Routekaart om te leren programmeren in Python.

Inhoud

Rosalie de Klerk – Bambara
www.bambara.nl
Versie 1.0
3-3-2023

Wie ben ik?

Sinds 1996 heb ik mij enthousiast beziggehouden met aan informatie- en communicatietechnologie (ICT). Mijn passie ligt in het creëren en ontwerpen van digitale toepassingen. In 1996 heb ik de opleiding Multimedia Design afgerond aan het Grafisch Lyceum Rotterdam, waar ik mij verdiepte in het ontwerpen van websites, 3D-tekeningen en animaties, én ik leerde werken met Adobe Photoshop *4.0*. Ik kijk met nostalgie terug naar die tijd.

Na het afronden van mijn mbo-opleiding begon ik mijn eigen bedrijf en ontwikkelde ik websites. Pas tien jaar later besloot ik verder te studeren. In 2014 heb ik mijn HBO-opleiding Bedrijfskundige Informatica afgerond en behaalde ik mijn tweedegraads bevoegdheid voor het vak informatica. Hierdoor kon ik ook mijn kennis over ICT delen met geïnteresseerden als docent.

Momenteel volg ik een studie Onderwijswetenschappen aan de Open Universiteit. Naast ICT en onderwijs houd ik ook van reizen en tekenen, zowel digitaal als analoog. Begin 2021 besloot ik mijn vaste baan als docent aan de Hogeschool Rotterdam op te zeggen, zodat ik mij volledig kan richten op tekenen, ondernemen en mijn studie. Aan de hogeschool gaf ik les in de programmeertaal Python en was ik studieloopbaancoach. De passie om mijn kennis te delen en leerlingen en studenten te begeleiden blijft onverminderd aanwezig, en het is juist deze passie die mij ertoe heeft aangezet om dit document te schrijven

Dit document is bedoeld voor iedereen, van kinderen tot ouders en verzorgers van kinderen die geïnteresseerd zijn in het leren programmeren in Python. Het bevat in hoofdlijnen de essentiële informatie die ik mijn leerlingen en studenten altijd heb bijgebracht, zoals *hoe je begint met programmeren*, *wat je moet doen* en *wat er van jou wordt verwacht als beginnend programmeur*.

Laten we beginnen...

Waar moet je beginnen?

"*Mevrouw, het boek is enorm dik. Kunt u het niet allemaal even voor doen? Dan kijken wij wel mee en zo leren wij het écht sneller*." Dit is één van de opmerkingen die zowel leerlingen als studenten maken. Uit ervaring en diverse onderzoeken, waaronder het onderzoek van Dunlosky (Dunlosky, Rawson, Marsh, Nathan, & Willingham, 2013), blijkt echter dat het passief aanschouwen van een docent die een stuk code schrijft niet leidt tot kennis. Als studenten geen ervaring hebben met het bedenken van codes, zullen zij niet in staat zijn om de toets te halen. Daarom vind je in dit document geen oplossingen voor opdrachten, maar een routekaart die uitlegt hoe je Python kunt leren programmeren.

"Hoe dan? Op welke manier kan ik dan het beste leren programmeren?"

Leerstrategie

Volgens Dunlosky bestaan er diverse leerstrategieën die variëren in hun effectiviteit. Maar hoe weet je welke strategie het meest geschikt is voor jou? Een van de besproken strategieën is elaboratie, waarbij de student zichzelf voortdurend vragen stelt over de leerstof, op zoek gaat naar verbanden en controleert wat hij denkt te weten. De specifieke vragen die je jezelf stelt bij elaboratie, zijn deels afhankelijk van de onderwerpen die je bestudeert. In het geval van het programmeren in Python zijn dit bijvoorbeeld vragen als: hoe werkt dit stukje code? Waarom gaat de robot naar links? Wanneer kan ik twee getallen bij elkaar optellen? Wat veroorzaakte die botsing precies? Wat is het resultaat van deze code?

Eerst de basis!

Ik vind dat studenten eerst moeten weten wat programmeren nu eigenlijk is, wat je ermee kunt en hoe je de basisprincipes onder de knie krijgt, voordat je aan de slag gaan met dikke boeken en opdrachten.

Computational thinking

Ik heb het in deze routekaart over het leren programmeren in Python, maar welke programmeertaal je ook kiest, je zult eerst moeten trainen in wat men noemt 'computational thinking'. Door op die manier te kunnen denken, ben je niet ineens een programmeur, maar je leert een probleem te ontleden in kleine deelproblemen. Je leert abstract na te denken. Dit doet een programmeur ook.

