

# *Python Programmeren voor beginners zonder ervaring*

## INHOUDSOPGAVE

<b>Python Programmeren voor beginners zonder ervaring.....</b>	<b>0</b>
<b>Inleiding.....</b>	<b>5</b>
<b>Basisvaardigheden voor Python-programmeren.....</b>	<b>8</b>
Werken met mappen en bestanden.....	8
Vinden en openen van de terminal of opdrachtprompt.....	9
Programma's uitvoeren in de terminal.....	10
Basisbegrippen van de commandoregel.....	11
<b>Installatie en Eerste Stappen.....</b>	<b>13</b>
Installeren van Python op Windows.....	13
Installeren van Python op macOS.....	14
Installeren en Gebruiken van Thonny.....	15
Eerste Python-programma: "Hello, World!" .....	16
<b>Basisprincipes van Python.....</b>	<b>17</b>
Variabelen en Datatypes.....	17
Basisbewerkingen.....	18
Strings en Stringbewerking.....	19
Input en Output.....	21
<b>Controlestromen en Beslissingen.....</b>	<b>23</b>
If-else statements.....	23
Logische operatoren.....	26
Geneste beslissingen.....	27
<b>Lussen en Iteraties.....</b>	<b>30</b>
De for-lus.....	30
De while-lus.....	32

Lussen en lijsten combineren.....	33
Break en continue in lussen.....	35
Geneste lussen.....	36
<b>Functies en Modulariteit.....</b>	<b>38</b>
Wat zijn functies?.....	38
Een functie definiëren en aanroepen.....	39
Argumenten en parameters.....	39
Return-waardes.....	41
Standaardwaarden voor parameters.....	41
Ingebouwde functies.....	42
Functies en modulariteit.....	43
<b>Lijsten, Tuples en Dictionaries.....</b>	<b>45</b>
Lijsten in Python.....	45
Tuples in Python.....	48
Dictionaries in Python.....	50
<b>Objectgeoriënteerd Programmeren in Python.....</b>	<b>53</b>
Inleiding tot Objectgeoriënteerd Programmeren.....	53
Klassen en Objecten.....	54
Methoden en Eigenschappen.....	55
Overerving en Subklassen.....	57
Encapsulatie en Getter- en Setter-methoden.....	59
<b>Bestanden Lezen en Schrijven.....</b>	<b>63</b>
Bestanden Openen en Lezen.....	63
Bestanden Schrijven en Aanpassen.....	65
CSV-bestanden Verwerken.....	67

<b>Foutafhandeling en Debugging.....</b>	<b>70</b>
Veelvoorkomende fouten en hoe ze op te lossen.....	70
Try-except blokken en foutafhandeling.....	71
Debugging: fouten opsporen en oplossen.....	74
Gebruik van debugging-tools in Thonny.....	75
<b>Werken met Externe Pakketten en Virtual Environments.....</b>	<b>77</b>
Wat is pip en hoe gebruik je het?.....	77
Wat zijn virtuele omgevingen en waarom zijn ze belangrijk?....	80
Een virtuele omgeving maken en activeren.....	81
Installeren van externe pakketten in een virtuele omgeving.....	82
<b>Databases en SQLite3 voor Beginners.....</b>	<b>85</b>
Wat is een database?.....	85
Basisprincipes van databaseontwerp.....	86
Wat is SQLite 3?.....	87
Werken met SQLite 3 in Python.....	87
<b>Grafische Gebruikersinterfaces (GUI) met Tkinter.....</b>	<b>94</b>
Wat is Tkinter en waarom gebruiken we het?.....	94
Een eenvoudig Tkinter-venster maken.....	95
Het hoofdvenster aanpassen.....	95
Widgets: Bouwstenen van een GUI.....	96
Labels en Knoppen.....	96
Invoervelden en Tekstvakken.....	98
Checkboxes en Radiobuttons.....	99
Frames en Lay-outbeheer.....	101
Bestanddialogen en Berichten.....	103

<b>Je Python-programma Inpakken en Verspreiden.....</b>	<b>105</b>
Een Python-script als uitvoerbaar bestand maken.....	105
Een Python-pakket maken en publiceren.....	106
Toekomstmogelijkheden: Wat kun je verder met Python?.....	108

