Kennismaken met programmeren

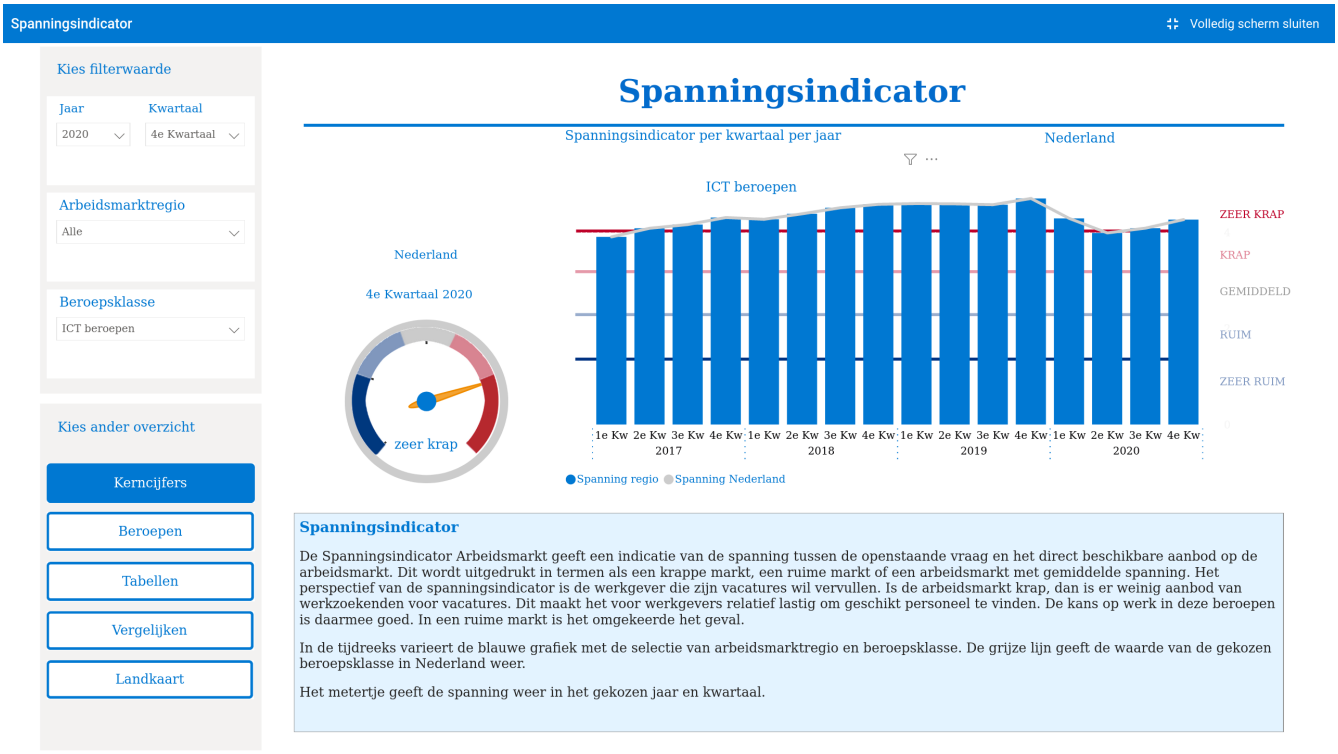
Erwin Matijsen

V1.0, 20-04-2021

Inhoudsopgave

Introductie

Als je bedenkt dat werkgevers al jaren moeite hebben om hun vacatures te vullen, dan begrijp je dat het UWV de ICT-branche als een kansrijk beroep kenmerkt. Misschien zit jij momenteel wel in een heel andere sector, een sector die juist *niet* als kansrijk wordt gekenmerkt.



Afbeelding 1. ICT-beroepen: al jaren een krappe markt

Tijd dus om je om te scholen naar de ICT-sector? Misschien wel. Maar hoe weet je of de ICT iets voor jou is? Misschien heb je geen (goed) beeld van wat 'de ICT' nou precies inhoudt. Of het wel iets voor jou is.

