



PROGRAMMEREN MOET JE LEREN

Programmeren is het schrijven van een programma met instructies voor een computer. Dat klinkt misschien ingewikkeld, maar dat hoeft het helemaal niet te zijn.

Want instructies geven doe je de hele dag door. Aan andere mensen, bijvoorbeeld als je aan iemand uitlegt hoe je die lekkere cupcakes bakt. Maar ook aan computers, bijvoorbeeld als je de oven instelt. In feite ben je in beide gevallen aan het programmeren.

In deze module gaan we aan de slag met instructies voor een computer aan de hand van een alledaags en menselijk voorbeeld. We leggen namelijk aan Olaf de Robot uit hoe je een oer-Hollandse boterham met hagelslag smeert.

Zo leer je de belangrijkste verschillen én overeenkomsten tussen het geven van instructies aan een persoon versus een computer. Ondertussen maak je kennis met de vijf belangrijkste begrippen die komen kijken bij programmeren. Net zoals het handig is om wegwijs te zijn in de keuken voordat je leert koken.

PROGRAMMEERTAAL

TAAL DIE GEBRUIKT WORDT OM INSTRUCTIES DOOR TE GEVEN AAN EEN COMPUTER

ALGORITME

EEN REEKS OPEENVOLGENDE INSTRUCTIES



DEBUGGEN

HET OPSPOREN EN OPLOSSEN VAN FOUTEN
IN EEN COMPUTERPROGRAMMA

STATEMENT

EEN EENVOUDIGE INSTRUCTIE VOOR EEN
COMPUTERPROGRAMMA OM EEN ACTIE UIT TE
VOEREN

RUNNEN

HET UITVOEREN VAN EEN COMPUTERPROGRAMMA



Stel je eens voor: op vakantie in Barcelona heb je afgelopen zomer Maria leren kennen. Jullie raakten meteen bevriend. Nu is ze naar Nederland gekomen en stuurt je een bericht in het Spaans. Je begrijpt er niks van, maar als je de tekst voorlegt aan de robot van Google Translate, krijg je de volgende vertaling:



"Hoe maak je toch zo'n lekkere boterham met hagelslag? Je vertelde er zo enthousiast over toen je zo'n heimwee had! Ik wil er graag één proeven, maar ik weet het recept niet. De ingrediënten heb ik al wel in huis en ik sta nu aan het aanrecht. Wil je me helpen?"

Natuurlijk wil je Maria graag kennis laten maken met deze Nederlandse lekkernij en je stuurt haar het recept.

HOE SMEER JE EEN BOTERHAM MET HAGELSLAG?

Schrijf op je kladblok het recept voor een boterham met hagelslag. Op het aanrecht liggen de volgende benodigdheden:

- een zak brood
- een pak hagelslag
- een pakje boter
- een bord
- een mes

Wat zijn de stappen die Maria moet volgen?

Je mag er van uit gaan dat Maria zelf de Nederlandse tekst voorlegt aan de robot van Google Translate die alles naar het Spaans vertaalt ;-)

Realiseer je je dat je net een programma hebt gemaakt voor een mens?

Dit recept schreven we in het Nederlands en Maria las het in het Spaans. Het maakt niet uit in welke menselijke taal we communiceren. Want ons brein en dat van Maria werken op precies dezelfde manier, met stroompjes tussen hersencellen.

Als het om computers gaat, spreken we over programmeertalen. Het brein van een robot bestaat uit bits en bytes.

Om een simpel gesprekje met Maria te voeren, hoef je niet te weten hoe haar hersenen werken. Hetzelfde geldt voor computers, ook daar kunnen tegenwoordig steeds eenvoudiger mee communiceren.

Maar een robot heeft minder kennis van de wereld om ons heen. Daarom moeten onze instructies extra duidelijk zijn. Wat we daarmee bedoelen, ontdek je door het recept van een boterham met hagelslag aan te passen voor Olaf de Robot.



We helpen je vast op weg met een lijst instructies.
Je hoeft ze alleen nog in de goede volgorde te zetten.
Zet op je werkblad nummers voor de zinnen.



HOE SMEERT OLAF DE ROBOT EEN BOTERHAM MET HAGELSLAG?

