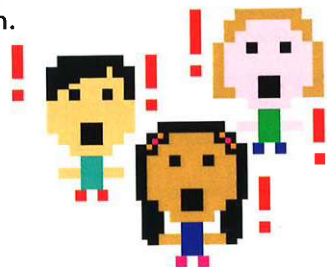


Deze codering bevat een onjuiste spatiering, die het uitvoeren van het programma tegenhoudt

ZIE OOK

Fouten en foutopsporing 148-149

Wat nu? 176-177



2 Fout gemarkeerd

Klik op 'OK' in het pop-upvenster en keer terug naar het programma. De fout of de plek daaromheen zal rood gemarkeerd zijn. Bestudeer deze regel zorgvuldig.



TIPS VAN EXPERTS

Klassieke fouten

Sommige fouten worden vaak gemaakt. Let daarom extra op deze gewone problemen:

Hoofdletters/kleine letters: de code moet precies overeenkomen. Schrijf je 'Print' in plaats van 'print', dan zal Python de instructie niet begrijpen.

Enkele en dubbele aanhalingstekens: gebruik deze niet door elkaar. Na elk aanhalingsteken openen moet er ook een volgen om te sluiten.

Minteken en underscore: verwar het minteken (-) niet met de underscore (_).

Verschillende haken: alle soorten haken, bijvoorbeeld {}, [] en (), hebben hun eigen toepassing. Gebruik dus altijd de juiste en controleer of je ze in paren hebt gebruikt.

Fouten in het Shell-venster

Soms verschijnt er een foutmelding in rode letters in het Shell-venster. Hierdoor zal het programma niet verder kunnen.

1 Naamfout

Als het bericht 'NameError' verschijnt, begrijpt Python een van de gebruikte woorden niet. Als de coderingsfout in het codevenster wordt weergegeven, klik dan met rechts op het bericht in het Shell-venster en kies 'Go to file/line'.



De regel in de codering (in het Code-venster) waarin de fout is aangetroffen

>>>

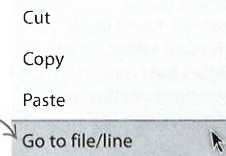
Traceback (most recent call last):

```
File "C:\PythonCode\errors.py", line 1, in <module>
    print('Hello World!')
```

NameError: name 'print' is not defined

Het woord dat Python niet begrijpt

Klik hier om de regel te markeren waarin de fout uit het Code-venster verschijnt



2 Herstel de fout

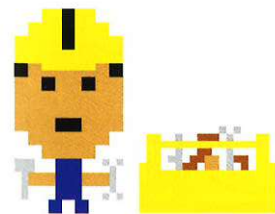
De regel met de fouten is gemarkeerd in het Code-venster. Er is 'print' getypt in plaats van 'print'. Je kunt de code nu bewerken om de fout te herstellen.

```
print('Hello World!')
```

Verander dit om 'print' te lezen

Fouten opsporen

Met de tips op deze twee pagina's kun je de coderingsregel waarin fouten staan opsporen en vervolgens dubbelklikken op die regel. Loop door de checklist rechts om te ontdekken wat er fout is gegaan.



Wanneer er iets fout gaat

Er bestaan enkele methodes waarmee je fouten makkelijker kunt opsporen. Rechts een handige checklist.

FOUTEN VOORKOMEN	
Controleer je codering op de volgende punten	✓
Heb je precies gekopieerd wat je moest intypen?	✓
Heb je alles correct gespeld?	✓
Staan er twee aanhalingstekens (') om de uitdrukking die je wilt printen?	✓
Staan er extra spaties aan het begin van de regel? Spaties zijn erg belangrijk in Python.	✓
Heb je de regels onder en boven de gemarkeerde regel gecontroleerd? Soms ligt het probleem daar.	✓
Heb je iemand anders gevraagd je codering te controleren? Mogelijk zien zij iets wat jij hebt gemist.	✓
Gebruik je Python 3 en niet Python 2? Programma's voor Python 3 werken niet altijd in Python 2.	✓

PROJECT 4

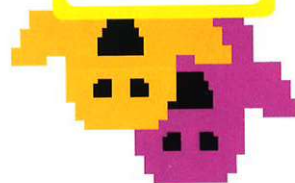
Spokenspel

Dit spel belicht enkele dingen waarop je moet letten als je programma's schrijft in Python. Als je de codering hebt ingevoerd, start dan het programma om het spel te spelen. Kun jij ontsnappen uit het spookhuis?