In dit (online) boek krijg je een eerste inkijkje in die ICT-sector, juist om te bepalen of het eventueel iets voor je is. Nu zijn er vele beroepen die tot de ICT-sector gerekend kunnen worden. In dit boek zoomen we in op één van die beroepen: de programmeur.

Het beroep programmeur komt in vele vormen en gradaties. We kunnen daarom alleen het topje van de ijsberg laten zien. Dit kleine voorproefje van de wondere wereld van programmeren heeft twee doelen:

1. Uitleggen *wat* programmeren is en wat er allemaal bij komt kijken.
2. Je laten *ervaren* wat programmeren is, zodat jij beter kunt inschatten of een carrièreswitch naar programmeur iets voor jou is.

Leeswijzer

In Hoofstuk 1 leggen we uit wat programmeren nu eigenlijk precies inhoudt. Wat doe je als programmeur, waar kom je te werken? Wat voor dingen kun je maken?

In Hoofdstuk 2 verlaten we de theorie om aan de slag te gaan. Op de micro:bit (zie hieronder) programmeren we — nog zonder een programmeertaal te gebruiken — het bekende spelletje *rock-paper-scissors*.

In Hoofdstuk 3 programmeren we hetzelfde spelletje nogmaals, maar dit keer in de programmeertaal Python.

We sluiten af in hoofdstuk 4 met wat voor jou mogelijke vervolgstappen zijn. Misschien vond je het wel helemaal niets. Goed dat je dat nu weet, even goede vrienden! Werd jij wel helemaal enthousiast over het creëren van je eerste programma? Super, dan is een opleiding, cursus of stage misschien iets voor je.

micro:bit

De micro:bit is een klein computertje met onder andere enkele LED-lampjes, twee knoppen en mogelijkheden om allerlei sensoren aan te sluiten. Je kunt de micro:bit hier kopen. Koop de Micro:bit V2 GO bundel om meteen van start te kunnen.

Je kunt dit boek ook prima volgen zonder micro:bit, maar voor het plezier en de ervaring raad ik aan er een te kopen.

Ik wens je heel veel plezier!

Erwin Matijsen
LinkedIN

Chapter 1. Wat is programmeren?

Rondom programmeurs bestaat voor velen een beeld van de zonderlinge, in donker gehulde, als een razende typende man (ja, meestal is het een man). Een beeld dat grotendeels door films en series is geschapen. In die films doen die programmeurs knappe maar onbegrijpelijke dingen, ofwel om de wereld te redden ofwel om de wereld te vernietigen.

Alhoewel niet zo dramatisch, ben ik ervan overtuigd dat software de wereld inderdaad kan veranderen (en dat ook al heeft gedaan). Ik ben er ook van overtuigd dat dit nooit zal gebeuren door die ene zonderlinge figuur. En zeker niet door het manisch typen van regels code, afgezonderd van de rest van de wereld.

Maar wat doet een programmeur dan wel? Laten we eerst het beeld bijstellen van *de* programmeur, voor zover die al bestaat.

1.1. Wat zijn de kenmerken van een programmeur?

In de Factsheet ict-beroepen van het UWV lezen we dat softwareontwikkelaars vrijwel uitsluitend middelbaar- of hoger zijn opgeleid. Bijna 80% werkt voltijds, het gros is twintiger, dertiger of veertiger (bijna 80% valt in de leeftijd 25-55 jaar).

Bekijk je de openstaande vacatures dan valt op dat het vrijwel altijd een functie betreft *in een team*. Het beeld van de programmeur als solist, eenzaam 's nachts aan het werk klopt dan ook niet. Programmeren is zoals veel kenniswerk: samenwerken, overleggen, lezen, leren, afstemmen, communiceren. De Factsheet van het UWV omschrijft dit helder:

> Daarnaast worden algemene kennis en sociale vaardigheden steeds relevanter. Zo is er in toenemende mate behoefte aan mensen met inzicht in bedrijfsprocessen en communicatieve vaardigheden, die klantwensen kunnen vertalen naar ict. Ict’ers worden geacht goed samen te werken met klanten. Ook moeten zij steeds vaker samenwerken met mensen uit andere disciplines.
>
> — UWV, Factsheet ict-beroepen - pagina 5

