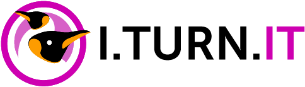
handleiding

herhalingen

# Kennismaking met programmeren

Project van de Pedagogische Academie, Hanzehogeschool Groningen en Groningen Programmeert in samenwerking met:



### **Leerdoelen**

* De kinderen kunnen voorbeelden geven van taken waarbij computers steeds opnieuw min of meer hetzelfde moeten doen
* De kinderen kunnen een reeks instructies schrijven als een herhaling
* De kinderen ervaren dat het sorteren van een rij getallen te beschrijven is als een herhalingslus waarbij variabelen steeds worden vergeleken

### **Materialen**

* Werkbladen
* Getalbladen

### **Lesverloop**

In deze les leren de leerlingen hoe ze herhaling kunnen gebruiken bij het programmeren. Ze leren een herhalingssituatie te herkennen en een bestaand programma te verkorten door het toepassen van een herhalings-element. Eerst spelen de kinderen samen met de leerkracht een spel, waarbij het concept van herhaling duidelijk uitgelegd wordt. Vervolgens gaan de leerlingen zelfstandig aan het werk met de werkbladen. Daarna volgt een uitloopopdracht op de computer, mits daar genoeg computers voor aanwezig zijn. Anders volgt deel 3 van de les, een klassikale demonstratie van hoe een willekeurige verzameling getallen gesorteerd kan worden door steeds dezelfde programmeercode te herhalen.

### **Deel 1: raad mijn getal**

De les begint met een uitleg over het belang van herhalingen. Computers doen heel vaak hetzelfde, zoals bijvoorbeeld heel vaak kleine vierkanten (pixels) tekenen om het scherm van een spelletje zoals Minecraft op het scherm af te beelden. Een ander voorbeeld is dat een computer vaak moet sorteren, bijvoorbeeld op leeftijd of op prijs in een webshop. De computer doet dan eigenlijk steeds hetzelfde, heel snel achter elkaar. Vandaag zullen de leerlingen leren hoe ze een **herhaling** kunnen inzetten om het programmeren makkelijker te maken.

Een leerling neemt een getal tussen de 0 en de 50 in gedachten. De docent gaat proberen door het getal zo snel mogelijk te raden. Hiervoor mag hij maar twee soorten vragen stellen:

* *Is het getal groter dan … ?*
* *Is het getal gelijk aan … ?*

Gebruik hiervoor in eerste instantie een strategie die niet handig (snel) is, zoals een paar keer in het wilde weg raden naar het getal (*Is het 11? Is het 23?)* Dit gaat erg lang duren en is niet systematisch. Vraag de kinderen of ze een betere strategie kunnen bedenken. Laat een van deze kinderen zijn of haar strategie voordoen.

De kinderen gaan nu in tweetallen aan de slag om met zo min mogelijk vragen tot het antwoord te komen. Ze moeten hierbij turven hoeveel vragen ze nodig hebben om het getal te raden. De kinderen spelen het spelletje 3 keer.

Bespreek na afloop van het spelletje de gebruikte strategieën klassikaal. Welke strategieën werkten goed? Welke strategieën leiden consequent tot een snelle oplossing? Met behulp van de getallenlijn kan een snelle strategie getoond worden. Teken een getallenlijn van 0 tot 50 op het bord. Deel deze in tweeën en stel hierbij de vraag: “*Is het meer dan 25?”* Afhankelijk van het antwoord kan een deel van de getallenlijn weggeveegd worden. Als het antwoord “ja” was, kunnen de getallen van 0 tot 25 weggehaald worden, want die vallen af.

Door steeds op de helft van de overgebleven getallen te gaan zitten kom je steeds dichter bij het juiste antwoord. Je kunt het bereik van de getallen steeds weergeven met behulp  van twee variabelen: *maximum* en *minimum*. Een van deze twee variabelen verandert steeds, op basis van het antwoord van de vraag die gesteld wordt.

Deze manier van oplossen heeft een herhaling in zich. Je stelt steeds een bereik op, met behulp van je twee variabelen *maximum* en *minimum*. Je gaat dan in het midden zitten en vraagt of het meer of minder is dan het midden. Vervolgens pas je je bereik aan en herhaal je je instructie weer. Je halveert steeds het gebied waar het getal kan liggen, totdat halveren niet meer kan en je zeker weet welk getal het is.