- Leg het mes neer
- Maak de zak met brood open
- Maak het kuipje boter open
- Leg de boterham op het bord
- Pak een boterham uit de zak met brood
- Pak het pak hagelslag op
- Zet het pak hagelslag neer
- Smeer de boter met het mes op de boterham
- Haal met het mes wat boter uit het kuipje
- Strooi de hagelslag op de boterham
- Pak het mes op



PROEF OP DE SOM

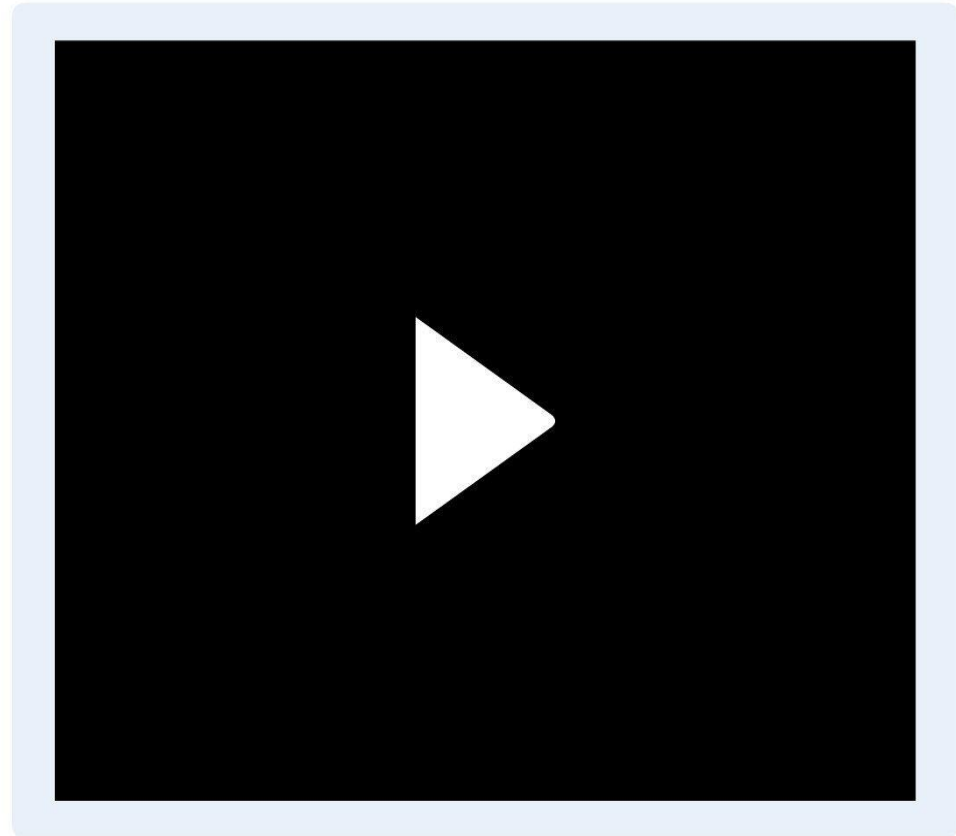
Gefeliciteerd, je hebt net je eerste programma geschreven voor een computer!

Merk je op dat er weinig verschil is tussen de twee programma's? Beide recepten zijn opgedeeld in stappen. Die voor Olaf de Robot zijn alleen wat gedetailleerder. En als we een recept schrijven voor een computer noemen we dat een **algoritme**. De stappen waar het uit bestaat heten **statements**.

Wat vind je van het programma? Denk je dat Olaf de Robot met deze instructie succesvol een boterham met hagelslag kan smeren? Of voorzie je onderdelen in het recept waar Olaf wellicht niet goed mee uit de voeten kan?

Laten we ons programma eens **runnen** om te testen hoe het onze robot vergaat. Dat betekent dat we de set instructies uitvoeren, in dit geval door de video te starten. Zoek op YouTube de video Olaf de robot en kijk wat er gebeurt.

Probeer zelf eens iemand te programmeren. Kies een simpele handeling, zoals bijvoorbeeld het inschenken van een glas water, en schrijf vooraf eerst alle stappen op. Vervolgens volg je alle handelingen zoals je ze opgeschreven hebt en debug je net zolang totdat je het juiste algoritme hebt. SUCCES!





Zoals je ziet, heeft Olaf de Robot veel duidelijkere uitleg nodig dan Maria om een boterham met hagelslag te smeren. Tijdens het **runnen** van de **algoritmes** voor de klas ging er heel wat mis. Als een computer onverwacht of onjuist gedrag vertoont, doordat de instructies niet helemaal kloppen, heet dat een **error**.