ZIE OOK

Spokenspel 98-99 >
gedecodeerd

Programmastroom
100-101 >



1 Start IDLE en open een nieuw venster via 'File'. Sla de game op als 'ghost game' (Spokenspel). Plaats beide vensters naast elkaar, zodat je ze allebei kunt zien en typ onderstaande codering in het codevenster.

```
# Ghost Game
from random import randint
print('Ghost Game')
feeling_brave = True
score = 0
while feeling_brave:
    ghost_door = randint(1, 3)
    print('Three doors ahead...')
    print('A ghost behind one.')
    print('Which door do you open?')
    door = input('1, 2, or 3?')
    door_num = int(door)
    if door_num == ghost_door:
        print('GHOST!')
        feeling_brave = False
    else:
        print('No ghost!')
        print('You enter the next room.')
        score = score + 1
    print('Run away!')
    print('Game over! You scored', score)
```

Dit moeten underscores zijn, geen mintekens

Deze sectie moet worden voorafgegaan door vier spaties. Gebeurt dit niet automatisch, check dan of er een dubbele punt staat na 'feeling_brave'

Deze inspringing bedraagt automatisch acht spaties en moet worden teruggebracht tot slechts vier

Verwijder hier alle spaties

'score' mag hier niet tussen aanhalingstekens staan

Gebruik enkele aanhalingstekens

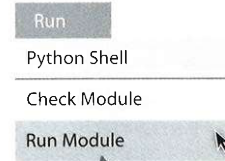
Gebruik alleen hoofdletters waar die worden weergegeven

Let erop dat je hier een dubbele punt zet

Gebruik hier twee ==-tekens



2 Als je de codering zorgvuldig hebt ingevoerd, selecteer dan 'Run Module' in het menu 'Run'. Je moet het programma eerst opslaan.



Kies 'Run Module' in het menu 'Run' in het Code-venster

3 De game begint in het Shell-venster. Het spook zit verstopt achter een van de drie deuren. Welke kies je? Typ 1, 2 of 3 en druk op 'Enter'.

Ghost Game
Three doors ahead...
A ghost behind one.
Which door do you open?
1, 2, or 3?

Voer je keuze in

4 Het doel van het spel is een deur kiezen waarachter geen spook zit. Als je dat lukt, ga je naar de volgende kamer en ben je een niveau verder.

Ghost Game
Three doors ahead...
A ghost behind one.
Which door do you open?
1, 2, or 3? 3
No ghost!

Hier verschijnt het getal dat je hebt ingetypt

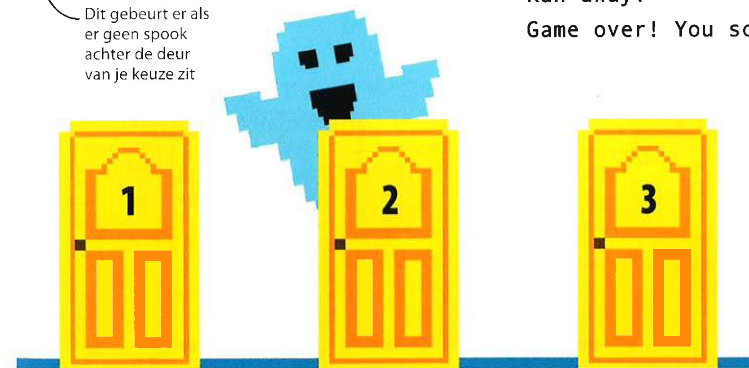
Dit gebeurt er als er geen spook achter de deur van je keuze zit

5 Als je pech hebt en je kiest een deur waarachter een spook zit, is het spel afgelopen. Begin opnieuw en kijk of je je laatste score kunt verbeteren.

Ghost Game
Three doors ahead...
A ghost behind one.
Which door do you open?
1, 2, or 3? 2
GHOST!
Run away!
Game over! You scored 0

Dit gebeurt er als er een spook achter de deur zit

De score geeft aan hoeveel kamers je hebt overleefd



Spokenspel gedecodeerd

Het spokenspel bevat enkele sleutelkenmerken van Python. Je kunt de code ontmantelen om te zien hoe het programma is opgebouwd en wat de verschillende onderdelen allemaal doen.