Eerst even een aantal definities

Seymour Papert (1928 – 2016) een in Zuid-Afrika geboren Amerikaans hoogleraar wiskunde en pedagogie aan het Massachusetts Institute of Technology introduceerde het begrip “Computational Thinking (CT)”. (Papert, Children,Computers and Poweful Ideas, 1980). Papert was een pionier in kunstmatige intelligentie.

Jeanette Wing, hoogleraar aan de Universiteit van Pittsburgh, definieert Computational Thinking als volgt: “Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science” (Wing, 2006 p.33).

De Stichting Leerplan Ontwikkeling (SLO) definieert computational thinking op haar website als volgt: “Computational thinking is het procesmatig (her)formuleren van problemen op een zodanige manier dat het mogelijk wordt om met computertechnologie het probleem op te lossen. Het gaat daarbij om een verzameling van denkprocessen waarbij probleemformulering, gegevensorganisatie, -analyse en -representatie worden gebruikt voor het oplossen van problemen met behulp van ICT-technieken en -gereedschappen” (SLO, 2015).

Als voorloper in het bijbrengen van Computational Thinking Skills wordt het Verenigd Koningrijk genoemd. Daar is programmeren sinds 2014 een verplicht onderdeel voor alle openbare scholen. Centraal in dat onderwijs staat het trainen van leerlingen in het oplossen van problemen in computer termen. Britse kinderen leren vanaf hun 5e jaar Computational Thinking. Dit wordt ook wel, in navolging van “begrijpend lezen”, “begrijpend programmeren” genoemd. Als kinderen op hun 11e jaar het primair onderwijs verlaten kennen ze twee computertalen en kunnen ze robots en veiligheidssystemen besturen. In het voortgezet onderwijs breiden zij hun kennis verder uit onder begeleiding van natuurkunde-, wiskunde- en IT-specialisten (Kennisnet, 2015).

Dat is toch wel even iets anders dan wat ik de afgelopen jaren in het Nederlands onderwijs heb meegemaakt.

Wat wordt er van mij verwacht?

Je hebt al interesse getoond door deze routekaart te downloaden, een goed begin! Het antwoord op de vraag wat er van jou wordt verwacht is eigenlijk heel simpel: wees altijd leergierig, blijf zelf nadenken, geef niet op als je het even niet weet en vooral, begin gewoon! Deze eigenschappen zijn de sleutel tot het beheersen van de basis van programmeren.

Als ervaren programmeerdocent heb ik gemerkt dat veel studenten vastlopen door dikke boeken te gebruiken met kleine lettertjes, te veel opdrachten in te weinig tijd en docenten die de opdrachten voor ze uitvoeren. Maar met deze routekaart kun jij op je eigen tempo de basis helemaal onder de knie krijgen.

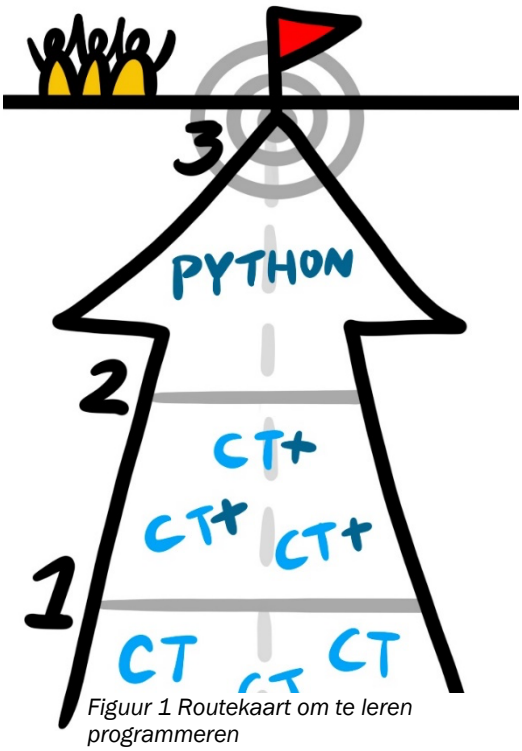
Duizenden mensen zijn je al voorgegaan, dus ik weet zeker dat jij het ook kunt! Blijf geïnteresseerd en toegewijd, en voor je het weet ben je een echte programmeerexpert.