Het opsporen van fouten in een computerprogramma noemen we **debuggen**. Letterlijk betekent dat "ontvlooien". Het houdt in dat we stap voor stap door de instructies lopen om fouten op te sporen en te herstellen. Net zo lang tot alle stappen duidelijk genoeg zijn en de computer met alle mogelijke situaties overweg kan.

In ons geval moeten we er bijvoorbeeld voor zorgen dat Olaf alles dat hij oppakt na gebruik weer keurig neerlegt, dat hij alles dat hij opent, na afloop weer netjes sluit. Maar ook dat we Olaf leren om om te gaan met onverwachte omstandigheden. We moeten bijvoorbeeld controleren of Olaf het mes niet aan de scherpe maar de botte kant vasthoudt. En uitleggen wat de robot moet doen als het pak hagelslag leeg is.

Als het programma helemaal getest is en Olaf in alle gevallen een boterham met hagelslag krijgt, noemen we het **monkeyproof**. Het houdt in dat het zo duidelijk uitgelegd is, dat zelfs een aap de instructies op kan volgen zonder er een puinhoop van te maken



FOUTEN EN OPLOSSINGEN

Hieronder zie je afbeeldingen van 3 errors:

1. Olaf kreeg niet alle boter met het mes uit het kuipje
2. Olaf pakte het boterkuipje niet op omdat hij de broodzak nog vast had
3. Olaf kon de boterham niet snijden zonder die vast te houden

Hoe hadden we die errors kunnen voorkomen?
trek een lijn naar de juiste instructie.



Duidelijk aangeven wat de computer met welke hand moet doen

Goed aangeven hoe lang, hoe veel of hoe vaak de computer iets moet doen

Alles dat je oppakt of opent, ook weer terugleggen of sluiten



KENNISCHECK

Wat is er simpeler dan het smeren van een boterham met hagelslag? In deze module hebben we het uitgelegd aan een mens en aan een computer.

Wij, Maria en Olaf spreken verschillende talen. Maar wat er precies in hun menselijke of digitale brein gebeurt, hoeven wij niet te begrijpen om eenvoudige instructies te geven.

In beide gevallen communiceerden we met een stappenplan. Maria kon met een paar complexe aanwijzingen uit de voeten, omdat wij mensen al heel veel weten en begrijpen van hoe de wereld in elkaar zit. Voor Olaf moesten we veel preciezer beschrijven wat de bedoeling was. Een robot heeft geen voorkennis, hij snapt niet wat er gebeurt en doet precies wat hij opgedragen krijgt. Eigenlijk zijn wij mensen dus veel slimmer.

In feite is dit waar het bij programmeren om draait: in kleine, duidelijke stappen aan een computer uitleggen om een grote, complexe handeling uit te voeren. Zoals je hebt gemerkt is dat vooral een kwestie van logisch nadenken. In de volgende modules gaan we aan de slag met voorbeelden in de wereld om je heen, oefeningen in de klas en opdrachten om online te programmeren.

Daarbij maak je steeds gebruik van de begrippen die we zojuist hebben behandeld. Heb jij nog goed in je hoofd wat ze betekenen?





WELKE UITLEG HOORT BIJ DE BEGRIPPEN?

De blauwe begrippen hebben we in deze module behandeld.
Schrijf het juiste begrip boven de juiste uitleg. SUCCES!



PROGRAMMEERTAAL
ALGORITME
STATEMENT
DEBUGGEN
RUNNEN

.....

Is een ander woord voor het uitvoeren van je programma. Alle instructies die je onder elkaar hebt opgeschreven, worden in die volgorde achter elkaar gerealiseerd. In één keer, van de eerste tot de laatste stap, tenzij er onderweg een error optreedt.

.....

Is het nalopen van alle instructies, om fouten op te sporen en te herstellen. Net zo lang tot de lijst van alle stappen klopt en compleet is.

.....

Is een weergave van de instructies die we aan een robot kunnen geven. Net zoals Nederlands en Spaans een weergave zijn van hoe wij denken.

.....

Is een verzameling van eenvoudige stappen die beschrijven hoe je een complexe handeling uit kunt voeren. Net als een recept.

.....