1.2. Wat doet een programmeur?

Laten we vooropstellen dat een programmeur vaak aan het programmeren is. Ofwel: in een teksteditor, in een bepaalde programmeertaal code aan het typen is. Maar voordat je als programmeur code intypt is er al een heel proces aan vooraf gegaan:

1. Er is besloten dat juist dít prioriteit heeft
2. Iemand (of meerdere personen) hebben nagedacht over wat het probleem is, en wat de oplossingsrichting is
3. Er is een inschatting gemaakt van hoeveel tijd dit (ongeveer) gaat kosten

En dit gaat alleen nog maar over het proces voorafgaande aan het schrijven van een klein stukje code. Daar weer vooraf zijn waarschijnlijk vele overleggen, plannen, schetsen, uitwerkingen, etc. gegaan.

Ben je dan toe aan het schrijven van de code, dan is dit nog steeds niet het enige dat je doet. Je vraagt je collega misschien om hulp, je zoekt in de documentatie van de betreffende programmeertaal, je zoekt op Stack Overflow naar hulp. Je bepaalt hoe jouw stukje code in het grotere geheel past en welke code je kunt hergebruiken. Hiervoor lees je de code die jij of een collega eerder al hebt geschreven.

Een veelgehoorde uitspraak in de softwareindustrie is: "Code wordt vaker gelezen dan geschreven."

Na het schrijven van jouw code — inclusief het schrijven van documentatie en geautomatiseerde tests! — is het tijd voor één van je collega's om jouw code te *reviewen*, ofwel mee te lezen of er geen grote of kleine fouten zijn gemaakt, of tips te geven voor een nog betere oplossing.

Code geschreven, getest en *gereviewed*? Dan is het tijd om de nieuwe code in gebruik te nemen. En nu maar hopen dat er geen *bugs* tevoorschijn komen…

1.3. Waar kom je te werken en wat maak je daar?

Elk bedrijf maakt gebruik van verschillende pakketten software. Denk aan boekhoudpakketten of Customer Relationship Managementsystemen. Maar ook specifieke software voor specifieke branches, zoals reserveringssystemen voor restaurants of leerlingvolgsystemen in het onderwijs.

Dit soort pakketten worden geleverd door softwarebedrijven. Bij uitstek een plek waar jij kunt komen te werken dus! Maar je kunt ook terecht komen bij een groot bedrijf waar je werkt aan software dat enkel door dat bedrijf wordt gebruikt. Echt maatwerk dus.

Software is overal, in elke branche. En programmeurs zijn dan ook in elke branche te vinden. Dit betekent dat jij je kunt specialiseren in een bepaalde branche, of je specialiseert je juist in een bepaalde techniek die je in elke branche kunt inzetten.

De mogelijkheden zijn eindeloos, en teveel om op te noemen. Om een indruk te krijgen kun je bijvoorbeeld eens kijken op ictergezocht.nl, bijvoorbeeld onder Software Developer, Backend Developer of Software Ontwikkelaar.

Chapter 2. Eerste stappen

Genoeg theorie, tijd om te ervaren wat programmeren is! Alhoewel programmeren meestal bestaat uit het schrijven van code, beginnen wij op een meer visuele manier van programmeren.

2.1. micro:bit

Het doel is om een spelletje te programmeren die je op de micro:bit kunt zetten. De micro:bit is een klein computertje met:

- 25 LED lampjes
- 2 knopjes
- Een microfoon en een luidspreker
- Een kompas
- Een snelheidsmeter
- Bluetooth
- USB
- Mogelijkheden om uit te breiden met sensoren

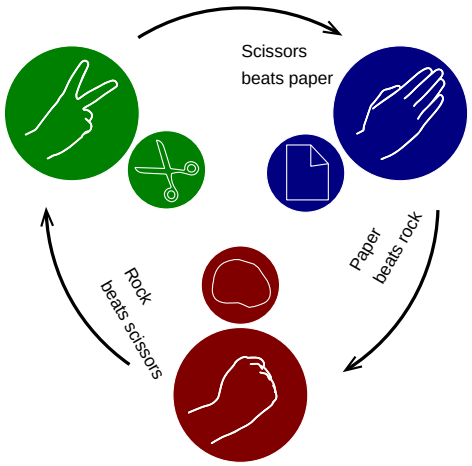
En dit alles werkend op 2 AAA-batterijen. Al met al genoeg mogelijkheden om leuke projecten mee te maken! Om het niet direct te overweldigend te maken gaan we alleen de snelheidsmeter en de LED-lampjes gebruiken.