# Inleiding

Programmeren is het proces van het schrijven van instructies die een computer kan begrijpen en uitvoeren. In essentie is een programma niets meer dan een reeks opdrachten die een bepaalde taak automatiseren. Waar een mens een recept leest en een gerecht bereidt, volgt een computer een programma en voert het berekeningen of bewerkingen uit. Programmeertalen maken het mogelijk om deze instructies te schrijven in een voor mensen begrijpelijke vorm, die vervolgens wordt vertaald naar machinetaal, zodat de computer ze kan uitvoeren.

In de moderne wereld speelt programmeren een cruciale rol in vrijwel alle aspecten van technologie. Van websites en mobiele apps tot kunstmatige intelligentie en data-analyse: software vormt de kern van innovatie. Programmeervaardigheden zijn niet langer beperkt tot IT-specialisten, maar worden steeds belangrijker in diverse vakgebieden zoals wetenschap, economie, en zelfs de gezondheidszorg. Het leren van programmeren opent deuren naar nieuwe carrièremogelijkheden en biedt tegelijkertijd een krachtig hulpmiddel om problemen systematisch en efficiënt op te lossen.

Python is een van de populairste programmeertalen ter wereld en wordt geroemd om zijn eenvoud en leesbaarheid. In tegenstelling tot veel andere programmeertalen gebruikt Python een intuïtieve en

gestructureerde syntax, waardoor het een uitstekende keuze is voor beginners. De taal wordt breed toegepast, van webontwikkeling en data-analyse tot machine learning en automatisering. Het grote voordeel van Python is dat het een lage instapdrempel heeft, terwijl het tegelijkertijd krachtig genoeg is voor professionele softwareontwikkeling.

Dit boek is bedoeld voor absolute beginners die geen enkele ervaring hebben met programmeren. De opbouw is stapsgewijs, zodat je vanaf de basis kunt beginnen en gaandeweg meer geavanceerde concepten leert. Je zult kennismaken met de fundamentele van Python, leren hoe je een programmeeromgeving installeert, en praktische oefeningen maken om je kennis direct toe te passen. Het doel is om een stevige basis te leggen waarmee je zelfstandig kunt programmeren en verder kunt groeien in je vaardigheden.

In de komende hoofdstukken beginnen we met het installeren van Python en het instellen van een programmeeromgeving op zowel Windows als macOS. Daarna behandelen we de basiselementen van Python, zoals variabelen, functies en bestandsbeheer. Het boek eindigt met een praktische toepassing, waarin je een klein project ontwikkelt om je opgedane kennis in de praktijk te brengen.

Programmeren is niet alleen een technische vaardigheid, maar ook een manier van denken. Door te leren programmeren ontwikkel je analytisch en probleemoplossend vermogen, wat in veel

verschillende contexten van pas kan komen. Hoewel programmeren in het begin misschien overweldigend lijkt, zul je merken dat oefening en experimenteren de sleutel zijn tot succes. Het belangrijkste is om nieuwsgierig te blijven, fouten te zien als leermomenten en plezier te hebben in het creëren van iets nieuws.

Met deze introductie ben je klaar om je eerste stappen te zetten in de wereld van Python-programmeren. In het volgende hoofdstuk beginnen we met de installatie van Python en het opzetten van je ontwikkelomgeving, zodat je direct aan de slag kunt.



# Basisvaardigheden voor Python-programmeren

Voordat we beginnen met programmeren in Python, is het belangrijk om enkele basisvaardigheden onder de knie te krijgen. Dit hoofdstuk richt zich op essentiële computervaardigheden die nodig zijn om effectief met Python te werken. We behandelen het aanmaken van mappen en bestanden, het openen en gebruiken van de terminal of de opdrachtprompt, en het installeren van Python zonder Thonny editor (wordt in volgende hoofdstuk behandeld). Dit hoofdstuk is bedoeld voor absolute beginners en biedt een solide basis voor de volgende stappen in het leerproces. Beschik je al over deze vaardigheden, dan mag je meteen aan de slag met het volgende hoofdstuk.

