

# Iedereen kan programmeren Voorschoolse en vroegschoolse educatie

Handleiding voor leerkrachten



## Inhoudsopgave

### Inleiding

### Commando's

- Les 1: Dagelijkse routines
- Les 2: Verhaalvolgorde
- Les 3: Dansmoves

### Functies

- Les 1: Papieren edelsteen
- Les 2: Songfestival
- Les 3: Mijn kalmeerfunctie

### Lussen

- Les 1: Herhalende bloemblaadjes
- Les 2: Hindernisbaan
- Les 3: Drumpatronen

### Variabelen

- Les 1: Zinken of drijven
- Les 2: Woordspel
- Les 3: Alles over mij

### App-ontwerp

### Informatiebronnen voor begeleiders

*Iedereen kan programmeren: Voorschoolse en vroegschoolse educatie is ontworpen om leerkrachten en gezinnen te helpen om kinderen uit de lagere groepen te laten kennismaken met programmeren. In dat stadium ontwikkelen kinderen namelijk de eerste vaardigheden op het gebied van computational thinking. Door middel van deze lessen gaan leerlingen van 5 t/m 8 jaar verkennen, ontdekken en spelen om zich een aantal fundamentele programmeerbegrippen eigen te maken.*



## Educatief ontwerp

Deze gids is verdeeld in vier modules, met een afsluitend app-ontwerpproject. Elke module bevat drie lessen, die elk gericht zijn op één begrip dat te maken heeft met programmeren. Binnen elke les vindt u drie activiteiten: Verkennen, Ontdekken en Spelen. De activiteiten kunnen worden opgedeeld in meerdere sessies of dagen.

### Dag 1: Discussie en praktisch leren

#### Verkennen

Kennismaking met en bespreking van het programmeerbegrip

#### Ontdekken

Vertrouwd raken met het concept door middel van creatieve activiteiten

ong. 25  
minuten

#### Spelen

- Programmeer samen met Byte in de Swift Playgrounds-app
- Oefen met programmeren in de bijbehorende werkbladen en Keynote-activiteiten
- Breng de wereld van Byte naar de echte wereld met unplugged programmeergames in de vorm van vloerpuzzels

ong. 25  
minuten

### Dag 2: Het verband tussen leren en programmeren

## Bereik en volgorde

De vier modules in deze gids zijn ontworpen voor gebruik met leerlingen van 5 t/m 8 jaar en kunnen in willekeurige volgorde worden gedaan. We raden u aan om de module ‘App-ontwerp’ op gezette tijden of zelfs meerdere keren per jaar te gebruiken naarmate bij de leerlingen het begrip van code en apps groeit.

### Voorbeeld:

Leeftijd	Module	Afsluitend project	Geschatte totale tijd
5 jaar	Commando's	App-ontwerp	4 uur
6 jaar	Functies	App-ontwerp	4 uur
7 jaar	Lussen	App-ontwerp	4 uur
8 jaar	Variabelen	App-ontwerp	4 uur

### Verder leren

Voor het lesgeven aan leerlingen van 9 t/m 14 jaar biedt het lesprogramma ‘Iedereen kan programmeren: Puzzels’, samen met het ‘Verlag voor app-ontwerp’ en de ‘Handleiding voor app-presentaties’, meer dan 45 uur aan leeractiviteiten. Meer informatie vindt u in de [Leswijzer bij ‘Iedereen kan programmeren’](#).



### Leerlingportfolio's (optioneel)

Verzamel tijdens deze modules de resultaten van de activiteiten om er samen met de leerlingen een portfolio van te maken.








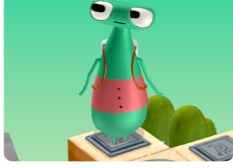


Module	Les	Suggesties voor werkstukken
<b>Commando's</b>	Dagelijkse routines	<ul style="list-style-type: none"> <li>• Werkblad 'Commando's geven'</li> <li>• Werkblad 'Commando toevoegen'</li> </ul>
	Verhaalvolgorde	<ul style="list-style-type: none"> <li>• Foto van punt in verhaallijn</li> <li>• Groepsfoto verhaallijn</li> </ul>
	Dansmoves	<ul style="list-style-type: none"> <li>• Dansmovekaarten</li> <li>• Video van dansmoves (optioneel)</li> </ul>
<b>Functies</b>	Papieren edelsteen	<ul style="list-style-type: none"> <li>• Vorm van papieren edelsteen</li> <li>• Werkblad 'Nieuw gedrag samenstellen'</li> <li>• Werkblad 'Nieuwe functie maken'</li> </ul>
	Songfestival	<ul style="list-style-type: none"> <li>• Concertvideo of geschreven functie voor Songfestival</li> </ul>
	Mijn kalmeerfunctie	<ul style="list-style-type: none"> <li>• Tekening of video van Mijn kalmeerfunctie</li> <li>• Werkblad 'Verzamelen, omzetten, herhalen'</li> </ul>
<b>Lussen</b>	Herhalende bloemblaadjes	<ul style="list-style-type: none"> <li>• Herhalende bloemblaadjes</li> <li>• Werkblad 'Lussen gebruiken'</li> <li>• Werkblad 'Doorlussen'</li> </ul>
	Hindernisbaan	<ul style="list-style-type: none"> <li>• Video of foto's van de hindernisbaan (optioneel)</li> </ul>
	Drumpatronen	<ul style="list-style-type: none"> <li>• Werkblad 'Naar de rand en terug'</li> <li>• Video of foto's van de drumsessie (optioneel)</li> </ul>
<b>Variabelen</b>	Zinken of drijven	<ul style="list-style-type: none"> <li>• Zinken of drijven</li> <li>• Werkblad 'Op de hoogte blijven'</li> </ul>
	Woordspel	<ul style="list-style-type: none"> <li>• Woordspelletjes</li> </ul>
	Alles over mij	<ul style="list-style-type: none"> <li>• Alles over mij</li> <li>• Alles over jou</li> </ul>
<b>App-ontwerp</b>		<ul style="list-style-type: none"> <li>• Wat is een app?</li> <li>• Mijn app-ontwerp</li> <li>• App-ontwerp-prototype</li> </ul>

## Aan de slag met Swift Playgrounds op iPad of Mac



Voordat u in de lessen duikt, moet u eerst [Swift Playgrounds](#), [Pages](#) en [Keynote](#) downloaden.

