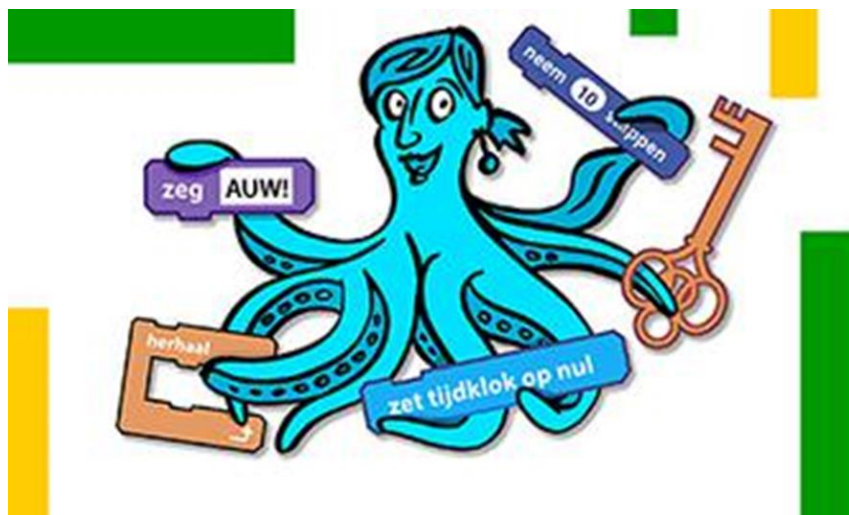


DelftX: ScratchX Docent-ouder handleiding

Scratch: Programmeren voor kinderen (8+)

MOOC – start datum 05.09.2018



Deze online cursus is gemaakt door:
Dr. Ir. Felienne Hermans

Inhoud

| | |
|--------------------------------------|----|
| 1. Introductie | 2 |
| 1.1 Hoe gebruik ik Scratch? | 2 |
| 1.2 edX..... | 3 |
| 1.2.1 Een account maken | 3 |
| 1.2.2 Werken met het materiaal | 3 |
| 1.2.3 Online discussie..... | 3 |
| 2. Scratch..... | 3 |
| 2.1 Beginnen met Scratch..... | 4 |
| 2.2 Zet de taal op Nederlands..... | 4 |
| Sprites..... | 5 |
| Scripts | 6 |
| Programmeren | 7 |
| Variabelen | 7 |
| Als-dan-(anders)..... | 8 |
| Lussen..... | 9 |
| Events | 9 |
| Signalen..... | 10 |
| Klonen..... | 10 |
| Het ScratchX lesmateriaal..... | 10 |
| Les 1: Giga | 10 |
| Les 2: Diepzeevissen..... | 11 |
| Les 3: Een spelshow..... | 12 |
| Les 4: Oversteken..... | 12 |

1. Introductie

Leuk dat je mee gaat doen aan ScratchX van de TU Delft!

In deze handleiding vind je van alles wat nodig is om de cursus tot een goed eind te brengen met je kinderen, neefjes, nichtjes, buurjongens of meisjes of leerlingen. Vanaf nu schrijf ik voor het gemak leerlingen, maar ik bedoel natuurlijk welk kind dan ook!

We leggen eerst uit hoe je dit lesmateriaal gebruikt. Dan volgt uitleg over edX, het online platform dat we gebruiken. Daarna gaan we het over programmeren met (en zonder) Scratch hebben, en het laatste hoofdstuk bevat alle informatie over de lessen in het lesmateriaal.

1.1 Hoe gebruik ik Scratch?

Het materiaal van ScratchX is ontwikkeld door de TU Delft, voor kinderen. De video's en puzzels moeten goed te volgen zijn voor kinderen vanaf 8 jaar.

De video's zijn kort, en dagen kinderen uit om zelf dingen te proberen. Na een quiz legt het volgende filmpje weer uit hoe ze verder moeten. Bij iedere puzzel krijgen kinderen ook uitleg over wat het goede antwoord is, en ze verwijzen ze waar nodig terug naar eerder lesmateriaal.

Eigenlijk hoef jij als leraar of ouder dus helemaal niet zoveel te doen. Wat wel nodig is, is het volgende:

- Maak voor iedere leerling een account aan edX, 'enroll' iedere leerling voor deze cursus. edX is niet gemaakt voor kinderen is niet altijd even intuïtief (zie verder edX)
- Installeer Scratch of ga naar scratch.mit.edu (zie verder Scratch)

Wat wel handig is, maar niet nodig:

- Speel zelf een beetje met Scratch. Je kunt hiervoor een paar van onze filmpjes kijken, of experimenteren op scratch.mit.edu
- Maak jezelf bekend met een paar programmeerconcepten (zie verder Programmeren)
- Lees het lesoverzicht door zodat je weet wat er komt (zie verder ScratchX)

Let op! Dit is een bericht voor ouders of leraren die zelf al goed kunnen programmeren. Dit lesprogramma is speciaal ontwikkeld om kinderen niet alleen te leren programmeren, maar ze ook spelenderwijs te laten reflecteren op netjes programmeren. In sommige filmpjes en opdrachten programmeren met dus expres niet al te goed. Grijp dan als volwassene bij voorkeur niet in, want dat gaan we in latere video's of opdrachten (de kinderen laten) oplossen.