2.2. Rock-Paper-Scissors

Het spelletje ken je vast: twee mensen tellen af en maken tegelijkertijd met hun hand een teken:

- Steen (Rock)
- Papier (Paper)
- Schaar (Scissors)

Een van de partijen wint doordat het ene symbool de andere verslaat. Zie onderstaande afbeelding:



Afbeelding 2. Rock-Paper-Scissors

Ons doel is onze hand te vervangen door de micro:bit. Door de micro:bit te schudden zal het willekeurig een steen, papier of schaar laten zien met behulp van oplichtende LED-lampjes.

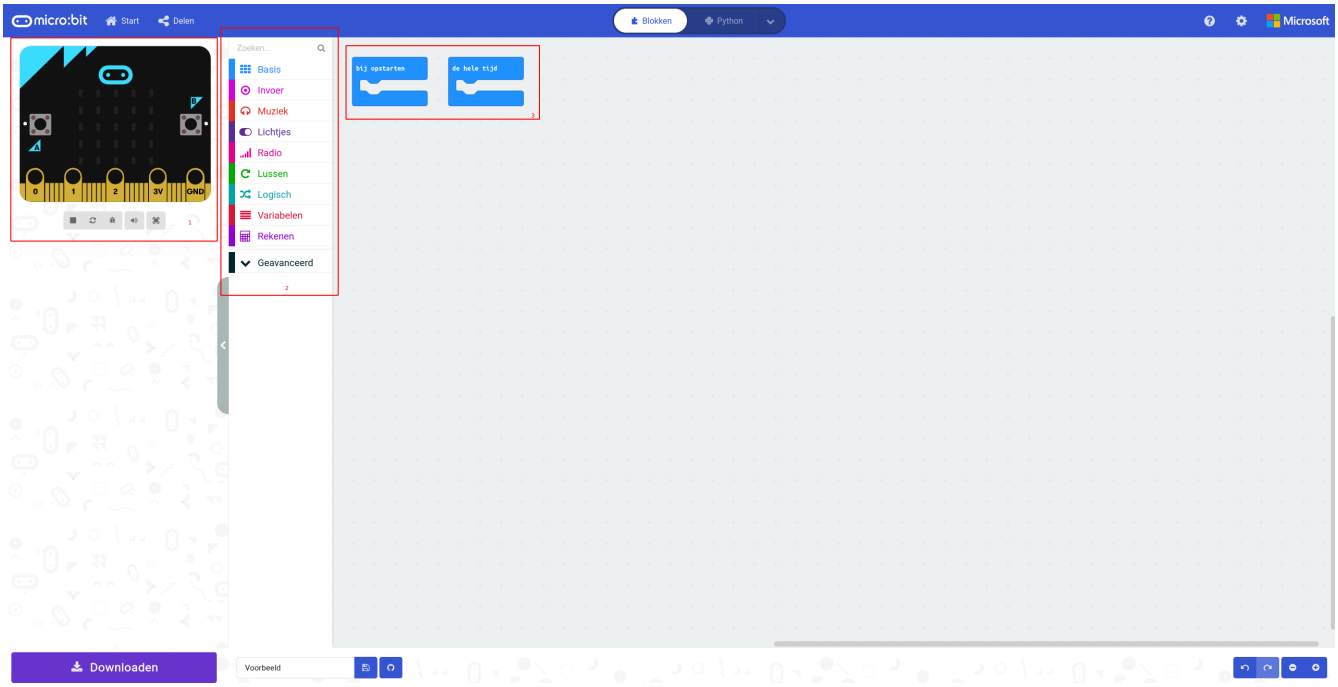
2.3. Programmeren met blokken

Je eerste stappen in de wondere wereld van programmeren zet je in een online omgeving (*MakeCode* genaamd) waar je kunt programmeren door blokken te plaatsen en te slepen. Een leuke, laagdrempelige manier om kennis te maken met het vak.