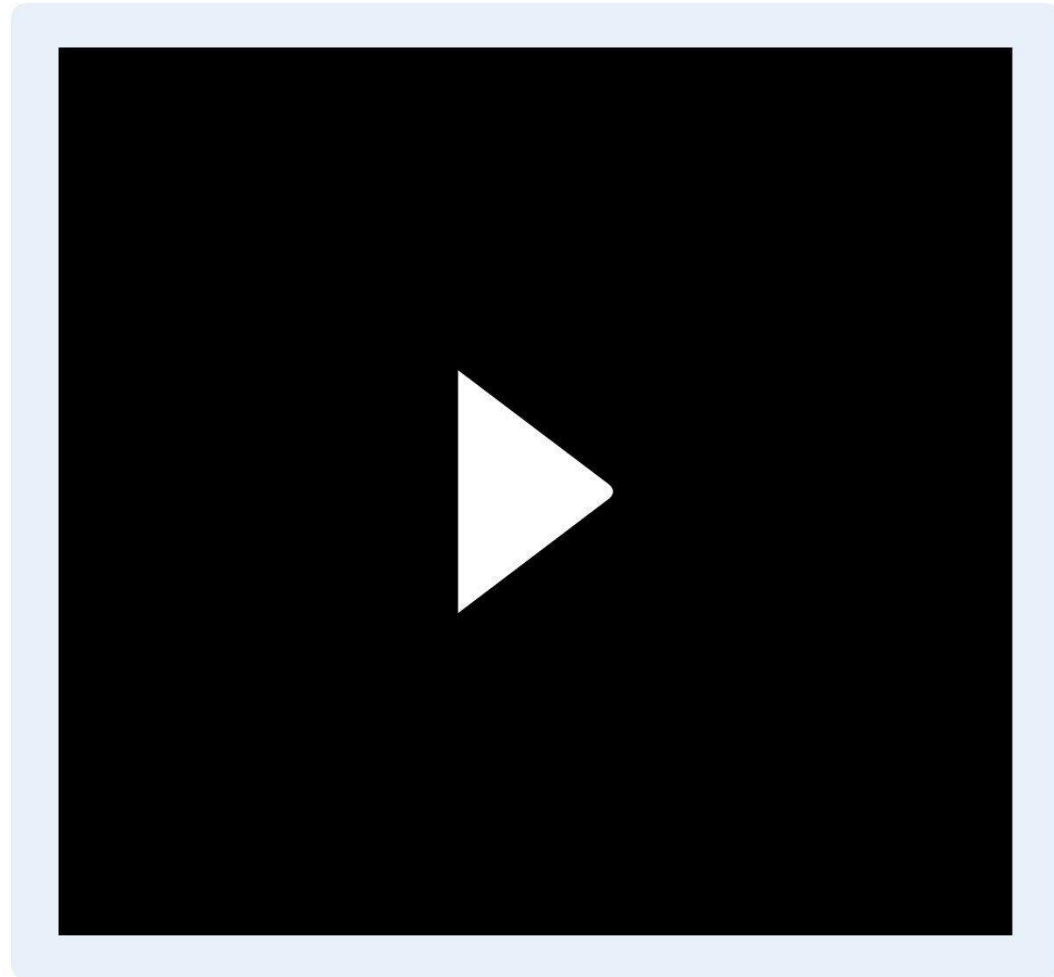
Is een simpele opdracht voor een computer. Bijvoorbeeld: "pak op", "zet een stap" of "leg neer".



YES, ALWÉÉR HETZELFDE!

Computers hebben veel meer rekenkracht dan jij en ik om ingewikkelde problemen op te lossen... en tegelijkertijd hebben ze de grootst mogelijke moeite met instructies die voor ons gesneden koek zijn. Dat komt omdat computers anders denken.

Herhaling. Voor ons mensen is het al gauw vervelend - en soms zelfs irritant. Computers kunnen er echter prima mee uit de voeten. Maar stel dat we die gegevens combineren... Ligt daar een kans?





HERHALING: SAAI OF SPANNEND?

Wij mensen zien herhaling vaak als saai of zelfs straf.

We zijn er niet goed in en proberen het zoveel mogelijk te vermijden met slimme uitwegen.

Met computers is het andersom. Die blinken juist uit in herhaling.

Maar als wij mensen aan computers opdrachten geven om te herhalen, dan wordt het spannend! Let maar eens op.

Neem onderstaande begrippen door en zie wat ze betekenen.