Coderingsstructuur

Python gebruikt spaties aan het begin van regels om aan te geven welke instructies bij elkaar horen. Deze spaties heten 'inspringingen'. De code na 'while feeling_brave' bijvoorbeeld springt in met vier spaties om te tonen dat het allemaal bij de hoofd-loop hoort.

```
# Ghost Game
from random import randint
print('Ghost Game')
feeling_brave = True
score = 0
while feeling_brave:
    ghost_door = randint(1, 3)
    print('Three doors ahead...')
    print('A ghost behind one.')
    print('Which door do you open?')
    door = input('1, 2 or 3?')
    door_num = int(door)
    if door_num == ghost_door:
        print('GHOST!')
        feeling_brave = False
    else:
        print('No ghost!')
        print('You enter the next room.')
        score = score + 1
print('Run away!')
print('Game over! You scored', score)
```

ZIE OOK
(96-97) Spokenspel
Programmastroom
100-101

1 Installatie
Deze instructies werken maar één keer: aan het begin van het spel. Ze installeren de titel, variabelen en het commando 'randint'.

```
# Ghost Game
from random import randint
print('Ghost Game')
feeling_brave = True
score = 0
```

Dit zet de score weer op 0

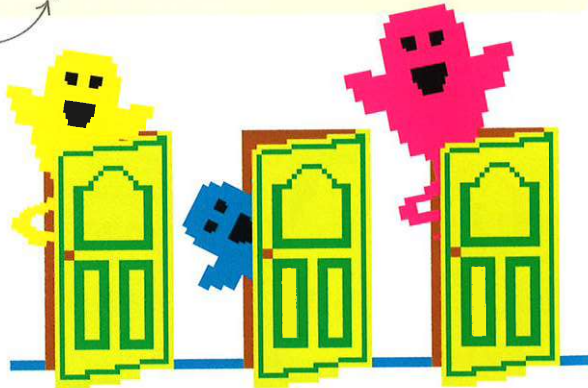
Dit installeert het commando 'randint', dat willekeurige getallen genereert

Het commando 'print' geeft de tekst weer als het spel bezig is

TIPS VAN EXPERTS

Typ zorgvuldig

Als je Python gebruikt, voer de codering dan zorgvuldig in. Als je een dubbele punt, aanhalingsteken of haakje vergeet, werkt het programma niet goed. Ook moet je hoofdletters en spaties exact overnemen.



2 De hoofd-loop

Deze loop vertelt het verhaal en ontvangt de gok van de speler. Hij blijft werken zolang er geen spook achter de gekozen deur zit. Verschijnt er een spook, dan verandert de variabele 'feeling_brave' in 'False' en stopt de loop.

```
while feeling_brave:
    ghost_door = randint(1, 3)
    print('Three doors ahead...')
    print('A ghost behind one.')
    print('Which door do you open?')
    door = input('1, 2 or 3?')
    door_num = int(door)
```

Dit selecteert een willekeurig getal tussen 1 en 3

Het commando 'print' geeft de tekst weer op het scherm

Deze regel vraagt naar het antwoord van de speler

3 Vertakkingsdeel

Het programma neemt een ander pad afhankelijk van of er een spook achter de gekozen deur zat of niet. Zat er een spook, dan wordt de variabele 'feeling_brave' vervangen door 'False'. Was er geen spook, dan neemt de score van de speler met 1 toe.

```
if door_num == ghost_door:
    print('GHOST!')
    feeling_brave = False
else:
    print('No ghost!')
    print('You enter the next room.')
    score = score + 1
```

Deze vertakking wordt genomen als er een spook achter de gekozen deur zit

Als er geen spook is, ziet de speler dit bericht

De score neemt toe met 1 als de speler een kamer binnengaat zonder een spook tegen te komen

4 Spel beëindigen

Dit werkt slechts eenmalig: als je het spook tegenkomt en de loop wordt beëindigd. Python weet dat het niet bij de loop hoort omdat er niet wordt ingesprongen.

```
print('Run away!')
print('Game over! You scored', score)
```

De score is een variabele; hij verandert afhankelijk van het aantal kamers dat de speler binnengaat

ONTHOUD

Wat heb je bereikt?

Gefeliciteerd – je hebt je eerste Python-game gemaakt! Later in het boek zul je meer leren over deze commando's, maar je hebt nu al veel bereikt:

Je hebt een programma in Python getypt en opgeslagen.

Je hebt geleerd hoe je een Python-programma kunt uitvoeren.

Je hebt inspringingen gebruikt om een programma te structureren.

Je hebt variabelen gebruikt om de score bij te houden.

Je hebt berichten op het scherm weergegeven.