De modules in deze handleiding gebruiken verschillende combinaties van playgrounds. Dit hebt u nodig voor elke module:

Module	Playgrounds	Downloaden in Swift Playgrounds
<b>Commando's</b>	 Leren programmeren 1  MeeBot Leer dansen	Om u te abonneren op de MeeBot Playgrounds-feed, scrolt u helemaal omlaag in het scherm 'Meer playgrounds' en tikt u op 'Voer de URL voor een abonnement in'. Vervolgens voert u het volgende in: <a href="https://ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json">ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json</a> .
<b>Functies</b>	 Leren programmeren 1	
<b>Lussen</b>	 Leren programmeren 1  MeeBot Leer dansen	Om u te abonneren op de MeeBot Playgrounds-feed, scrolt u helemaal omlaag in het scherm 'Meer playgrounds' en tikt u op 'Voer de URL voor een abonnement in'. Vervolgens voert u het volgende in: <a href="https://ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json">ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json</a> .
<b>Variabelen</b>	 Leren programmeren 2  Steen, papier, schaar  Programmeermachine	'Steen, papier, schaar' en 'Programmeermachine' vindt u in het gedeelte 'Boeken' van het scherm 'Meer playgrounds'.
<b>App-ontwerp</b>		

Bekijk de minimumvereisten voor Swift Playgrounds in de [App Store](#). Ga naar [Apple Support](#) als u hulp nodig hebt bij Swift Playgrounds.

## Tips voor begeleiders

Om het meeste rendement te halen uit de lessen met uw leerlingen, kunt u deze tips proberen.

### De activiteiten 'Verkennen' en 'Ontdekken':

- Vereenvoudig de syntaxis en speciaal gebruik van hoofdletters bij het schrijven of laten zien van code, bijvoorbeeld:
  - `var names = ["Rose", "Sam", "Joy"]` --> `var names = Rose, Sam, Joy`
  - `var ages = [7, 8, 7, 8, 7]` --> `var ages = 7, 8, 7, 8, 7`
  - `var myFavoriteColor = ■` --> `var my favorite color = ■`

### Spelactiviteiten:

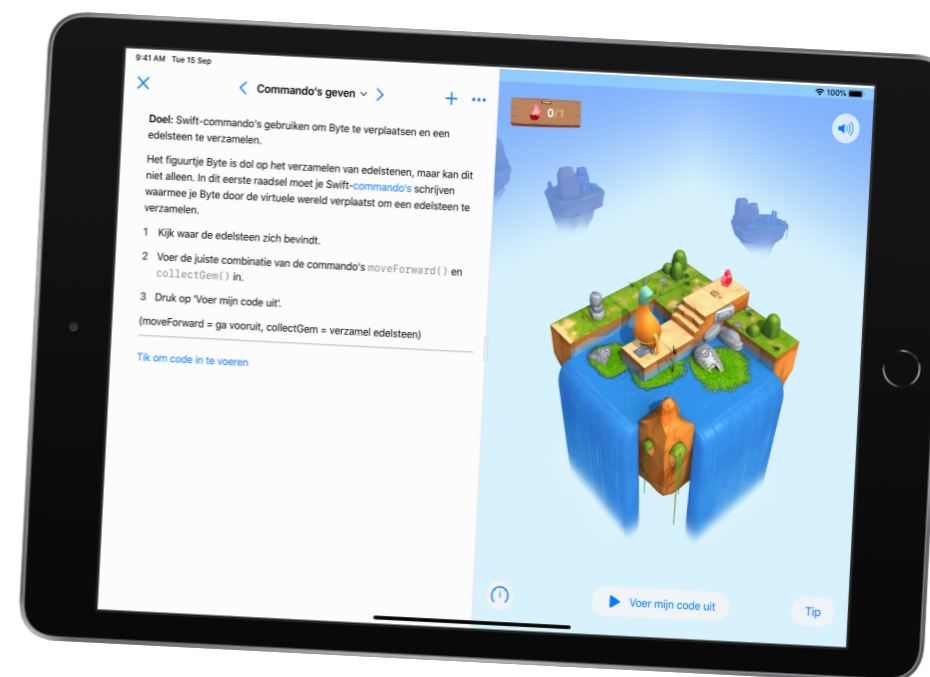
- Om het gebruik van de Swift Playgrounds-app nog eenvoudiger te maken voor uw vroege leerlingen, volgt u de instructies in de lesplannen. Een kleine greep daaruit:
  - Lees de introducties met de hele groep
  - Geef leerlingen beknopte aanwijzingen voor de bijbehorende werkbladen, zodat ze met eigen oplossingen kunnen komen
  - Gebruik één begeleiders-iPad of -Mac om de puzzels in de app op te lossen
- `let` en `var`: Het sleutelwoord `let` wordt in deze handleiding niet behandeld. Om verwarring in Swift Playgrounds te voorkomen, verandert u het sleutelwoord `let` overal waar het voorkomt in `var` voordat u de pagina's aan de leerlingen voorlegt. In de playgrounds die wij aanbevelen, zijn de twee sleutelwoorden uitwisselbaar.
  - `let` = variabele verandert niet
  - `var` = variabele verandert wel

### Als er tijd over is:

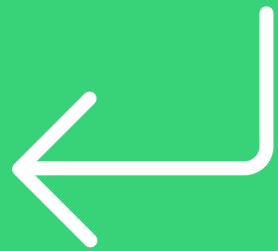
- Breid de vloerspelactiviteiten uit met rekenen, lezen, woorden zien, spelling enzovoort. Probeer voor inspiratie de activiteit 'Vloerspel' in de module 'Functies'.
- Maak de vloerspelactiviteiten persoonlijk voor de leerlingen door ze eigen kaarten te laten maken voor commando's, bijvoorbeeld `draaien()` of `springen()`.