Naast de video's zijn er ook lesboekjes voor iedere week, die kun je in plaats van de video's gebruiken. Er staan vergelijkbare opdrachten in de boekjes, maar, je krijgt dan als leerling natuurlijk geen feedback zoals in de online omgeving. Als je daarvoor kiest, zul je dus als leraar of ouder wel wat meer ondersteuning moeten geven.

1.2 edX

Voor ScratchX gebruiken wij edX, een online leerplatform. De TU Delft heeft een strategische overeenkomst gesloten met edX, dat een initiatief is van de Amerikaanse universiteiten Harvard en MIT.

edX is eigenlijk gericht op volwassen 'learners' zoals zij dat noemen. Dat met kinderen gebruiken is voor ons een experiment. Dus als je feedback hebt, is dat van harte welkom!

Er zijn een aantal redenen dat we edX gebruiken, en niet, bijvoorbeeld, een boek hebben geschreven of een set filmpjes op YouTube hebben gezet.

Onze ervaring met programmeeronderwijs is dat het soms erg lastig is om te weten wat kinderen snappen. Met een beetje prutsen krijgen ze wel wat blokjes werkend, maar snappen ze echt de concepten? Daarom worden onze filmpjes afgewisseld met quizen en puzzels. Zo weten kinderen zelf wat ze snappen, en jij als docent ook. Als blijkt dat een concept nog niet klikt bij een leerling, dan krijgen ze een linje terug naar eerdere videos, of naar een extra puzzel.

Om die reden vragen we jullie ook om voor iedere leerling een eigen account aan te maken. Zo krijgt ieder kind les op maat!

Onze lessen maken deel uit van een langdurig plan van de TU Delft om onderzoek te doen naar programmeren voor kinderen. Ook daarom gebruiken we edX, want wij kunnen zien welke quizen en goed en slecht gemaakt worden. Daar leren we van! Welk concept snappen kinderen meteen? Wat is er lastiger? In welke puzzels zien we grote spreiding?

We bekijken die data geaggregeerd, en publiceren natuurlijk geen data over jouw leerlingen.

1.2.1 Een account maken

Voor je je kan inschrijven, moet je een account aanmaken op edX. Zoals hierboven uitgelegd hebben we het liefst dat je voor iedere leerling een eigen account aanmaakt.

1.2.2 Werken met het materiaal

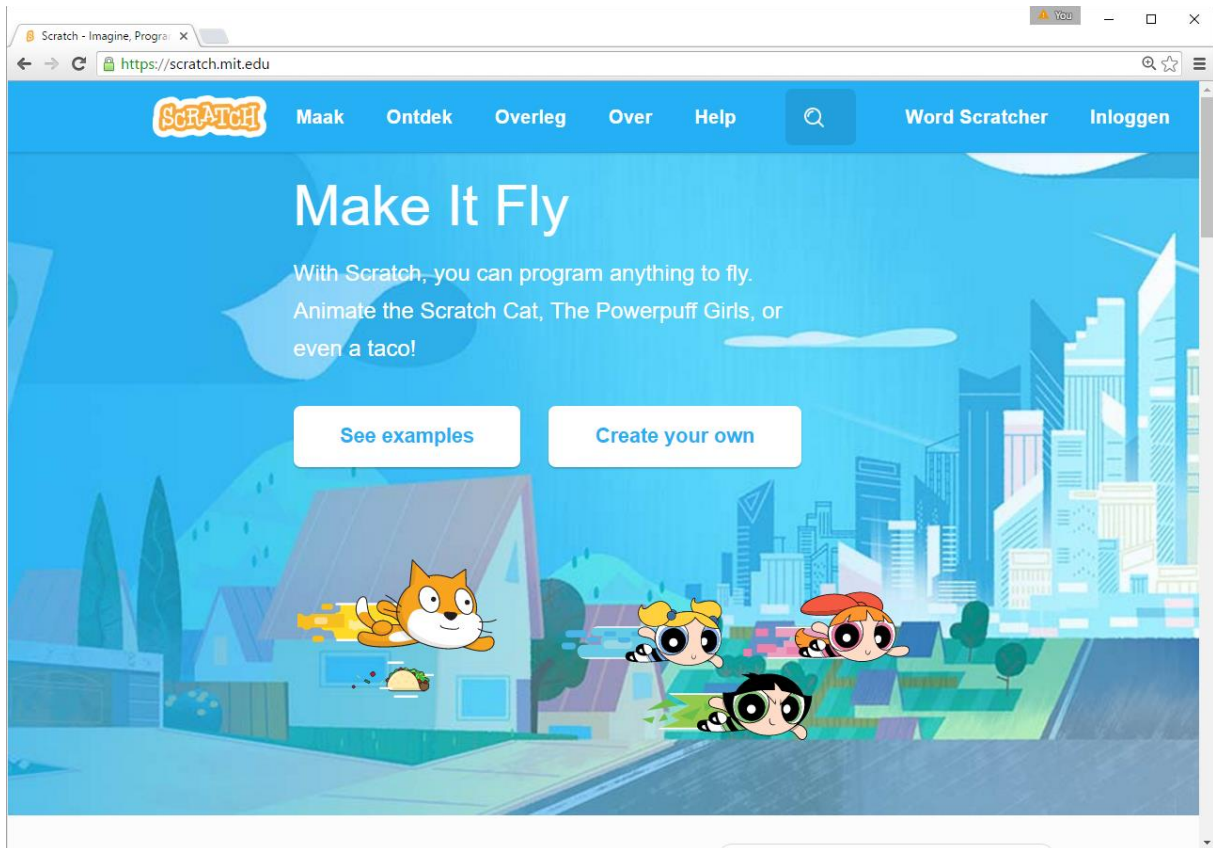
Het materiaal van ScratchX bestaat uit video's en quizen.