Start MakeceCode als volgt:

1. Ga naar makecode.microbit.org/
2. Klik op 'Nieuw project' en geef je project een naam
3. Klik op 'Aanmaken'

Je ziet het volgende scherm:



Afbeelding 3. MakeCode Beginscherm

Helemaal links (1) zie je een virtuele micro:bit. Je kunt je programma dus maken zonder echte micro:bit te gebruiken (maar dat is wel minder leuk). Naast de virtuele micro:bit zie je een menu (2). Hieruit kun je blokken kiezen om je programma mee te gaan bouwen. In het rechterveld (3) zie je al twee van deze blokken staan (bij opstarten en de hele tijd).

Het geeft niets als nu nog niet direct helder is wat je met de blokken kunt doen, dat wordt in de volgende stappen duidelijk.

2.4. Werken met blokken

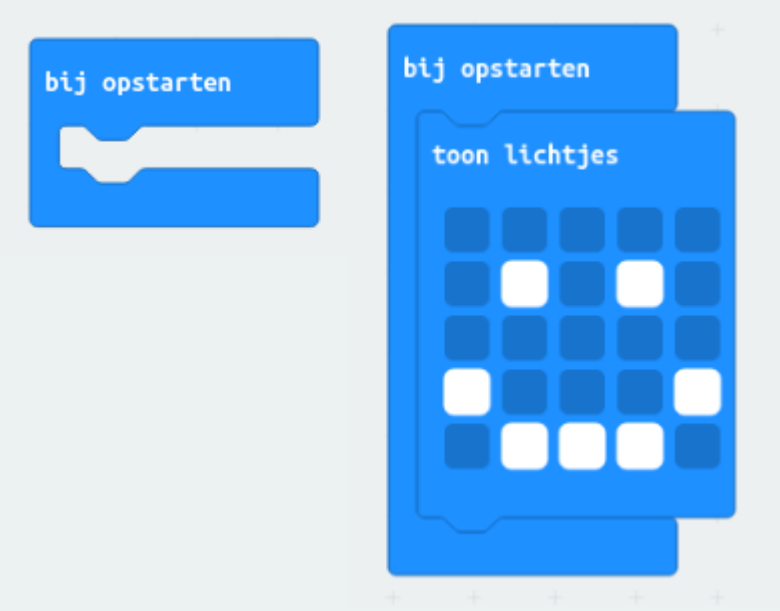
2.4.1. Als dit, dan dat

Veel van programmeren kan terug worden gebracht tot 'Als dit, dan dat'.

> **Als** ik de micro:bit schud, laat **dan** een steen, papier of schaar zien.

In *MakeCode* doe je dit veelal met de blokken die andere blokken kunnen omvatten. Neem bijvoorbeeld het 'Bij opstarten' blok in onderstaand figuur. Links is het blok zonder code. Rechts hebben we de code 'toon lichtjes' toegevoegd (het laat een smiley zien).

> **Als** de micro:bit opstart, toon **dan** een smiley met oplichtende LED-lampjes



Afbeelding 4. Een 'Bij opstarten' blok - met en zonder code

2.4.2. Variabelen

Een ander veelvoorkomend concept in programmeren is de **variabele**. Een variabele is de naam die je geeft aan waarden in je programma, zodat je ze later makkelijk kunt (her)gebruiken. Stel dat je de waardes `10` en `Hallo` wilt opslaan om later te gebruiken:

```
x = 10
y = "Hallo"
```

Later in je programma kun je nu `x` en `y` gebruiken, bijvoorbeeld om `x` op het scherm te tonen:

```
print(x)
>> 10
```