Introductiepagina



Playgroundpagina



# Commando's



## Overzicht

### Les 1: Dagelijkse routines

- Verkennen: Discussie over bakken en commando's
- Ontdekken: Activiteit 'Dagelijkse routines'
- Spelen: 'Commando's geven' en 'Commando toevoegen'

### Les 2: Verhaalvolgorde

- Verkennen: Discussie over de volgorde binnen een verhaal en commando's
- Ontdekken: Activiteit 'Verhaalvolgorde'
- Spelen: Vloerpuzzelspel

### Les 3: Dansmoves

- Verkennen: Discussie over dansmoves en commando's
- Ontdekken: Activiteit 'Dansmoves'
- Spelen: 'Hallo MeeBot' en 'Basisbewegingen'

## Leerlingen kunnen

- Stapsgewijze instructies beschrijven aan de hand van alledaagse voorbeelden
- Instructies in een logische volgorde zetten
- Instructies en code testen en debuggen

## Termen

- **Sequentie:** De volgorde waarin dingen gebeuren
- **Stap:** Eén bewerking in een groter proces
- **Wijzigen:** Veranderen
- **Commando:** Code die een app opdracht geeft om een bepaalde bewerking uit te voeren
- **Bug:** Een programmeerfout
- **Debuggen:** Het zoeken en corrigeren van programmeerfouten



## Verkennen

**Doelstelling:** Kennismaking met het begrip 'commando' door het te koppelen aan het bakken van brownies.

### Discussie:

- Zouden ze zich bij het maken van brownies aan een recept houden?
- Zouden ze de stappen van het recept volgen in de gegeven volgorde?

**Wat steken ze hiervan op:** Elke stap of instructie in een recept is als een commando in code. Laat de leerlingen zelf commando's bedenken.

## Ontdekken

**Doelstelling:** Modelleer het proces van een dagelijkse routine door stapsgewijze instructies te identificeren.

**Materiaal:** 'Was je handen'-kaarten

### Aanwijzingen:

1. Schud de stapel 'Was je handen'-kaarten en leg ze op een tafel of hang ze op het bord. De kaarten mogen niet in de juiste volgorde liggen.
2. Vraag de leerlingen of ze denken dat er een fout zit in uw reeks instructies voor handen wassen.
3. Vraag leerlingen om de instructies te debuggen (verbeteren) door één kaart tegelijk te verplaatsen naar de juiste plek.

### Alternatief:

Laat de leerlingen in tweetallen of kleine groepjes werken en geef elk groepje een set kaarten.

### Als er tijd over is:

Laat de leerlingen zelf stapsgewijze instructies bedenken voor iets dat ze elke dag doen en foto's maken van de specifieke stappen.

↓ [Download de kaarten voor 'Was je handen'](#)



## Spelen

**Doelstelling:** Leerlingen kunnen de commando's in de juiste volgorde toevoegen om hun eerste edelstenen te verzamelen in 'Leren programmeren 1' in de Swift Playgrounds-app.

### Aanwijzingen:

1. Projecteer de introductiepagina van het hoofdstuk 'Commando's' van de playground 'Leren programmeren 1' op een scherm.
2. Inleiding:
  - Lees de pagina's met de hele klas door en stop zo nodig voor vragen.
3. Commando's geven:
  - Bekijk de twee commando's die leerlingen nodig hebben om Byte naar de edelsteen te leiden, `moveForward()` en `collectGem()`.
  - Vraag de leerlingen om te experimenteren met manieren om Byte van de startpijl naar de edelsteen te leiden en deze te verzamelen. Ze kunnen de commando's op het werkblad of op een apart vel papier noteren.
  - Verzamel ideeën uit de klas en schrijf de code in de Swift Playgrounds-app om de puzzel te voltooien. Klik of tik op 'Voer mijn code uit'.
  - Probeer verschillende ideeën.
  - Vier het met Byte!

### Als er tijd over is:

Als de leerlingen er klaar voor zijn, gaat u naar de volgende pagina, 'Commando toevoegen'. Hier gaan de leerlingen een nieuw commando gebruiken, `turnLeft()`.




