

Python

D. Leeuw

10 november 2025

0.0.0

© 2025 Dennis Leeuw



Dit werk is uitgegeven onder de Creative Commons BY-NC-SA Licentie en laat anderen toe het werk te kopiëren, distribueren, vertonen, op te voeren, en om afgeleid materiaal te maken, zolang de auteurs en uitgever worden vermeld als maker van het werk, het werk niet commercieel gebruikt wordt en afgeleide werken onder identieke voorwaarden worden verspreid.

1 Wat is Python?

Python is een programmeertaal. Met een programmeertaal kan je een computer dingen voor je laten doen. Doordat de computer dan werk voor je doet kan jij andere dingen doen (nee, niet gamen). Tijdens het programmeren gebruik je een programmeertaal om programma-code of kortweg code te schrijven. De programma-code schrijf je in een tekst-bestand zonder opmaak, we gebruiken daarom geen Word om een computer programma te schrijven maar een editor. Een editor is een heel simpele tekstverwerker zonder alle speciale functies als het invoegen van plaatjes, het nummeren van pagina's of het maken van hoofdstuk-koppen. De meeste programmeurs gebruiken een IDE (Integrated Development Environment). Een IDE heeft naast een editor ook mogelijkheden om code direct uit te voeren, daarnaast geven IDE's vaak tips en adviezen om veel voorkomende fouten te voorkomen en om code aan te vullen.

Er zijn verschillende talen om een computer te vertellen wat hij moet doen. Python is er daar één van. Programmeertalen kunnen grofweg in twee soorten verdeeld worden. We hebben de scripting-talen en de talen die eerst gecompileerd moeten worden.

Talen die eerst gecompileerd moeten worden worden ook geschreven in een editor of een IDE, maar nadat het programma geschreven is gaat deze eerst door een compiler. Deze compiler maakt van de programma-code een binary. Op Windows systemen is dit vaak een bestand met de extensie .exe van executable. Een executable is een programma dat direct door een computer uitgevoerd kan worden. De binary bevat 1-en en 0-en die direct door de computer begrepen worden. Een executable kan dan ook op elke computer, met dezelfde processor en hetzelfde operating system, uitgevoerd worden zonder dat de programma-code aanwezig is.

Scripting talen hebben geen compiler, maar een interpreter. De taal waarin het programma is geschreven wordt op de computer door de interpreter omgezet naar machinetaal, of wel de 1-en en 0-en, om dan uitgevoerd te worden. Op elke computer waarop je het programma wilt draaien moet dus een interpreter en de code aanwezig zijn om de code te kunnen uitvoeren.

Python is een scriptingtaal en heeft dus een Python-interpreter nodig om gebruikt te kunnen worden op je computer.

1.1 Opdrachten

1. Zoek op Internet naar 3 programmeertalen die gecompileerd moeten worden en welke compiler daarvoor gebruikt kan worden

2. Zoek op Internet naar 3 programmeertalen die net als Python ook scripting-talen zijn en welke interpreter daarvoor nodig is