HERHALINGEN

EEN DEEL VAN EEN COMPUTERPROGRAMMA DAT MEERDERE KEREN
HERHAALD WORDT, OOK WEL EEN LUS GENOEMD

ABSTRACTIE

HET VERKLEINEN VAN MOEILIKHEDEN EN HET
VERGROTEN VAN EEN ALGORITME

PATRONEN

EEN BESCHRIJVING, OF WEERGAVE VAN EEN PROBLEEM OF
LOSSING DIE MEERDERE KEREN HERHAALD OF HERBRUIKT KAN
WORDEN

HOE TEKEN JE EEN MONSTER?

Laten we beginnen met het tekenen van een monster. Voor het gemak nemen we een exemplaar dat niet al te angstaanjagend is, zoals het voorbeeld rechts.

Bekijk het eens goed en probeer na te denken over de volgende vragen:

- Hoe zou je het aanpakken om dit monster zelf te tekenen?

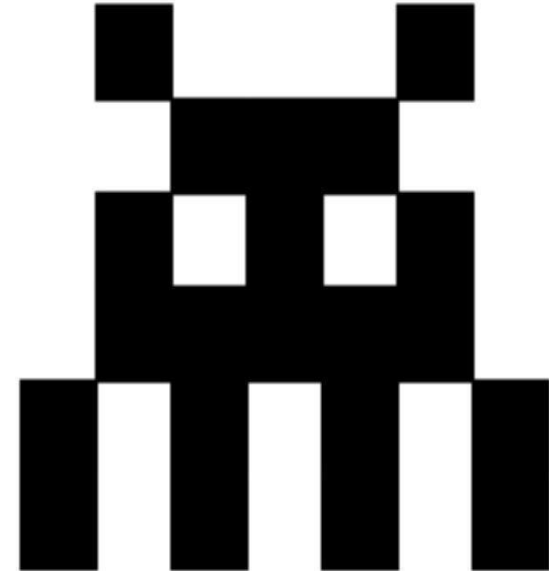
- Wat heb je nodig om aan iemand (die jou niet kan zien!) uit te leggen hoe hij of zij dit monster kan tekenen?

- Hoe zou je een computer uitleggen om dit monster te tekenen?

WAT HEB JE NODIG OM EEN MONSTER TE TEKENEN

Let bij het geven van je antwoord op de volgende zaken:

- Uit welke vormen het monster is opgebouwd
- Welke materialen je nodig hebt om dit monster te (laten) tekenen
- Hoe je aan kunt geven welk deel van de tekening is ingekleurd



EEN MONSTERLIJKE WEERGAVE

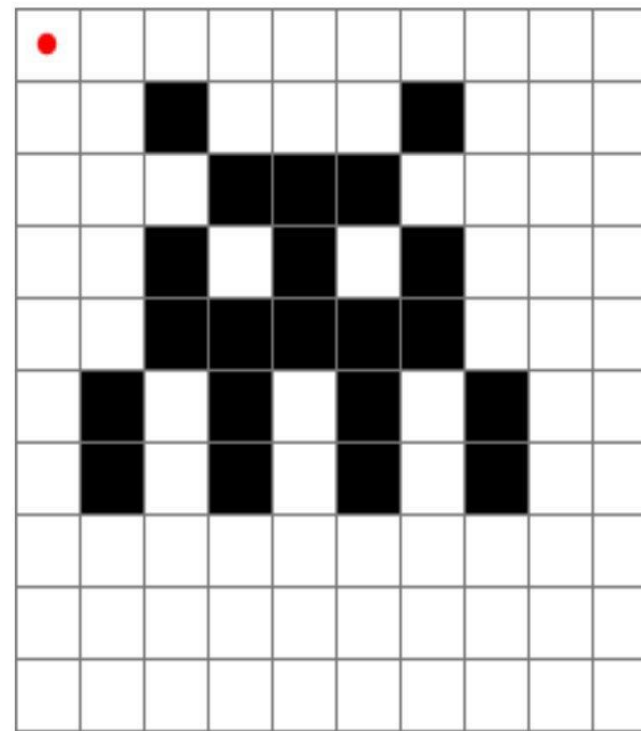
We kunnen de tekening van het monster namaken met een vel ruitjespapier.

De vakjes geven houvast om het monster na te tekenen.
Het is vooral een kwestie van goed kijken en tellen.