### Leren programmeren 1

#### Materiaal voor de begeleider:

- iPad of Mac
- Swift Playgrounds-app
- Playground 'Leren programmeren 1'
- Projector of scherm

#### Materiaal voor de leerling:

- Werkbladen 'Commando's geven' en 'Commando toevoegen'
- Potloden
- Extra papier (optioneel)


[Download de werkbladen van 'Leren programmeren'](#)

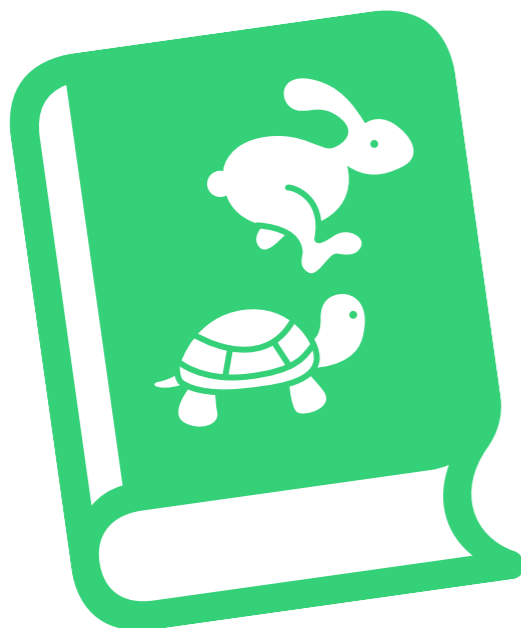
## Verkennen

**Doelstelling:** Verkennen hoe boeken een bepaalde volgorde hanteren (van begin tot midden tot eind) om te zorgen dat de verhalen geen onzin zijn.

### Discussie:

- Vraag de leerlingen of boeken een volgorde volgen.
- Wat zou er gebeuren als het begin, het midden en het einde van een boek niet in de goede volgorde stonden?
- Verken enkele voorbeelden.

**Wat steken ze hiervan op:** Leg de link met code en benadruk hoe belangrijk het is om commando's in code in de juiste volgorde te geven, net als de punten in een verhaallijn.



## Ontdekken

**Doelstelling:** Na het maken van afbeeldingen van verschillende punten in een verhaallijn kunnen leerlingen de afbeeldingen in de juiste volgorde zetten om het verhaal correct te reconstrueren.

### Materiaal voor de begeleider:

- Whiteboard
- Stiften

### Materiaal voor de leerling:

- Papier
- Stiften of kleurpotloden
- Alternatief: iPad-devices met een teken-app

### Aanwijzingen:

1. Lees een verhaal dat de leerlingen goed kennen. Bepaal als klas de belangrijkste punten in de verhaallijn. Mik op vier tot zes punten.
2. Maak kleine groepjes met evenveel leerlingen als er verhaallijnpunten zijn (vier punten betekent bijvoorbeeld vier leerlingen per groepje).
3. Laat elke leerling van het groepje één punt van de verhaallijn tekenen.
4. De groepjes staan om de beurt voor de klas, waarbij de leerlingen hun verhaalpunten niet in de juiste volgorde houden.
5. De toehoorders herschikken de tekeningen een voor een.
6. Maak een foto van elk groepje als de leerlingen eenmaal in de juiste volgorde staan.

### Als er tijd over is, of als alternatief:

Laat elk groepje leerlingen werken aan een ander verhaal, waarbij ze als groep bepalen wat de punten in de verhaallijn zijn voordat ze de afbeeldingen gaan tekenen.

## Spelen

**Doelstelling:** Leerlingen kunnen Byte door een fysiek raster naar een edelsteen leiden met behulp van richtingscommando's.

**Vorbereiding:** De leerlingen gaan werken in groepjes van drie. Maak met afplaktape voor elk groepje een raster van vier bij vier op de vloer.

### Aanwijzingen:

1. Deel het materiaal uit en deel de leerlingen in in groepjes van drie.
2. Lees alle rollen door en wijs aan elke leerling van het groepje een rol toe voor het eerste spel.
3. Laat de leerlingen het spel spelen, te beginnen met de rol van ontwerper.
4. Speel drie keer, waarbij de rolkaarten telkens rouleren.

### Rollen:

- Ontwerper: Plaats de edelsteen en de startpijl op het rooster.
- Programmeur: Plaats met hulp van je medeleerlingen de commandokaarten op of naast het raster om Byte naar de edelsteen te leiden en die te laten verzamelen.
- Tester: Begin met Byte op de pijl en volg de commandokaarten om Byte over het raster te verplaatsen. Als je de edelsteen verzamelt, ben je goed bezig! Als het niet lukt, moeten jullie als team de code samen debuggen (verbeteren).

### Alternatief:

Als leerlingen individueel met u werken of thuis leren, kunnen ze dit spel in hun eentje spelen met behulp van de downloadbare alternatieve Keynote-activiteit.

### Materiaal voor de begeleider:

- Afplaktape

### Materiaal voor de leerling:

- Rolkaarten
- Commandokaarten: `moveForward()`, `turnLeft()`, `turnRight()` en `collectGem()`
- Edelsteen
- Byte
- Pijl

↓ [Download het materiaal](#)

↓ [Download de alternatieve activiteit](#)



## Verkennen

**Doelstelling:** Verken het idee dat coderen creatief kan zijn!

### Discussie:

- Vraag de leerlingen of ze ooit een dans hebben geleerd.
- Had de dans een volgorde van te volgen stappen?
- Hoe wisten ze wat ze nu moesten doen?
- Hebben de dansmoves een naam?
- Gebruiken leerlingen ooit dezelfde moves op verschillende tijdstippen in een dans of in verschillende dansroutines?

**Wat steken ze hiervan op:** Help leerlingen in te zien dat programmeren creatief is en dat je als programmeur net als iemand die een dans ontwerpt, nieuwe commando's kunt bedenken en die op verschillende en interessante manieren kunt combineren.

## Ontdekken

**Doelstelling:** Een korte dansroutine maken, met kaarten om de dansmoves weer te geven. Elke dansmovekaart komt overeen met een commando in de playground 'Leren programmeren'.

### Materiaal voor de leerling:

- iPad-devices
- Keynote-app
- Camera-app
- Genoeg ruimte om te dansen

### Aanwijzingen:

1. Laat tweetallen of kleine groepjes leerlingen een korte dansroutine maken.
2. Zodra de leerlingen de routine hebben bedacht, maken ze kaarten van de verschillende dansmoves. Leerlingen moeten een tekening van de move en de naam ervan op elke kaart zetten, zo creatief en gek mogelijk.
3. Elk groepje voert zijn dans uit. Houd daarna een dansparty met de hele klas.