Veel succes en plezier gewenst!

Programmeren

Het is wel duidelijk dat ik geen voorstander ben van dikke theorieboeken en te veel voordoen. Dat ga ik dan, zoals beloofd, ook écht niet doen. En ik 'gooi' je ook niet meteen in het diepe. Wist je dat Youtube, Instagram, Netflix en nog talloze andere bekende bedrijven allemaal zijn gebouwd met de programmeertaal Python? Ja, echt waar! Je gelooft het niet? Check dan **deze** te gekke video van Buvesa. **Welke route moet je volgen om goed te leren programmeren?** Koop in ieder geval niet meteen een boek, je weet wel zo'n dik boek met veel opdrachten. Nee, volg gewoon deze routekaart.

Uitleg Route



Figuur 1 Routekaart om te leren programmeren

Stap 1: Visuele basis van programmeren met RoboMind Academy

Leer hoe je bevelen kunt geven aan het robot autootje door gebruik te maken van een visueel programma-interface.

1. Ga naar de website van RoboMind Academy (www.robomindacademy.com) en maak een gratis account aan.
2. Start met de cursus 'Ho*ur of Code'* en vervolg je weg met '*Basics 1 Course*' en '*Basics 2 Course*'.
3. Vind je het héél leuk? Probeer dan ook deze cursus nog: Advanced 1 Course.

Stap 2: Visuele programmeren in Python met RoboMind Academy

Leer de basis van Python (If, then else, While, for loops etc), functies en variabelen zijn geen onbekende begrippen meer straks.

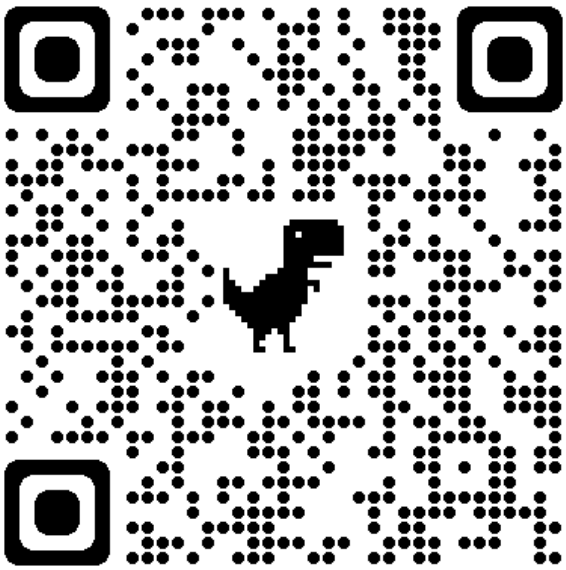
1. Ga naar de volgende cursus in RoboMind Academy, genaamd 'Python Course'.
2. Doorloop de verschillende lessen en oefeningen en leer de basisprincipes van de taal Python.
3. Probeer verschillende opdrachten uit en verfijn je kennis en vaardigheden.

Stap 3: Naar de 'echte' programmeeromgeving van Python

Gebruik voor de laatste drie stappen het volgende boek: www.spronck.net/pythonbook/pythonboektablet.pdf

1. Na het voltooien van de cursus 'Programming in Python' in RoboMind Academy, is het tijd om de overstap te maken naar de 'echte' programmeeromgeving van Python.
2. Download een populaire Python-ontwikkelomgeving, zoals IDLE of PyCharm.
3. Experimenteer met het schrijven van simpele scripts en oefen met het uitvoeren van taken in deze omgeving. Blijf oefenen en leren om je vaardigheden verder te verbeteren.

Ben je tevreden met dit document? Dan vraag ik je om dit kenbaar te maken middels een vrijwillige bijdrage: scan de QR-code.



Bibliografie

Dunlosky, J. (2013). *Strengthening the student toolbox*. American Educator. https://www.academia.edu/29950252/Strengthening_the_Student_Toolbox_study_strategies_to_Boost_learning

Papert, Seymour. (1980). Mindstorms: Children, Computers, and Powerful Ideas ..

SLO. 2015. Computational Thining. Geraadpleegd van http://curriculumvandetoekomst.slo.nl/21e-eeuwse-vaardigheden/digitale-geletterdheid/computational-thinking

Wing, Jeannette. (2006). Computational thinking. Communications of the ACM. ACM