1.2.3 Online discussie

Als jij of je leerlingen een vraag hebben, kun je ons via het discussieforum een vraag stellen.

2. Scratch

Scratch, gemaakt door MIT, is een programmeertaal die speciaal gemaakt is voor kinderen. Kinderen kunnen er makkelijk spelletjes en animaties mee maken. De interface is kleurrijk en nodigt kinderen uit tot zelf dingen proberen.



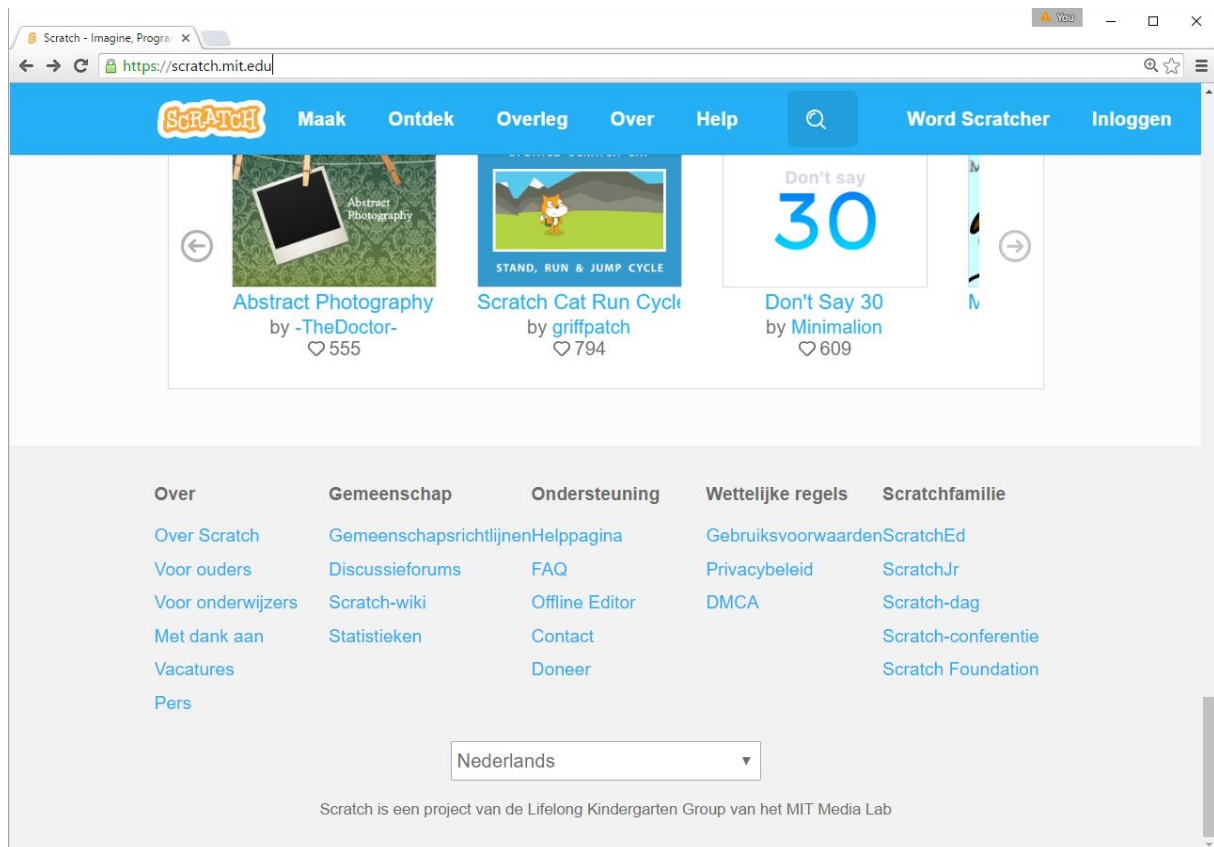
2.1 Beginnen met Scratch

Je hoeft niks te installeren om met Scratch te beginnen, je kunt direct in de browser spelen. Er is wel een offline app, voor als de laptop van je leerling geen internet heeft. Je kunt de offline versie downloaden via: <https://scratch.mit.edu/scratch2download/>