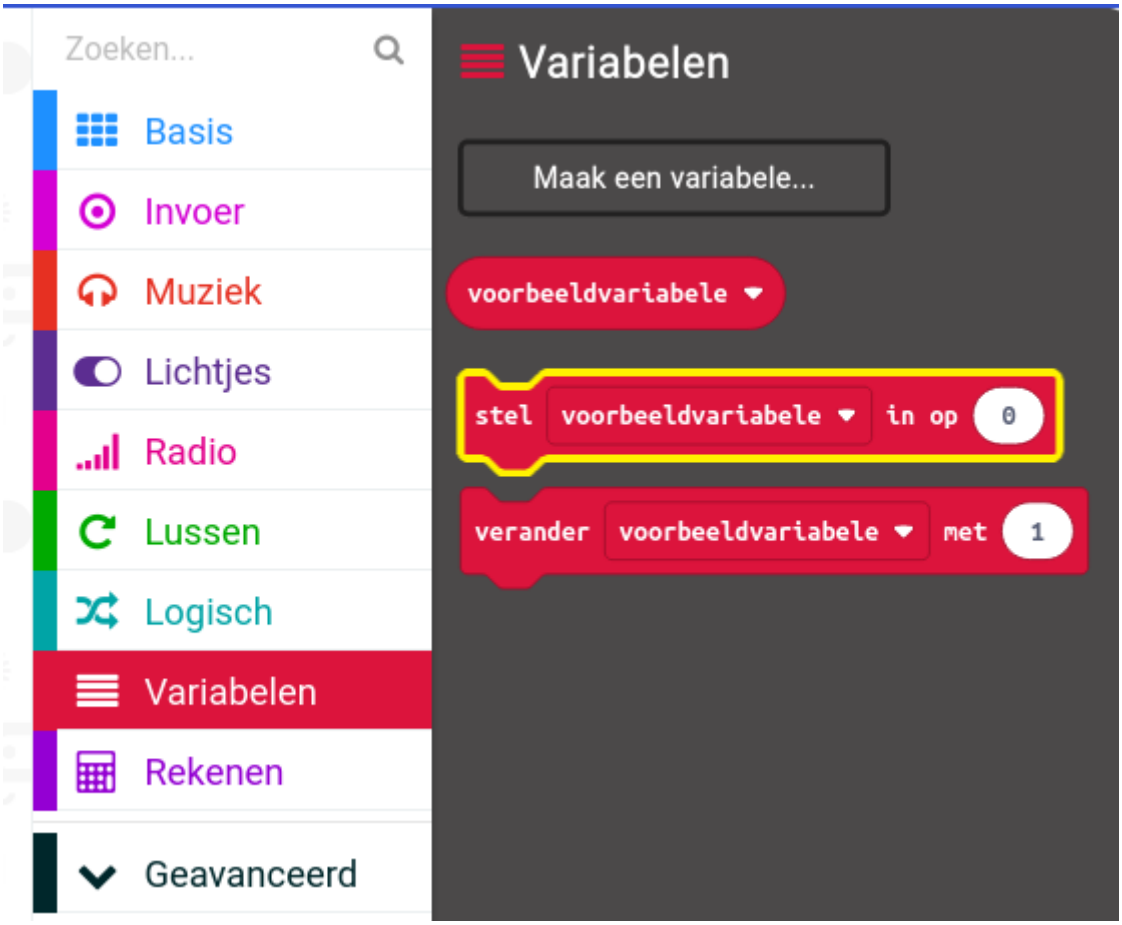
Waarom zou je niet gewoon `print(10)` gebruiken? Omdat je variabelen meestal gebruikt om een bepaald resultaat op te slaan die je later wilt gebruiken, een resultaat die per keer kan verschillen. Bijvoorbeeld:

```
# Maak een lijst die start bij 1 en stopt bij een willekeurig getal tussen 5 en 15
lijst = [1, randint(5, 15)]

# Tel alle getallen in de lijst op
resultaat = sum(lijst)

# Toon het resultaat
print(resultaat)
>> 12  # Of 17, of 20, of ...
```

In *MakeCode* kun je variabelen aanmaken via het menu:



Afbeelding 5. De variabele `voorbeeldvariabele`

In de uitwerking van de opdracht zul je leren hoe je een variabele kunt gebruiken.

2.4.3. LED-lampjes

De micro:bit heeft 25 LED-lampjes die allemaal onafhankelijk van elkaar aan en uit kunnen. Op deze manier kun je allerlei figuurtjes tonen — in ons geval een steen, papier of schaar.