### Alternatief:

Leerlingen kunnen de downloadbare dansmovekaarten hieronder gebruiken om hun dans te maken, of de kaarten als voorbeeld gebruiken voor kaarten die ze zelf maken.

### Als er tijd over is:

De leerlingen maken een video van hun dans om die aan de groep te laten zien.

 [Download de dansmovekaarten](#)



## Spelen

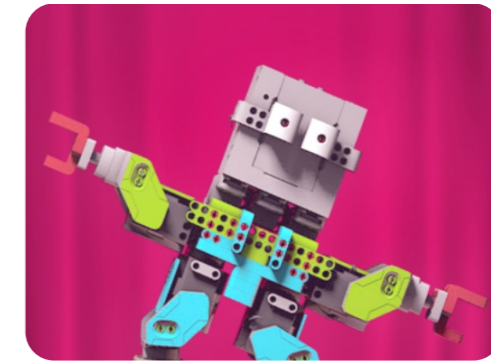
**Doelstelling:** Een reeks stappen maken om de MeeBot-robot een nieuwe dans te leren.

### Aanwijzingen:

1. Projecteer de playground 'MeeBot Leer Dansen' op een scherm. U moet u abonneren op de playground, als u dat nog niet hebt gedaan.
2. Inleiding:
  - Lees de pagina's met de hele klas door en stop zo nodig voor vragen.
3. Hallo MeeBot:
  - Klik of tik op 'Voer mijn code uit' en kijk hoe de robot danst.
4. Basismoves:
  - Als groep, in paren of individueel op hun eigen iPad, kiezen leerlingen acht commando's uit de lijst met suggesties en kijken hoe de robot danst.
  - Laat de leerlingen hun dansen delen, of maak met de hele klas een paar verschillende dansen.
  - Dans met de robot!

### Als er tijd over is:

- Ga verder naar de volgende pagina, 'Dansroutine', waar leerlingen moves kunnen toevoegen binnen de functie `myDanceRoutine()`, zo veel of zo weinig commando's als ze willen.



**MeeBot Leer dansen**

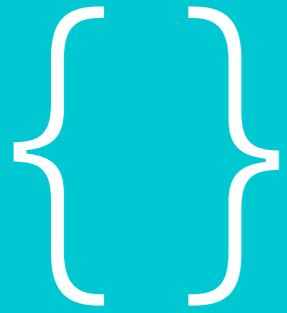
### Materiaal voor de begeleider:

- iPad of Mac
- Swift Playgrounds-app
- Playground 'MeeBot Leer Dansen'
- Projector of scherm

### Materiaal voor de leerling:

- iPad-devices (optioneel)





# Functies



## Overzicht

### Les 1: Papieren edelsteen

- Verkennen: Discussie over stapsgewijze instructies
- Ontdekken: Activiteit 'Papieren edelsteen'
- Spelen: 'Nieuw gedrag samenstellen' en 'Nieuwe functie maken'

### Les 2: Songfestival

- Verkennen: Discussie over hoe je een functie moet noemen
- Ontdekken: Activiteit 'Songfestival'
- Spelen: Vloerpuzzelspel

### Les 3: Mijn kalmeerfunctie

- Verkennen: Discussie over meerdere manieren om een probleem op te lossen
- Ontdekken: Activiteit 'Mijn kalmeerfunctie'
- Spelen: 'Verzamelen, omzetten, herhalen'

## Leerlingen kunnen

- Een groot probleem of grote taak opdelen in kleinere stappen
- Een reeks stappen bedenken om een probleem op te lossen of een taak te voltooien
- Functies een naam geven
- Code testen en debuggen

## Termen

- **Functie:** Een benoemde set commando's die op elk gewenst moment kan worden uitgevoerd
- **Omzetten:** In- of uitschakelen

## Verkennen

**Doelstelling:** Onderzoeken hoe je een reeks commando's verpakt en die reeks een naam geeft.

**Discussie:** Kies met de klas een dagelijkse routine waar je je op gaat concentreren. Laat de leerlingen de naam van hun dagelijkse routine identificeren en de stappen waaruit deze bestaat.

**Voorbeeld:** Bedtijdroutine

- Stap 1: Tandpoetsen
- Stap 2: Naar de wc
- Stap 3: Lezen
- Stap 4: Welterusten zeggen
- Stap 5: Het licht uit doen

**Wat steken ze hiervan op:** Een set instructies bedenken en er een naam aan geven is eigenlijk hetzelfde als een functie maken.

**Als er tijd over is:** Vraag de leerlingen of de instructies voor een van hun stappen specifieker kunnen zijn. Wat zijn bijvoorbeeld de specifieke stappen voor tandpoetsen?

## Ontdekken

**Doelstelling:** Leerlingen beginnen met het volgen van de instructies om een papieren edelsteen te maken. Daarna schrijven of tekenen ze de instructies om een andere vorm naar keuze te maken.

**Materiaal voor de leerling:**

- Papier
- Scharen
- Potloden
- iPad-devices (optioneel)

**Aanwijzingen:**

Laat leerlingen zien hoe je een papieren edelsteen maakt:

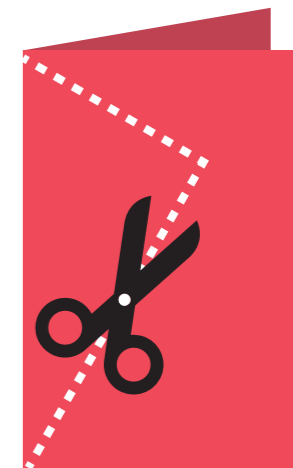
1. Vouw een stuk papier dubbel.
2. Trek een lijn van de bovenhoek aan de gevouwen kant tot een paar centimeter boven het midden van het papier.
3. Trek nog een lijn van waar de eerste lijn ophoudt naar de benedenhoek aan de gevouwen kant.
4. Knip langs de lijnen die je hebt getekend.
5. Haal de edelsteen van het kladpapier en vouw hem open.