In de filmpjes gebruiken wij de versie in de browser. Daar kom je door op de 'Maak' knop links bovenin te klikken. Er staat voor het gemak een pijl bij in het plaatje hierboven. Die staat natuurlijk niet echt op de website :-)

2.2 Zet de taal op Nederlands

Als de Scratch site niet automatisch op Nederlands is overgeschakeld, dan kun je dit handmatig aanpassen. Scroll daarvoor helemaal naar beneden en kies Nederlands uit de dropdownlijst.

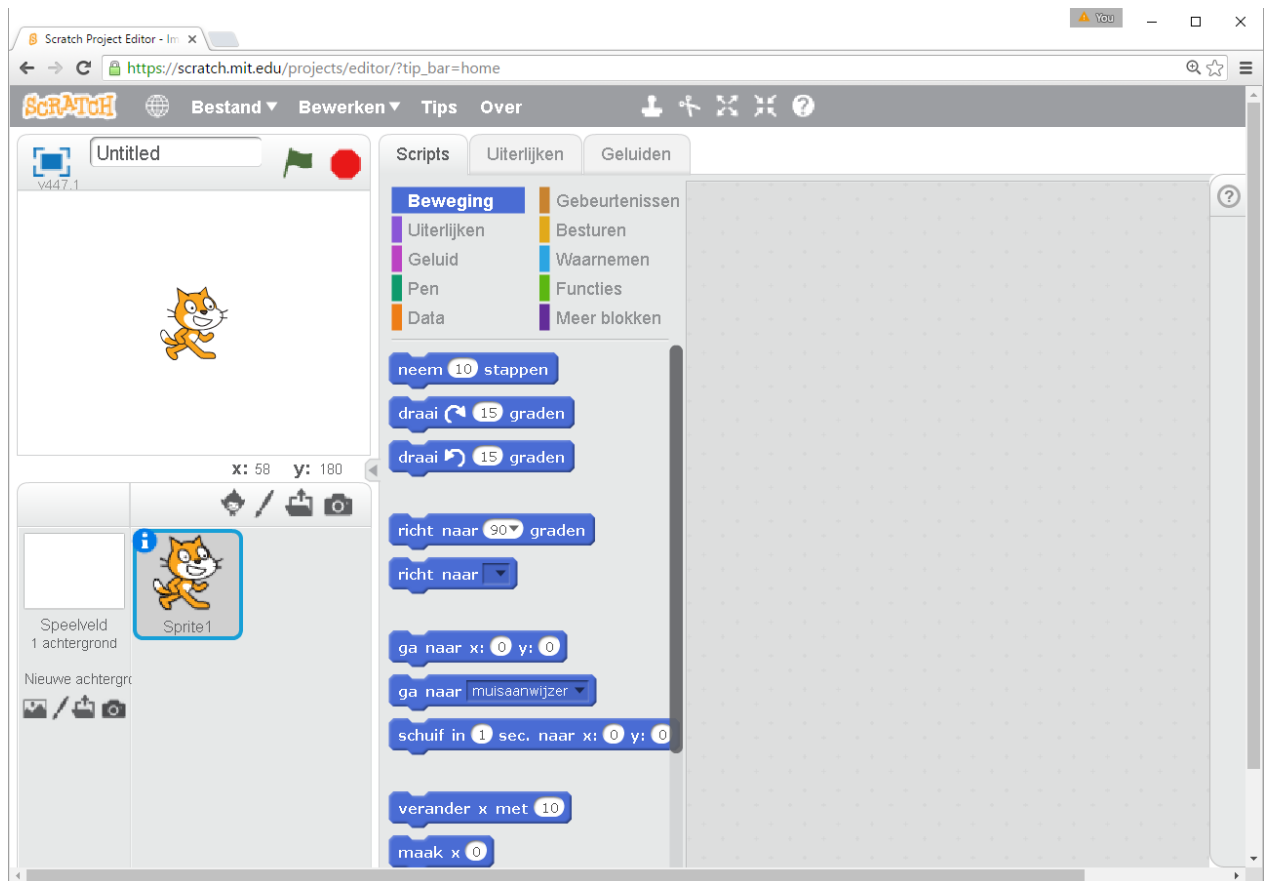


We leggen nu een aantal Scratch concepten uit die je tegen gaat komen in het lesmateriaal.