### **Deel 2: tekenen**

In de voorgaande lessen hebben de leerlingen geleerd dat ze heel precies moeten zijn wanneer ze een computer opdrachten geven. Maar, als een computer steeds dezelfde taak moet doen, zoals bij het raden van het getal van hiervoor, is het veel werk om precies op te schrijven wat er moet gebeuren, zeker wanneer de instructies zo sterk op elkaar lijken. Het zou veel handiger zijn om het een stuk korter op te schrijven.

De presentatie laat een duidelijk voorbeeld van het principe van herhaling zien. Rechts staat een cirkel met 8 punten en links een programmeervak. De regels die in het programmeervak komen te staan horen hebben invloed op de cirkel. Als er staat *teken-lijn-van-0-naar-1* betekent dat dat er in de cirkel een lijn getekend wordt van punt 0 naar punt 1. De presentatie gaat door met steeds meer programmeerregels, tot er uiteindelijk een groot ingewikkeld figuur ontstaat. De vraag is echter of het niet korter opgeschreven kan worden. Vertel de leerlingen dat met slechts 3 regels deze zelfde figuur gemaakt kan worden (waar nu 14 regels voor nodig zijn).

De regel *herhaal x van 0 t/m 7* houdt in dat X de waarde van 0 t/m 7 kan hebben. Voor elke waarde van X wordt de regel die eronder staat herhaald. X begint als 0, dus dan ontstaat de programmeerregel *teken-lijn-van-0-naar-0*. Dit leidt uiteraard niet tot een lijn in de figuur. Wanneer bij de volgende iteratie X de waarde van 1 heeft, wordt er wel een lijn getekend. Als X ingevuld wordt in de tweede programmeerregel staat er *teken-lijn-van-0-naar-1*, en dit leidt wel tot een getekende lijn. Op die manier kan de hele eerste helft van de figuur getekend worden met slechts twee regels. De tweede helft van de figuur kan getekend worden door de regel *teken-lijn-van-4-naar-x* toe te voegen aan het programma. Voor elke waarde van X voert het programma nu twee regels code uit.

De leerlingen kunnen nu zelfstandig verder met het werkblad. Bij opdracht 1 moeten de leerlingen het bijbehorende programma zelf uitvoeren en een tekening produceren. Bij opdracht 2 moeten de leerlingen de figuur analyseren en de achterliggende programmeercode achterhalen. In het vakje kunnen ze zelf de code schrijven die hoort bij de figuur. Wanneer de leerlingen er moeite mee hebben kan als tip de structuur van de voorgaande opdracht en die van de presentatie gebruikt worden: *herhaal x van … t/m …: teken-lijn-van-…-naar-x*. Bij opdracht 3 moeten de leerlingen in tweetallen werken. Eén schrijft een programma en laat zijn buurman of –vrouw deze tekenen, en omgekeerd.

Wanneer leerlingen moeite hebben met het opschrijven van de code inclusief herhaling, kunnen ze als tussenoplossing eerst de gewone code, zonder herhaling, opschrijven. Vervolgens moeten de leerlingen hun geschreven code analyseren om te kijken of ze een herhaling kunnen ontdekken. Daarna schrijven ze de code met herhaling op, waardoor er een stuk minder code geschreven wordt geschreven.

Na afloop van het maken van de werkbladen kunnen de leerlingen online verder werken op <http://studio.code.org/s/course2/stage/6/puzzle/1> of <http://studio.code.org/s/frozen/stage/1/puzzle/1>

### **Deel 3: sorteren**

De laatste opdracht bestaat uit het sorteren van leerlingen. Zoals in het begin van de les al gezegd is moet een computer vaak sorteren, zoals bij een webshop, en bij deze opdracht gaan de leerlingen hiervoor een programma maken. Hiervoor zijn 8 vrijwilligers nodig. Ze gaan met hun stoel voor de klas zitten, met hun gezicht naar de klas, en krijgen één van de getalbladen. De kinderen krijgen dit getal willekeurig. Het programma zorgt ervoor dat de leerlingen straks netjes op een rij zitten. De stoel op positie 1 (meest linker stoel vanuit de klas gezien) wordt overgeslagen. We beginnen met de leerling op positie 2 en nemen het programma door. Het programma staat op de presentatie. Klopt het, wordt iedereen netjes gesorteerd? Welke variabelen worden er gebruikt in dit programma? (naam van volgende kind, positie, eigen getal, getal van buurman).

### **Afsluiting**

Soms moeten computer heel vaak de zelfde opdracht uitvoeren, die slechts een klein beetje anders is dan de vorige opdracht. Door een variabele in het programma te stoppen en te herhalen, kan het programma een stuk korter geschreven worden. Het resultaat is gelijk, maar het programma zelf is een stuk korter.