De vakjes geven zelfs zoveel houvast, dat je iemand anders kunt uitleggen om deze tekening te maken, zonder dat hij jou (of het monster) kan zien, door gebruik te maken van de volgende instructies:

- ga een vakje naar rechts
- ga een vakje omlaag
- ga een vakje naar links
- ga een vakje omhoog
- kleur dat vakje in

Probeer het maar eens met de volgende oefening.



TEKEN DE EERSTE RIJEN

Als je linksboven start bij de rode stip en de vakjes volgt, welke instructies heb je dan nodig om de eerste drie rijen van het monster te tekenen? Zet een krul voor de juiste code. Er zijn meerdere antwoorden mogelijk.

Let op: je gaat steeds verder vanuit het vakje waar je gebleven was.

Ga een vakje omlaag, ga een vakje naar links, kleur dat vakje in, ga een vakje naar links, kleur dat vakje in, ga een vakje naar links, kleur dat vakje in.

Ga een vakje naar rechts, ga een vakje naar rechts, ga een vakje naar rechts, ga een vakje naar rechts, kleur dat vakje in.

Ga een vakje naar rechts, ga een vakje naar rechts, ga een vakje omlaag, kleur dat vakje in.

EEN SYMBOLISCH MONSTER

In de vorige oefening heb je vast gemerkt dat zinnestjes in menselijke taal niet het handigst zijn om uit te leggen hoe we ons monster kunnen tekenen. De instructies zijn lang en het herhalen van woorden maakt ze onoverzichtelijk.

Het wordt al flink eenvoudiger door gebruik te maken van symbolen:

- als symbool voor "ga een vakje naar rechts"
- ↓ als symbool voor "ga een vakje omlaag"
- ← als symbool voor "ga een vakje naar links"
- ↑ als symbool voor "ga een vakje omhoog"
- als symbool voor "kleur dat vakje in"

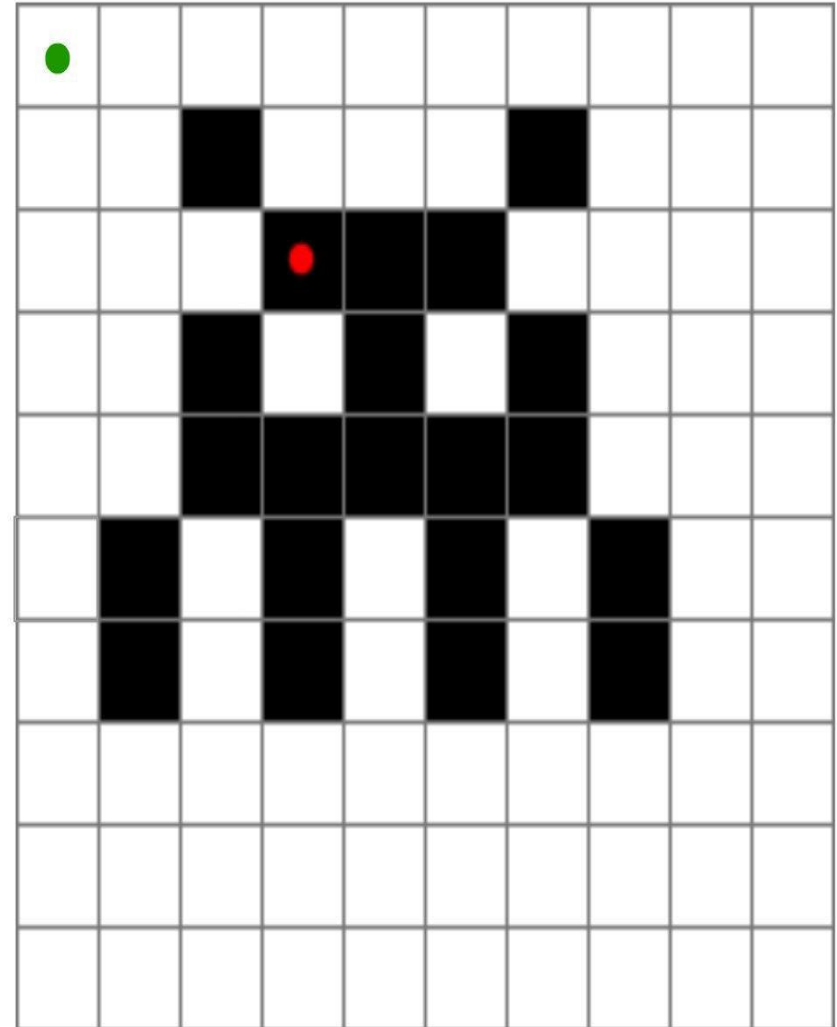
De instructies voor het tekenen van eerste twee regels kunnen we nu