Vraag de leerlingen om hun eigen vormen te maken:

1. Deel de leerlingen in kleine groepjes.
2. Laat de groepjes bedenken welke vorm ze willen maken.
3. Geef de leerlingen de tijd om een of twee keer te oefenen met het maken van de vorm.
4. Laat de leerlingen de aanwijzingen voor het maken van de vorm opschrijven of tekenen en hun aanwijzingen een naam geven, zoals 'Maak een cirkel' of 'De letter T'.

**Alternatief:**

Een video maken die laat zien hoe je hun vormen maakt.





## Spelen

**Doelstelling:** Door als groep te werken, kunnen leerlingen bepalen welke stappen nodig zijn om Byte bij de edelsteen te krijgen.

### Aanwijzingen:

1. Projecteer de playground 'Leren programmeren 1' op een scherm. Ga naar het hoofdstuk 'Functies' in 'Leren programmeren 1'.
2. Inleiding:
  - Lees de pagina's met de hele klas door en stop zo nodig voor vragen.
3. Nieuw gedrag samenstellen:
  - Bekijk de commando's `moveForward()`, `turnLeft()` en `collectGem()`. Houd er rekening mee dat er geen `commandoturnRight()` is.
  - Laat de leerlingen experimenteren met manieren om Byte van de startpijl naar de edelsteen te leiden en deze te verzamelen. Ze noteren de commando's op het werkblad of op een apart vel papier.
  - Verzamel ideeën uit de klas en schrijf de code in de Swift Playgrounds-app om de puzzel te voltooien. Klik of tik op 'Voer mijn code uit'.
  - Probeer verschillende ideeën.
  - Vier het met Byte!
4. Nieuwe functie maken:
  - Laat de leerlingen op basis van wat ze hebben geleerd op de laatste playground-pagina, 'Nieuw gedrag samenstellen', ideeën bedenken om zelf een functie `turnRight()` te maken.
  - Laat de leerlingen met behulp van hun eigen `turnRight()`-functie experimenteren met manieren om Byte van de startpijl naar de gesloten schakelaar te leiden en deze om te zetten.
  - Verzamel ideeën uit de klas en schrijf de code in de Swift Playgrounds-app om de puzzel te voltooien. Klik of tik op 'Voer mijn code uit'.
  - Probeer verschillende ideeën.
  - Vier het met Byte. Dit was een moeilijke puzzel!



### Leren programmeren 1

#### Materiaal voor de begeleider:

- iPad of Mac
- Swift Playgrounds-app
- Playground 'Leren programmeren 1'
- Projector of scherm

#### Materiaal voor de leerling:

- Werkbladen 'Nieuw gedrag samenstellen' en 'Nieuwe functie maken'
- Potloden
- Extra papier (optioneel)



[Download de werkbladen van 'Leren programmeren'](#)

## Verkennen

**Doelstelling:** Kennis van commando's en functies toepassen op liedjes door ze beschrijvende namen te geven.

**Discussie:** Laat de leerlingen verschillende liedjes bedenken en geef ze elk een beschrijvende functienaam.

**Voorbeeld:** Voor het nummer 'Twinkle, Twinkle, Little Star' zou de functieaanroep kunnen zijn: `singTwinkle()`, maar `singSong1()` zou geen goede naam zijn omdat het eerste nummer zou kunnen veranderen.

**Wat steken ze hiervan op:** Het benoemen van functies met beschrijvende namen is belangrijk omdat het code voor jezelf en anderen begrijpelijker maakt.

## Ontdekken

**Doelstelling:** De leerlingen maken een concert door in een concertfunctie verschillende songcommando's aan te roepen.

### Materiaal voor de begeleider:

- iPad of Mac
- Projector of scherm
- Whiteboard
- Stiften

### Aanwijzingen:

1. Help leerlingen functienamen te verzinnen voor een paar nummers, bijvoorbeeld `singHappyBirthday()`.
2. Kies als groep de volgorde van de liedjes.
3. Schrijf een functiedefinitie voor een concert, en vul de functie in met de songcommando's.

### Voorbeeld:

```
func createConcert() {
    singHappyBirthday()
    singTwinkleTwinkle()
    singMaryHadALittleLamb()
}
createConcert()
```

### Alternatief:

De leerlingen zingen in kleine groepjes, waarbij elk groepje zijn eigen lijst opstelt met liedjes, namen van de songfuncties en de volgorde van de liedjes. Elk groepje voert vervolgens zijn liedjes uit en maakt een video van zijn concert.

## Spelen

**Doelstelling:** De leerlingen lossen een eenvoudige vergelijking op, plaatsen een edelsteen op het antwoord en leiden Byte vervolgens door het raster met behulp van richtingscommando's.

**Vorbereiding:** De leerlingen gaan werken in groepjes van drie. Maak met afplaktape voor elk groepje een raster van vier bij vier op de vloer. Plaats de startpijl in een vierkant en plaats in elk resterend vierkant één nummer.