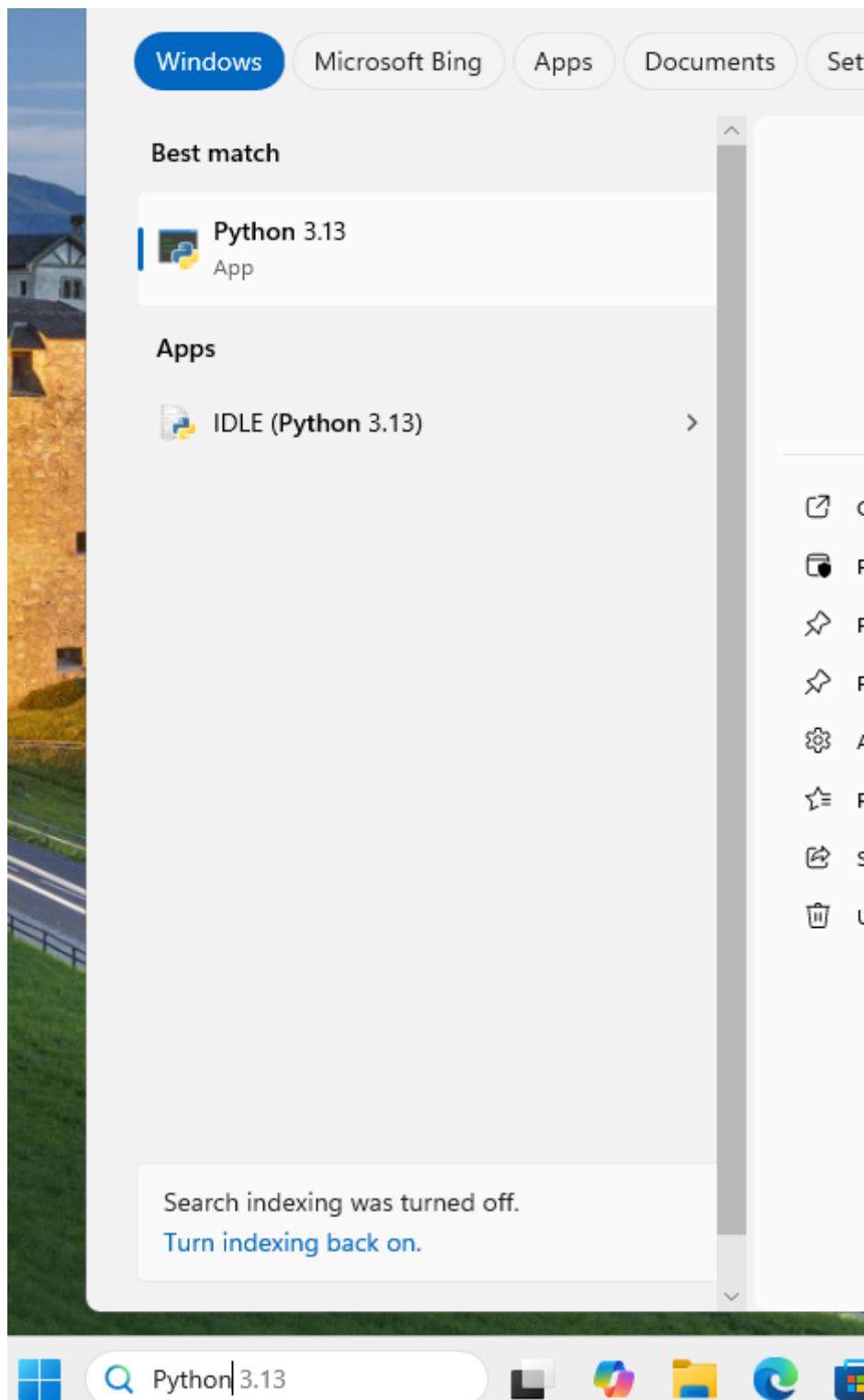
3. Zijn compilers en interpreters binaries? Leg je antwoord uit.

2 Installatie van de Python-interpreter

2.1 Windows

Volg de instructies op <https://learn.microsoft.com/en-us/windows/python/scripting> om Python te installeren. Alleen het kopje “Install Python”. Je installeert nu de Python-interpreter.

Als Python geïnstalleerd is kan je via de zoekbalk Python opzoeken en opstarten, zoals je kan zien in 1



Figuur 1: Zoek en start Python

Om de Python-interpreter af te sluiten type je `exit()`. Denk om de haakjes aan het einde.

3 Hello World!

De meeste programmeertalen beginnen bij hun uitleg over de taal met het op het scherm printen van de tekst “Hello World!”, laten wij voor deze Python lessen daar ook maar mee beginnen. Python maakt gebruik van een interpreter en dus kunnen we de interpreter opstarten en daar het commando intypen. Start Python op door op de command line `python` te typen en enter te geven.

```
Python 3.11.2 (main, Apr 28 2025, 14:11:48) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello World!")
Hello World!
>>> exit()
```

De »> geeft aan dat je op de Python prompt zit. Na deze drie groter dan tekens mag je python code intypen. In ons geval bestaat de opdracht aan Python uit het commando `print()` met tussen quotes de tekst die we op het scherm afgebeeld willen zien. Na de enter zal Python direct de tekst laten zien. Met het commando `exit()` verlaten we de interpreter.