## Werken met mappen en bestanden

Een veelvoorkomende taak bij programmeren is het organiseren van bestanden en mappen. Dit helpt bij het structureren van projecten en zorgt ervoor dat code en gegevens overzichtelijk blijven. Om een map te maken op Windows, kun je de Verkenner openen, naar een locatie op je computer navigeren, met de rechtermuisknop klikken en "Nieuwe map" selecteren. Geef de map een logische naam, zoals

"MijnEerstePythonProject". Op een Mac werkt dit op een vergelijkbare manier via de Finder.

Binnen deze map moet een Python-bestand worden aangemaakt. Open de map, klik met de rechtermuisknop en kies "Nieuw tekstbestand" op Windows of gebruik de Finder op Mac om een nieuw bestand aan te maken. De bestandsnaam moet eindigen op **.py**, bijvoorbeeld **mijn\_script.py**. Dit geeft aan dat het een Python-programma is.

Een alternatieve manier om een bestand aan te maken is via de terminal of opdrachtprompt. Op Windows kan dit door de opdrachtprompt te openen, naar de juiste map te navigeren met het commando `cd`, en vervolgens `echo. > mijn_script.py` in te voeren. Op Mac en Linux werkt `touch mijn_script.py` in de terminal.

## **Vinden en openen van de terminal of opdrachtprompt**

De terminal of opdrachtprompt is een essentieel hulpmiddel bij het werken met Python. Op Windows wordt dit de "Opdrachtprompt" of "PowerShell" genoemd. Om deze te openen, klik je op de zoekbalk in het Startmenu, typ je "cmd" of "PowerShell" en druk je op Enter.

Op macOS is de terminal te vinden in de map "Hulpprogramma's" binnen "Programma's". Je kunt deze ook snel openen door op **Command** + **Spatie** te drukken, "Terminal" te typen en op Enter te drukken.

Op Linux is de terminal meestal direct beschikbaar in het menu of kan deze worden geopend met de sneltoets **Ctrl** + **Alt** + **T**.

Om te controleren of de terminal correct functioneert, kan het commando `python --version` of `python3 --version` worden uitgevoerd. Dit zou een versieaanduiding van Python moeten weergeven. Als dit niet het geval is, moet Python mogelijk nog worden geïnstalleerd.

## Programma's uitvoeren in de terminal

Om een Python-programma uit te voeren, moet je in de juiste map navigeren via de terminal of opdrachtprompt. Dit kan met het `cd`-commando, gevolgd door de padnaam naar de map waarin het Python-bestand is opgeslagen. Bijvoorbeeld, als het bestand zich bevindt in de map "MijnEerstePythonProject" op het bureaublad, kan op Windows de opdracht `cd`

```
C:\Users\JouwGebruikersnaam\Desktop\MijnEerstePythonProject
```

worden gebruikt. Op macOS en Linux werkt `cd`

```
~/Desktop/MijnEerstePythonProject.
```

Als je eenmaal in de juiste map bent, kan het Python-script worden uitgevoerd door `python mijn_script.py` of `python3 mijn_script.py` in te voeren. De terminal zal dan de uitvoer van het programma weergeven.

## Basisbegrippen van de commandoregel

Bij het werken met Python is het handig om enkele basiscommando's te kennen.

- **ls (Mac/Linux) / dir (Windows):** Toont de bestanden in de huidige map.
- **cd mapnaam:** Verplaatst naar een andere map.
- **cd ..:** Gaat één map omhoog in de mappenstructuur.
- **clear (Mac/Linux) / cls (Windows):** Leegt het terminalscherf.
- **exit:** Sluit de terminal of opdrachtprompt af.

Deze commando's helpen bij het navigeren door bestanden en mappen zonder een grafische interface te gebruiken.

### SAMENVATTING

In dit hoofdstuk heb je geleerd hoe je een map en een Python-bestand aanmaakt, hoe je de terminal of opdrachtprompt opent, en hoe je Python installeert zonder Thonny. Ook heb je een aantal basiscommando's geleerd die nuttig zijn bij het werken met