een stuk simpeler weergeven: ● → → ↓ ■ → → → → ■ ↓ ← ■ ← ■ ← ■

DE VOLGENDE RIJEN VAN HET MONSTER

Hieronder zie je het **algoritme** voor de volgende rijen van het monster. We starten bij de rode stip. Zet een kruis in het vakje waar je eindigt na het volgen van deze stappen.

● ← ↓ ■ → → ■ → → ■ ↓ ■ ← ■ ← ■ ← ■ ← ■ ← ← ↓





ZOMBIES EN ZONNEBLOEMEN

De bovenste helft van ons monster kunnen we nu als volgt maken.

→ → ↓ ■ → → → → ■ ↓ ← ■ ← ■ ← ■

← ↓ ■ → → ■ → → ■ ↓ ■ ← ■ ← ■ ← ■ ← ■ ← ← ↓

Kijk er nog eens goed naar. Vallen je patronen op die vaker terugkomen?

De herhaling van → → (ga een vakje naar rechts, ga een vakje naar rechts) kunnen we ook weergeven als → 2 (ga twee vakjes naar rechts).

Daarmee kunnen we de reeks → → → → versimpelen tot → 4 (ga vier vakjes naar rechts).

De laatste herhaling van ← ← kunnen we dan opschrijven als ← 2 (ga twee vakjes naar links).

We geven met het getal aan dat we dezelfde instructie vaker uitvoeren.

Maak je geen zorgen als het je duizelt met alle pijlen en vakjes. Er wacht een leger monsters op je om dit stap voor stap te oefenen.

Ga naar de website <https://studio.code.org/s/course3/stage/2/puzzle/1> om met behulp van herhalingen de zombie naar de zonnebloem te loodsen... zonder dat je ten prooi valt aan een vleesetende plant!

Probeer ten minste één oefening, als je de smaak te pakken hebt mag je door tot en met de 8e.





KRIJG JIJ HET MONSTER KLEIN?

Heb je de herhaling onder de knie? Nu wordt het spannend!
We stoeien nog één keer met ons monster.

Als je goed kijkt naar het algoritme voor de bovenste vijf rijen, valt je misschien nog een patroon op.

→ 2 ↓ ■ → 4 ■ ↓ ← ■ ← ■ ← ■

← ↓ ■ → 2 ■ → 2 ■ ↓ ■ ← ■ ← ■ ← ■ ← 2 ↓

De herhaling van ← ■ ← ■ ← ■ ← ■ (naar links, kleur in, naar links, kleur in, naar links, kleur in, naar links, kleur in) kunnen we opschrijven als (← ■) 4 (kleur naar links vier vakjes in).

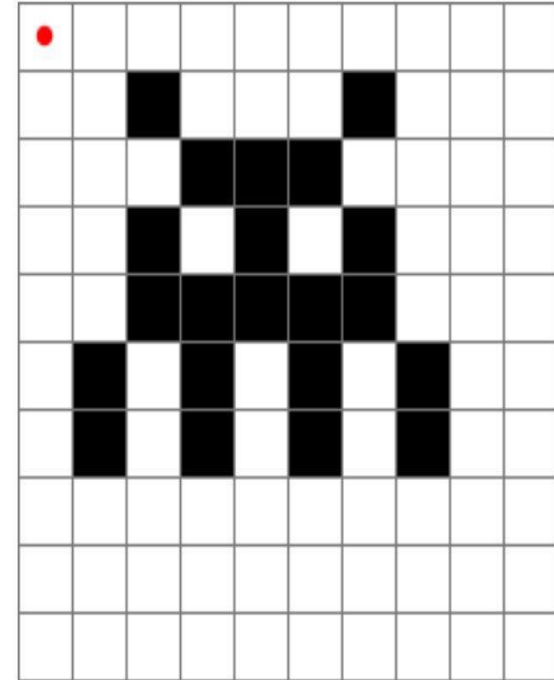
Het patroon → 2 ■ → 2 ■ (naar rechts, naar rechts, kleur in, naar rechts, naar rechts, kleur in) kan ook beschreven worden als (→ 2) 2 (sla naar rechts steeds een vakje over en kleur de volgende in).