### Aanwijzingen:

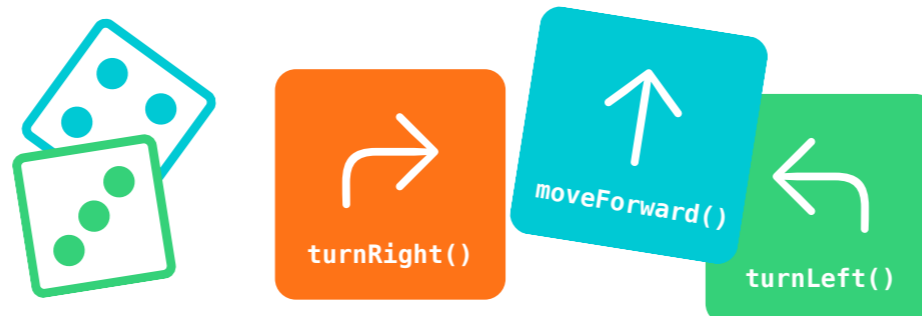
1. Deel het materiaal uit en deel de leerlingen in in groepjes van drie.
2. Lees alle rollen door en wijs aan elke leerling van het groepje een rol toe voor het eerste spel.
3. Laat de leerlingen het spel spelen, te beginnen met de rol van ontwerper.
4. Speel drie keer, waarbij de rolkaarten telkens rouleren.

### Rollen:

- Ontwerper: Gooi twee dobbelstenen. Tel met hulp van je medeleerlingen de twee getallen bij elkaar op en plaats de edelsteen op het vierkant waar de uitkomst staat.
- Programmeur: Plaats met hulp van je medeleerlingen de commandokaarten op of naast het raster om Byte naar de edelsteen te leiden en die te laten verzamelen.
- Tester: Begin met Byte op de pijl en volg de commandokaarten om Byte over het raster te verplaatsen. Als je de edelsteen verzamelt, ben je goed bezig! Als het niet lukt, moeten jullie de code samen verbeteren.

### Alternatief:

Als leerlingen individueel met u werken of thuis leren, kunnen ze dit spel in hun eentje spelen met behulp van de downloadbare alternatieve Keynote-activiteit.



### Materiaal voor de begeleider:

- Afplaktape
- Eén set gedrukte nummers voor elk raster

### Materiaal voor de leerling:

- Rolkaarten
- Commandokaarten: `moveForward()`, `turnLeft()`, `turnRight()` en `collectGem()`
- Edelsteen
- Byte
- Pijl
- Twee dobbelstenen

↓ [Download het materiaal](#)

↓ [Download de alternatieve activiteit](#)

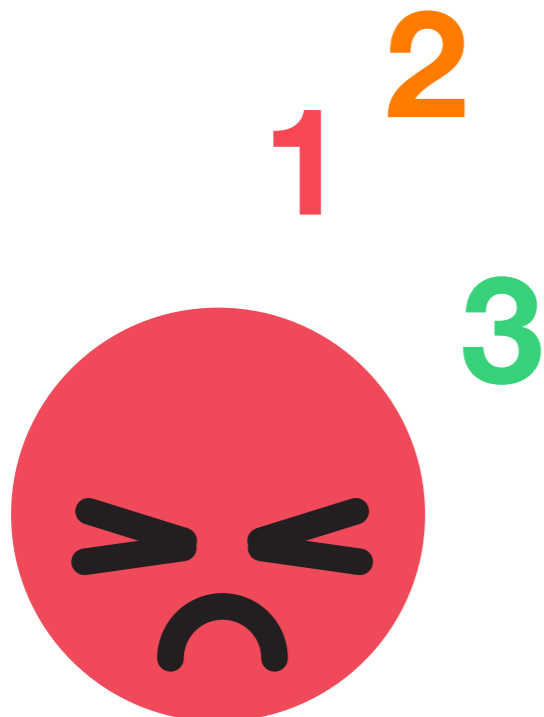


## Verkennen

**Doelstelling:** Leerlingen begrijpen dat er meestal meer dan één manier is om een probleem op te lossen.

**Discussie:** Laat de leerlingen nadenken over een probleem dat ze hebben gehad en vervolgens vertellen over de manieren waarop ze het hebben opgelost. Vraag de groep of iemand dat probleem op een andere manier zou oplossen. Verken verschillende problemen en oplossingen.

**Wat steken ze hiervan op:** Leerlingen leggen het verband met code en leren dat er meestal meer dan één manier is om een programmeerprobleem op te lossen.



## Ontdekken

**Doelstelling:** De leerlingen schrijven een functie voor de techniek die ze gebruiken om zichzelf te kalmeren, en geven deze functie een naam.

### Materiaal voor de leerling:

- Werkblad 'Mijn kalmeerfunctie'
- Potloden
- Kleurpotloden of -pennen

### Aanwijzingen:

Tip: Bij deze activiteit kunnen leerlingen het beste alleen werken, als dat kan.

1. Vraag de leerlingen om te brainstormen over manieren waarop ze zichzelf kunnen kalmeren, thuis of op school, als ze van streek zijn. Laat ze hun kalmeringstechnieken opdelen in stappen.
2. Deel het werkblad 'Mijn kalmeerfunctie' uit en vraag de leerlingen om de stappen van hun kalmeringstechniek te tekenen.
3. Laat de leerlingen een naam bedenken voor hun kalmeringstechniek. Ze kunnen daarvoor camelCase gebruiken, bijvoorbeeld `countToTen()`, of gewoon een kort zinnetje, zoals 'Tot tien tellen'.

### Als er tijd over is:

Unplugged: Laat de leerlingen hun kalmeringstechniek in kleine groepjes of voor de klas uitbeelden.

Met iPad: Leerlingen maken een video van hun kalmeringstechniek om die met de klas te delen.



[Download het werkblad 'Mijn kalmeerfunctie'](#)

## Spelen

**Doelstelling:** Leerlingen kunnen een functie schrijven die bestaat uit verschillende soorten commando's en die functie vervolgens gebruiken om een puzzel te voltooien.