Sprites

Programmeren in Scratch doe je binnen 'sprites', dat zijn poppetjes in het spel. In de videos gebruik ik zowel het woord 'sprite' als poppetje. In de omgeving staat ook in de Nederlands versie sprite, vandaar dat ik dat woord soms ook gebruik.

Als je op 'Maak' hebt geklikt krijg je een nieuw spel, met vanzelf het eerste poppetje: de vrolijke kat die ook het logo van Scratch is.

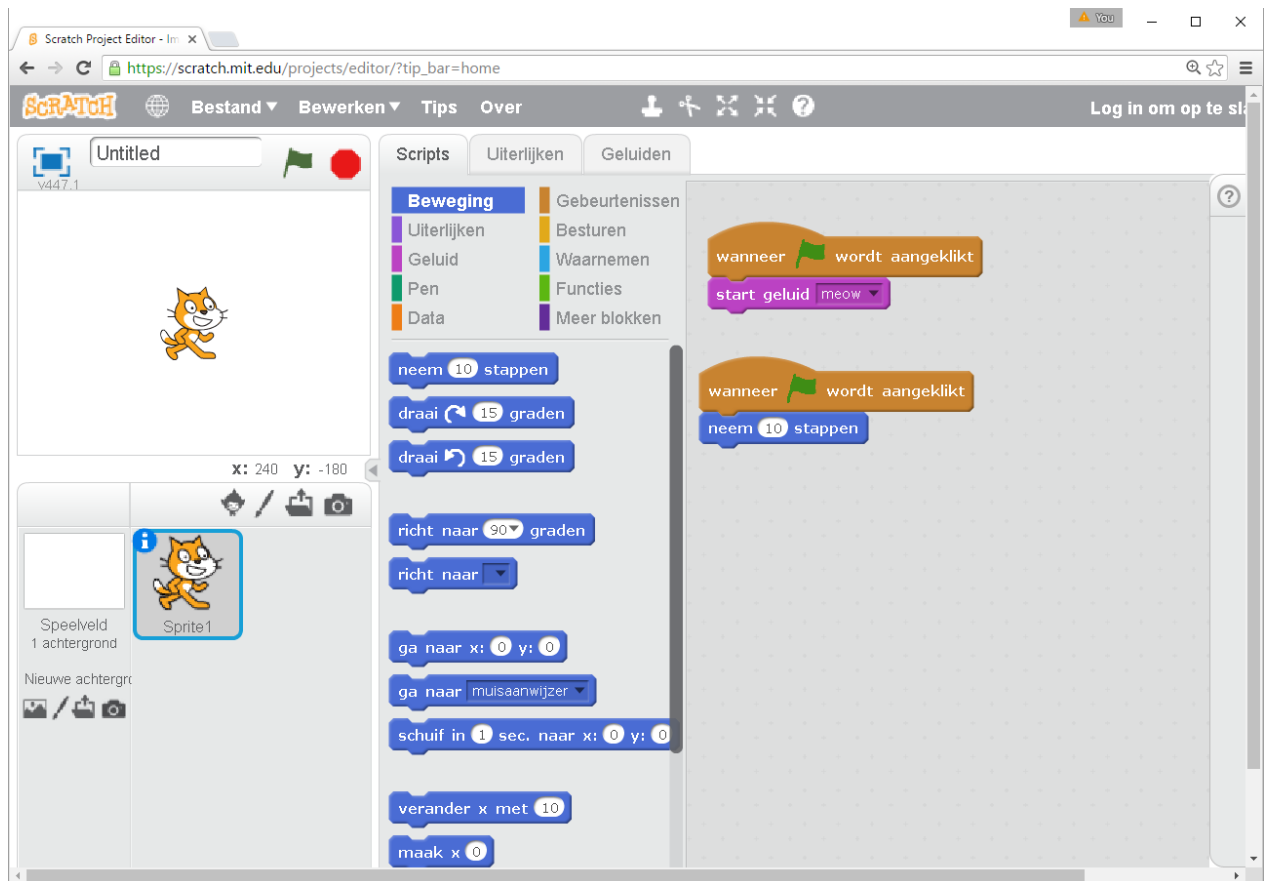


Een spel heel veel 'sprites' hebben. Scratch heeft een uitgebreide bibliotheek, maar leerlingen kunnen ook zelf plaatjes uploaden om ze tot leven te laten komen. Dit doen we ook voor in de filmpjes.