Het algoritme tot nu toe kunnen we ook versimpelen tot:

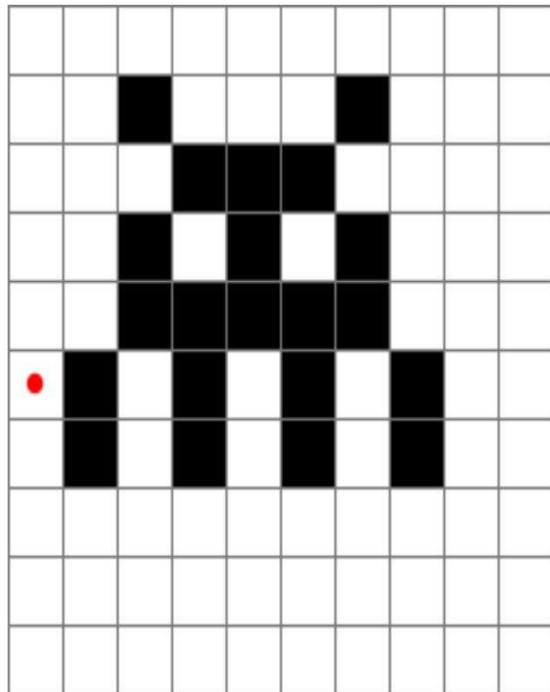
→ 2 ↓ ■ → 4 ■ ↓ (← ■) 3

← ↓ ■ (→ 2 ■) 2 ↓ ■ (← ■) 4 ← 2 ↓

Laten we eens proberen of we de rest van het monster klein kunnen krijgen.



KRIJG JIJ HET MONSTER KLEIN?



HOE KLEIN KRIJG JIJ DE VOETEN?

Welke van de onderstaande algoritmes tekenen de voeten van het monster?

We beginnen weer vanaf de rode stip.

1. $\rightarrow \blacksquare \rightarrow \rightarrow \blacksquare \rightarrow \rightarrow \blacksquare \rightarrow \rightarrow \blacksquare \downarrow \blacksquare \leftarrow \leftarrow \blacksquare \leftarrow \leftarrow \blacksquare \leftarrow \leftarrow \blacksquare$
2. $\rightarrow \blacksquare (\rightarrow \rightarrow \blacksquare) 3 \downarrow (\blacksquare \leftarrow \leftarrow) 3 \blacksquare$
3. $\rightarrow \blacksquare \downarrow \blacksquare \rightarrow \uparrow \rightarrow \blacksquare \downarrow \blacksquare \rightarrow \uparrow \rightarrow \blacksquare \downarrow \blacksquare \rightarrow \uparrow \rightarrow \blacksquare \downarrow \blacksquare \rightarrow \uparrow$
4. $(\rightarrow \blacksquare \downarrow \blacksquare \rightarrow \uparrow) 4$
5. $\downarrow \rightarrow \blacksquare \uparrow \blacksquare \rightarrow \downarrow \rightarrow \blacksquare \uparrow \blacksquare \rightarrow \downarrow \rightarrow \blacksquare \uparrow \blacksquare \rightarrow \downarrow \rightarrow \blacksquare \uparrow \blacksquare \rightarrow$
6. $(\downarrow \rightarrow \blacksquare \uparrow \blacksquare \rightarrow) 4$

Zet een kruis in het juiste vakje

- 1 en 2
- 3 en 4
- 5 en 6
- 1, 2, 3, 4, 5, en 6



SLIMME ZOMBIES

In de voorgaande oefeningen, zijn we steeds op zoek gegaan naar **patronen** in het monster. Op basis daarvan hebben we het **gerepresenteerd** als gekleurde vakjes op ruitjespapier.

Ons algoritme om het te tekenen, maakten we eerst in woorden.

Weer zochten we **patronen** en maakten we het eenvoudiger met symbolen.

Ook daarin zagen we **herhaling**, die we aangaven met getallen.

In de laatste stap combineerden we patronen met herhalingen tot een algoritme dat steeds korter en krachtiger werd. Dat noemen we **abstractie**.

Het is één van de meest belangrijkste onderdelen van programmeren - dat in het begin best lastig kan zijn. Laat je daar niet door van de wijs brengen.

Ga naar deze website <https://studio.code.org/hoc/1> en start met programmeren.