We kunnen ook een tekstbestand maken in onze editor met daarin het commando:

```
print("Hello World!")
```

Als we dit document opslaan met een `.py` extensie, bijvoorbeeld als `hello.py` dan kunnen we dit script door `python` laten uitvoeren door het script als argument mee te geven aan `python`:

```
python hello.py
```

Het uitvoeren van commando's direct in de interpreter is makkelijk om zaken te testen, maar om dingen ook voor de toekomst veilig te stellen is het beter om scripts te gebruiken zodat scripts die je een keer gemaakt hebt ook later weer kunt gebruiken.

4 IDE

Een Integrated Development Environment is meer dan een code editor. Een IDE helpt je bij het schrijven van code door syntax-highlighting, code aanvullingen en waarschuwingen als je iets schrijft dat tegen de regels (syntax)

is. Ook een vast onderdeel van een IDE is de mogelijkheid om je code snel te testen met vaak ook opties om dat bijvoorbeeld stap voor stap te doen, zodat je makkelijker kan zien op welke regel het fout gaat.

4.1 Windows: Visual Studio Code

- Zoek in de Windows Store op Visual Studio Code en installeer deze
- Installeer de Python extension voor Visual Studio Code
- Start PowerShell en run:

```
python.exe -m pip install --upgrade pip
```

4.2 Linux: Visual Studio Code

- Download je linux versie van <https://code.visualstudio.com/>
- Voor .deb doe:

```
sudo dpkg -i ~/Downloads/ode_1.101.0-1749655245_amd64.deb # Vervang  
het versienummer met jouw versie
```

Voor .rpm doe:

```
sudo rpm -i ~/Downloads/ode_1.101.0-1749655245_amd64.rpm # Vervang  
het versienummer met jouw versie
```

- Sta Microsoft toe om de applicatie toe te voegen aan de package database

5 Code documentatie

Tijdens het programmeren is het belangrijk om vast te leggen waarom je iets doet of gedaan hebt. Als je 5 jaar later naar je eigen code kijkt is het soms al moeilijk om te begrijpen waarom je een bepaald probleem zo hebt opgelost laat staan als iemand anders naar je code kijkt. Het is dus niet zinnig om uit te leggen wat de code doet, dat kan iedereen lezen, maar het is wel belangrijk om uit te leggen waarom je code doet wat het doet en wat het doel van de code is.

Om commentaar (comments) toe te voegen binnen je code gebruiken we het `#` karakter. Commentaar wordt niet uitgevoerd door de interpreter. Alles vanaf het `#` teken tot het einde van de regel wordt door de interpreter beschouwd als commentaar en daar wordt niets mee gedaan. We kunnen

commentaar op een eigen regel zetten door de regel te beginnen met het `#` of we kunnen commentaar na een opdracht zetten door na de code een `#` te plaatsen met het commentaar.

```
# De opdracht was om in Python te zorgen  
# dat de uitkomst van 5+6 op het scherm te zien is.  
print(5+6) # Commentaar na een opdracht
```

Het is ook handig om aan het begin van je code aan te geven wat anderen met je code mogen doen, dus welk copyright en welke licentie we hebben. Ook kan je aangeven wat het programma doet en welke opties er bijvoorbeeld beschikbaar zijn.

```
# (C) 2025 Dennis Leeuw  
# Licentie: GPLv3  
  
# Dit programma zorgt ervoor dat ...
```

Vanaf nu maken we er een goede gewoonte van om onze code altijd te voorzien van commentaar. We geven voordat de code begint aan wat het totale programma doet en waar nodig beschrijven we de keuzes die we gemaakt hebben en waarom we voor een bepaalde oplossing gekozen hebben.

Index

binary, 2
code, 2
commentaar, 6
comments, 6
compiler, 2
editor, 2
executable, 2
gecompileerd, 2
IDE, 2, 5
Integrated Development Environment, 2, 5
interpreter, 2
machinetaal, 2
programma-code, 2
programmeertaal, 2
programmeren, 2
scripting, 2