Je kunt ook de achtergrond (het speelveld) veranderen, die heeft net iets andere programmeerblokjes dan de sprites. Een achtergrond kun je bijvoorbeeld niet laten bewegen.

Scripts

Iedere sprite kan meerdere stukjes programma hebben, die noemen we scripts. Bijvoorbeeld: een script het maken van een geluid, en eentje voor lopen, zoals in het plaatje hieronder. Alle scripts lopen 'parallel': ze worden tegelijk uitgevoerd.



Programmeren

Misschien vraag je je als docent of ouder af wat programmeren eigenlijk is. Je leerlingen gaan dat misschien ook wel vragen. Er zijn veel antwoorden op die vraag.

De simpelste is denk ik dat programmeren is: een computer laten doen wat je wilt. En helaas spreken computer (nog) geen mensentaal, dus moeten we daar een computertaal voor gebruiken.

Scratch is een programmeertaal voor kinderen, maar dat betekent niet dat het geen krachtige taal is. Alle belangrijke programmeerconcepten die ook een "grote mensen" talen zitten, kom je in Scratch tegen. In dit hoofdstuk licht ik er een aantal voor je toe, zodat je met je leerlingen kunt meepraten. Of meedoen natuurlijk. Er is niets op tegen om nu deze handleiding weg te leggen en eerst lekker zelf de filmpjes en quizzen te gaan bekijken! Ik geeft in dit hoofdstuk voorbeelden in Scratch, maar ook in Python. Python is een taal voor professionele programmeurs, maar de taal is simpel van opzet, dus is ook erg toegankelijk voor niet-programmeurs. Er is zelfs een leuk boekje voor kinderen!

Ik geef voorbeelden beide talen, zodat je meteen ziet wat de verschillen zijn.

Variabelen

Variabelen zijn opslagplaatsen voor data. Denk bijvoorbeeld aan een bankrekening, waar een bepaald getal aan geld op kan staan, of aan een schoolbord waarop je de klassenoudste van de week kunt schrijven. Een variabele kun je schrijven en lezen.

Het simpelste voorbeeld in de Scratch lessen is het aantal punten in een spel. Steeds als je, bijvoorbeeld, een muntje pakt, krijg je er een punt bij. Het hoeft niet perse een getal te zijn, we kunnen ook tekst opslaan in een variabele.



Deze code maakt 'punten' eerst 0 en verhoogt het dan met 1.

In Python zouden we dit schrijven:

```
points = 0
points = points + 1
```

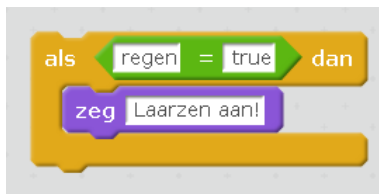
Als-dan-(anders)

Een van de centrale concepten in programmeren is de als-dan, of in het Engels if-then. Als dit, dan dat. Je kunt je leerlingen hier zeker ook zonder computer over laten nadenken. Dit gaan we trouwens ook in de puzzels doen!

Als het regent, doe ik mijn laarzen aan.

Als ik een broertje krijg, eten we beschuit met muisjes.

In Scratch gaat dat zo:



Soms wordt de als-dan gecombineerd met een anders: Als het regent, doe ik laarzen aan, anders schoenen. In het Engels heet dat de 'else': if-then-else.

In Scratch:



In Python (en in het Engels)

```
if rain:
    print('Wear boots.')
else:
    print('Wear shoes.')
```

Lussen

Een lus is een stukje programmeertaal dat andere opdrachten herhaalt. Ook in het Nederlands gebruiken we vaak de Engelse term 'loop'. In de videos noem ik het een herhaalblok. De simpelste vorm is een herhaling zonder conditie:



In Python gaat dat zo:

```
while True:  
    move(10)
```

De Scratch code zorgt ervoor dat het bijbehorende poppetje eindeloos vooruit gaat. In theorie dan, in de praktijk zal de sprite bij de rand ophouden en dan vastlopen.

Je hebt ook herhaalblokken met een conditie. Herhaal tot, bijvoorbeeld.



Hier gaat de sprite door tot deze de muur raakt en stopt dan netjes. Je ziet dat het 'herhaal tot' blok een "haakje" onderaan heeft, nadat de sprite de muur raakt kan de code daar verder gaan.

Lussen met condities kom je in de "vrije natuur" tegen, bijvoorbeeld: loop rechtboor tot je bij de grote eikenboom bent.

Python heeft geen 'herhaal tot' maar wel een 'while' een herhaal zolang. Dus dat zou er zo uitzien:

```
while touching_border <> true: ← de <> betekent niet gelijk aan  
    move(10)
```

Opmerking: de volgende concepten zijn wat Scratch specifieker dan de bovenstaande.