In *MakeCode* zijn er drie blokken waarmee je dit eenvoudig doet, te vinden in het menu `BASIS`. Het eerste heet `Toon lichtjes`, en het is een blok dat je gebruikt in een ander blok (in `Bij schudden` bijvoorbeeld). Met je muis kun je eenvoudig figuurtjes tekenen. Een voorbeeld hiervan zag je al in het 'bij-opstarten' blok. Daarnaast heb je `Toon pictogram`, wat hetzelfde doet als `Toon lichtjes` maar met vooringestelde icoontjes. Tot slot heb je `Toon tekens`, waarmee je woorden kunt laten 'scrollen'.

Programmeren omvat uiteraard meer dan als-dit-dan-dat, variabelen en LED-lampjes, maar voor ons spel zijn dit alle basisingrediënten! Zoals je in Wat is programmeren kon lezen, omvat programmeren meerdere stappen. Een belangrijke stap is het uitdenken van je code.

2.5. Opdracht

Probeer de stappen uit te denken die nodig zijn om het Rock-Paper-Scissors spel te maken. Welke blokken heb je nodig? In welke volgorde? Klik eens door alle beschikbare blokken in het menu om een indruk te krijgen van de mogelijkheden.

Probeer de stappen eerst op papier uit te denken. Denk je visueel, dan helpt het

> misschien om te schetsen. Denk je abstract, dan kun je wellicht een stroomschema maken of het in woorden uitwerken.

Probeer hierna op makecode.microbit.org het spel te programmeren met behulp van de blokken.

2.6. Uitwerking

In onderstaande video neem ik je stap voor stap mee door de uitwerking. Ook op microbit.org vind je een uitwerking.

Afbeelding 6. Uitwerking Rock-Paper-Scissors met blokken (klik om video te openen)

Chapter 3. Programmeren met Python

Als het goed is kun je nu *Rock-Paper-Scissors* spelen op je micro:bit. Je hebt dit geprogrammeerd met behulp van een visuele programmeertaal. Dit heet ook wel no code of low code programmeren, en is iets dat de laatste jaren steeds meer in opkomst is. Het gebruik van no/low code oplossingen zie je met name in bedrijfsprocessen waar werknemers zelf zaken willen (en kunnen) automatiseren.

Veel applicaties zijn echter nog te ingewikkeld om goed in no/low code oplossingen te kunnen maken. Daarom zul je als programmeur vooral code schrijven. Dit code schrijven kan in één van de vele beschikbare programmeertalen. In dit hoofdstuk gebruiken we Python, een eenvoudige maar krachtige programmeertaal.

Op makecode.microbit.org is Python ook beschikbaar als taal. Bovenin staat standaard `Blokken` geselecteerd. Hier kun je ook kiezen voor `Javascript` (een andere programmeertaal) of `Python`.

Afbeelding 7. Je kunt ook programmeren in Python in MakeCode

3.1. Werken met Python

In Werken met blokken las je al dat programmeren vaak teruggebracht kan worden tot 'Als dit, dan dat'. In de uitwerking van de vorige opdracht heb je dit ook gedaan met het blok `Als, anders als, anders`.

In Python doe je dit met de woorden `if`, `elif` (van else if) en `else`. Bijvoorbeeld:

```
if hand == 1:
    print("Steen")
elif hand == 2:
    print("Papier")
else:
    print("Schaar")
```

Let op dat inspringen betekenis heeft in Python. Alles wat met 4 spaties is ingesprongen onder `if hand == 1:` zal worden uitgevoerd als de variabel `hand` gelijk is aan 1.

Omdat we in *MakeCode* werken hebben we ook toegang tot functionaliteit die speciaal voor de micro:bit bedoeld is. Als je Python op computer zou gebruiken, dan zou je deze functionaliteit eerst moeten installeren. De functionaliteit om de LED-lampjes een icoontje te laten tonen werkt bijvoorbeeld zo:

```
basic.show_icon(IconNames.SMALL_SQUARE)
```

Dit lijkt erg op het blok `Toon icoon` (Show icon) uit het menu `Basis` (Basic) zoals je die eerder gebruikt hebt.