Events

Scratch is "event-based". Dat betekent dat Scratch programma's kunnen reageren op gebeurtenissen in de omgeving. Bijvoorbeeld: Wanneer de groene vlag wordt aangeklikt, of wanneer er op de spatiebalk gedrukt wordt. Ieder 'script' in Scratch moet altijd beginnen met een event-blok, die je kunt herkennen aan hun hoedvorm:



Signalen

Als sprites met elkaar willen communiceren, kan dat met signalen. Bijvoorbeeld als een sprite wil doorgeven aan andere speler dat hij af is.

Klonen

Als je veel poppetjes hebt met dezelfde functionaliteit, bijvoorbeeld spookjes in Pacman, dan is het niet handig om die een voor een te programmeren. Dat is veel te veel moeite! Scratch ondersteunt daarom 'klonen': daarmee kun je meerdere kopien van dezelfde sprite maken, met dezelfde logica.

Het ScratchX lesmateriaal

Er zitten zes lessen in het lesmateriaal. In iedere les maken de leerlingen met behulp van de filmpjes een eigen spel.

Dit hoofdstuk geeft een kort overzicht van de inhoud van de lessen. Je kunt voor de spellen ook de boekjes gebruiken, die volgen grotendeels de verhaallijn van de video's.

Les 1: Giga

In deze les maken leerlingen kennis met Scratch. Ze gaan een relatief simpel spel programmeren met één poppetje: Giga. Giga kan bestuurd worden met de pijltjestoetsen. Als ze de muur raakt, gaat ze terug naar de linkerbovenhoek. Ook programmeren we een spook, dat gaat proberen Giga te pakken. Als dat lukt moet ze ook terug naar de linkerbovenhoek.

De volgende tabel geeft aan welke Scratch blokken en concepten in deze les worden behandeld. De dikgedrukte concepten komen uit de leerlijn programmeeronderwijs. Ik geef hier de interne Engelse namen, zodat je ze makkelijk op kunt zoeken op de Scratch wiki:

[https://wiki.scratch.mit.edu/wiki/Scratch_File_Format_\(2.0\)/Block_Selectors](https://wiki.scratch.mit.edu/wiki/Scratch_File_Format_(2.0)/Block_Selectors)

| Concept | Blok | Properties | |
|--------------------|-------------------------------|------------|---------|
| | | Category | Shape |
| Events | whenGreenFlag | Events | Hat |
| Acties | forward: | Motion | Stack |
| Acties | changeXposBy: | Motion | Stack |
| Acties | changeYposBy: | Motion | Stack |
| Events | whenKeyPressed | Events | Hat |
| Herhaling | doForever | Control | C |
| Wachten | wait:elapsed:from: | Control | Stack |
| Voorwaarden | dolf | Control | C |
| Voorwaarden | touchingColor: | Sensing | Boolean |
| Acties | say: | Looks | Stack |
| Herhaling | dolfElse | Control | C |
| Voorwaarden | touching: | Sensing | Boolean |
| Acties | glideSecs:toX:y:elapsed:from: | Motion | Stack |

| | | | |
|------------------|---------------|-----------|----------|
| Acties | bounceOffEdge | Motion | Stack |
| Logica | | Operators | Boolean |
| Acties | pointTowards: | Motion | Stack |
| Variabele | setVar:to: | Data | Stack |
| Variabele | readVariable | Data | Reporter |
| Variabele | changeVar:by: | Data | Stack |

Les 2: Diepzeevissen

In les 2 maken we een vissenspel. Visje Gerrie eet kleine visjes om te groeien, maar moet een giftige vis ontwijken die hem graag lust. Qua programmeren herhalen we een aantal concepten uit les 1. De herhaal en als-dan komen terug, want daar kun je niet omheen als programmeur.

Verder maken we in deze les kennis met drie verschillende concepten. Ter eerste: “random” getallen. Dat zijn getallen die steeds weer willekeurig door de computer worden gekozen. Zo kun je je spel steeds anders maken, en is het zelfs voor jou een verrassing wat er gaat gebeuren.

Ook gaan we in deze les ‘signalen’ gebruiken. Met signalen kunnen sprites elkaar berichten doorgeven. Denk hierbij aan “buut vrij” roepen bij verstoppertje. Iedereen hoort het signaal en weet dat hij of zij weer vrij is.

Tenslotte gaan we eigen blokken maken in les 2. Soms doe je vaak hetzelfde en dat is helemaal niet handig. Dan kun je blokken bij elkaar in een eigen blok stoppen. Een soort ‘mapje’ op de computer met opdrachten erin die steeds worden uitgevoerd.

| Concept | Blok | Properties | |
|--------------------|--------------------|------------|----------|
| Uiterlijk | lookLike: | Looks | Stack |
| Voorwaarden | dolf | Control | C |
| Sensoren | touching: | Sensing | Boolean |
| Uiterlijk | hide | Looks | Stack |
| Wachten | wait:elapsed:from: | Control | Stack |
| Uiterlijk | show | Looks | Stack |
| Acties | gotoX:y: | Motion | Stack |
| Random getallen | randomFrom:to: | Operators | Reporter |
| Voorwaarden | touching: | Sensing | Boolean |
| Communicatie | broadcast: | Events | Stack |
| Communicatie | whenIReceive | Events | Hat |
| Acties | say: | Looks | Stack |
| Uiterlijk | changeSizeBy: | Looks | Stack |
| Uiterlijk | setSizeTo: | Looks | Stack |
| Events | whenGreenFlag | Events | Hat |
| Herhaling | doForever | Control | C |
| Beweging | forward: | Motion | Stack |
| Beweging | bounceOffEdge | Motion | Stack |
| Uiterlijk | setSizeTo: | Looks | Stack |
| Events | whenGreenFlag | Events | Hat |
| Herhaling | doForever | Control | C |

| | | | |
|-------------------|---------------|-------------|-------|
| Beweging | pointTowards: | Motion | Stack |
| Beweging | forward: | Motion | Stack |
| Beweging | bounceOffEdge | Motion | Stack |
| Abstractie | procdef: | More blocks | Hat |
| Functie | procdef: | More blocks | Hat |
| Patronen | call | More blocks | Stack |

Les 3: Een spelshow

In les 3 maken we een quiz. Je draait aan een rad dat op een bepaalde kleur terecht komt. Afhankelijk van de kleur krijgt de deelnemers 1 of 2 punten voor een vraag. De vragen mogen de leerlingen zelf verzinnen.

De les introduceert het idee dat sprites kunnen draaien, en herhaalt herhaalblokken, variabelen en als-dan-anders.

Het belangrijkste concept dat achter deze les zit is, is het concept 'lijsten' Lijsten komen heel veel voor in programmeren. Je kunt in een lijst dingen opslaan, een bepaald item uit de lijst halen of een hele lijst wissen. Dat oefenen we met de vragen.

| Concept | Blok | Properties | |
|----------------------|--------------------|------------|----------|
| | | | |
| | keyPressed: | Sensing | Boolean |
| Beweging | turnRight: | Motion | Stack |
| Beweging | turnLeft: | Motion | Stack |
| Variabele | setVar:to: | Data | Stack |
| Variabele | readVariable | Data | Reporter |
| Variabele | changeVar:by: | Data | Stack |
| Herhaling | doUntil | Control | C |
| Voorwaarden | = | Operators | Boolean |
| Uiterlijk | say: | Looks | Stack |
| Input | doAsk | Sensing | Stack |
| String functions | concatenate:with: | Operators | Reporter |
| Voorwaarden | doIfElse | Control | C |
| Representatie | append:toList: | Data | Stack |
| Events | whenGreenFlag | Events | Hat |
| Representatie | deleteLine:ofList: | Data | Stack |
| Representatie | getLine:ofList: | Data | Reporter |
| Random getallen | randomFrom:to: | Operators | Reporter |
| Fouten | lineCountOfList: | Data | Reporter |

Les 4: Oversteken

In les 4 leren we kinderen over algoritmes. Een algoritme is een soort recept, dat de computer kan uitvoeren. Een recept zelf is al een goed echt voorbeeld! Doe 3 eieren in een kom, voeg 250 milliliter

melk toe, en roer, is zeker een algoritme. Meestal zit in een algoritme ook condities en herhalingen, bijvoorbeeld bij oversteken, waar deze week over gaat.

Kijk links, kijk rechts, kijk links

Als er geen auto komt, steek dan over

Als er wel een auto komt, herhaal dit algoritme

Behalve over algoritmes, maken we deze weer ook kennis met het wacht tot blok. Dat lijkt of herhaal tot, maar bij wacht tot gebeurt er verder niks, de sprite wacht alleen.

| Herhaling | doForever | Control | | C |
|-------------------------|-------------------------------|----------------|----------|---|
| Herhaling | doUntil | Control | C | |
| Acties | xpos | Motion | Reporter | |
| Voorwaarden | > | Operators | Boolean | |
| Algoritmes | distanceTo: | Sensing | Reporter | |
| String functions | concatenate:with: | Operators | Reporter | |
| Functie | rounded | Operators | Reporter | |
| Acties | glideSecs:toX:y:elapsed:from: | Motion | Stack | |
| Acties | say: | Looks | Stack | |
| Herhaling | doWaitUntil | Control | Stack | |
| Acties | say: | Looks | Stack | |
| Uiterlijk | show | Looks | Stack | |
| Voorwaarden | touching: | Sensing | Boolean | |
| Representatie | getAttribute:of: | Sensing | Reporter | |
| Abstractie | procdef: | More blocks | Hat | |
| Functie | call | More blocks | Stack | |