3.2. Opdracht

In *MakeCode* maken we het spel *Rock-Paper-Scissors* nog een keer, maar nu in Python. Ga naar makecode.microbit.org, start een nieuw project en kies in plaats van `Blokken` voor `Python`.

Het voert te ver om de basis van Python te leren. Daarom krijg je hier de benodigde puzzelstukjes waarmee je zelf het spel kunt programmeren.

Variabelen

```
x = 1
```

Willekeurig getal

```
x = randint(<beginwaarde>, <eindwaarde>)
```

Als, dan

```
if x == <waarde>:
    <voer dit uit>
elif x == <waarde>:
    <voer dit uit>
else:
    <voer dit uit>
```

Bij schudden

```
input.on_gesture(Gesture.SHAKE, on_gesture_shake)
```

`on_gesture_shake` is een zogenaamde **functie** die uitgevoerd wordt als de micro:bit wordt geschud.

Een functie maken

```
def on_gesture_shake():
    <schrijf hier wat de functie moet doen>
```

Met deze puzzelstukjes kun je nu het spel *Rock-Paper-Scissors* maken.

3.3. Uitwerking

In onderstaande video neem ik je stap voor stap mee door de uitwerking. Ook op microbit.org vind je een uitwerking. Klik dan steeds op `Python` om de uitwerking in Python te zien.

Afbeelding 8. Uitwerking Rock-Paper-Scissors met Python (klik om video te openen)

Chapter 4. Hoe verder?

In dit boek heb je een klein beetje kunnen proeven wat het vak programmeren inhoudt. Je hebt geleerd wat een programmeur zoal doet, wat zijn of haar profiel is én je hebt je eerste spelletje op de micro:bit geprogrammeerd.

Maar hoe nu verder? Er zijn een paar mogelijkheden. Je concludeert:

- Programmeren is niets voor mij.
- Interessant, maar ik zou eerst nog meer willen weten.
- Ineens weet je het: ik word programmeur!

Als je nu concludeert dat programmeren niets voor jou is: even goede vrienden! Misschien vind je het nog wel leuk om wat kleine projectjes met je micro:bit op te zetten, dan moedig ik je vooral aan om op microbit.org rond te kijken.

4.1. Verder onderzoeken

Met dit boek heb ik geprobeerd om je te laten ervaren wat programmeren is, maar ik realiseer me ook ter dege dat het slechts een (klein) topje van de ijsberg is. Je hebt nog niet ervaren hoe het is om in een team te werken, een echte applicatie te maken, om te gaan met verzoeken van klanten, et cetera. Is je interesse wel gewekt, dan zijn er enkele logische vervolgstappen.

micro:bit

Op microbit.org staan vele projecten die je kunt uitvoeren met je micro:bit. Ik moedig je aan om er zelf een paar te proberen, en probeer zowel met de blokken als met Python te werken. Wil je de functionaliteit van de micro:bit uitbreiden, kijk dan bijvoorbeeld eens op microbit.store voor de mogelijkheden.

Snuffelstage

Om te ervaren hoe het is om als programmeur te werken, kan het leuk en leerzaam zijn om een dag met een programmeur mee te lopen.

Een programmeertaal leren

Wil je eerst meer ervaren hoe het is om te programmeren *an sich*, dan is een beginnerscursus programmeren een goede optie. Ik raad je aan om met Python te beginnen: dit is een toegankelijke, veelzijdige taal en biedt daarmee veel kansen op de arbeidsmarkt. Python wordt onder andere gebruikt voor:

- Het ontwikkelen van webapplicaties en API's
- Data science, data analyse, Artificial Intelligence, Machine Learning
- Scripting / automatisering en systeembeheer

Er zijn online voldoende beginnerscursussen Python te vinden. Is zelfstudie meer iets voor jou? Dan kan ik het boek Python Crash Course aanbevelen.

4.2. Tot slot

Hoe je ook besluit verder te gaan: ik hoop dat je in elk geval plezier aan dit boek hebt beleefd! Mocht je nog vragen of opmerkingen hebben, neem dan gerust contact met me op:

Erwin Matijsen
LinkedIN