### Aanwijzingen:

1. Projecteer de pagina 'Verzamelen, omzetten, herhalen' uit de playground 'Leren programmeren 1' op een scherm en wijs op de lege functie die leerlingen gaan helpen afmaken.
2. Verzamelen, omzetten, herhalen:
  - Neem de commando's `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` en `toggleSwitch()` door.
  - Laat de leerlingen proberen te bepalen welke onderdelen van de puzzel herhalend zijn, en laat ze vervolgens de functie in de app naar eigen inzicht afmaken en er een naam aan geven.
  - Laat de leerlingen een symbool voor de functie bedenken en het symbool en de functienaam noteren onder 'Commando's' op het werkblad.
  - Met het extra commando experimenteren leerlingen met manieren om Byte alle edelstenen te laten verzamelen en alle schakelaars te laten inschakelen. Ze noteren de commando's op het werkblad of op een apart vel papier.
  - Verzamel ideeën uit de klas en schrijf de code in de Swift Playgrounds-app om de puzzel te voltooien. Klik of tik op 'Voer mijn code uit'.
  - Probeer verschillende oplossingen.
  - Vier het met de hele klas. Dit was een moeilijke puzzel!




### Leren programmeren 1

#### Materiaal voor de begeleider:

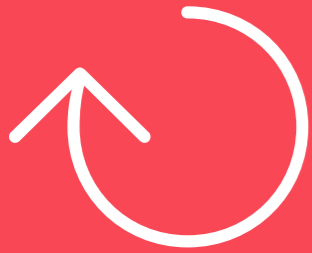
- iPad of Mac
- Swift Playgrounds-app
- Playground 'Leren programmeren 1'
- Projector of scherm

#### Materiaal voor de leerling:

- Werkblad 'Verzamelen, omzetten, herhalen'
- Potloden
- Extra papier (optioneel)


[Download het werkblad van 'Leren programmeren 1'](#)





# Lussen



## Overzicht

### Les 1: Herhalende bloemblaadjes

- Verkennen: Discussie over herhalende stappen in code en in het echte leven
- Ontdekken: Activiteit 'Herhalende bloemblaadjes'
- Spelen: 'Lussen gebruiken' en 'Doorlussen'

### Les 2: Hindernisbaan

- Verkennen: Discussie over eindpunten in een lus
- Ontdekken: Activiteit 'Hindernisbaan'
- Spelen: Vloerpuzzelspel

### Les 3: Drumpatronen

- Verkennen: Discussie over lussen in muziek
- Ontdekken: Activiteit 'Drumpatronen'
- Spelen: 'Naar de rand en terug' en 'Dans loop'

## Leerlingen kunnen

- In code een lus identificeren
- Een groot probleem of grote taak opdelen in kleinere stappen
- Een reeks commando's maken en die reeks herhalen door middel van een lus
- Instructies en code testen en debuggen

## Termen

- **Lus:** Een blok code dat een bepaald aantal keren wordt herhaald

## Verkennen

**Doelstelling:** Het idee van lussen in verband brengen met het echte leven.

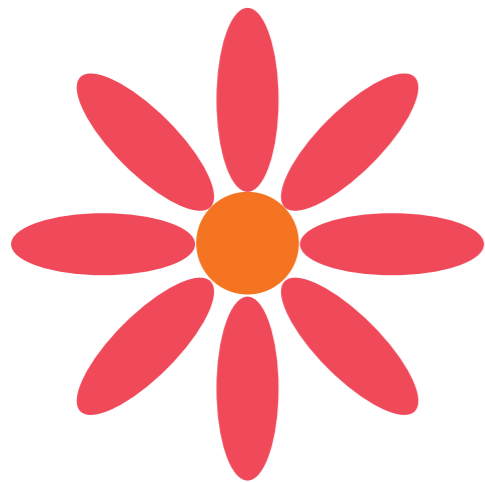
**Discussie:** Onderzoek momenten waarop een taak of stap in het echte leven herhaald wordt.

### Voorbeelden:

- Wandelen
- Fietsen
- Naaien, breien of haken

### Wat steken ze hiervan op:

Lussen herhalen een commando of een reeks commando's zo vaak als je opgeeft.



## Ontdekken

**Doelstelling:** De leerlingen gaan het begrip 'lussen' verkennen door een unieke bloem te maken.

### Materiaal voor de leerling:

- Werkblad 'Herhalende bloemblaadjes'
- Gekleurd papier
- Potloden
- Scharen
- Lijmstiften
- Dobbelstenen

### Aanwijzingen:

1. Leerlingen tekenen een enkel blaadje (ongeveer ter lengte van hun handpalm) op een stuk gekleurd papier en knippen het uit. Dit wordt het bloembladsjabloon voor hun bloem.
2. Elke leerling gooit vervolgens twee dobbelstenen, telt de getallen bij elkaar op en vult het ontbrekende getal in de lus in op het werkblad 'Herhalende bloemblaadjes'. Dit is het aantal bloemblaadjes dat de bloem krijgt.
3. Met behulp van hun bloembladsjabloon kunnen leerlingen hun bloemblaadje overtekenen op gekleurd papier en het juiste aantal bloemblaadjes voor hun bloem uitknippen.
4. Aan de hand van het werkblad 'Herhalende bloemblaadjes' kunnen de leerlingen hun bloem in elkaar zetten en de onderdelen op hun plaats lijmen.



[Download het werkblad 'Herhalende bloemblaadjes'](#)



## Spelen

**Doelstelling:** Leerlingen kunnen code binnen een lus schrijven om alle edelstenen te verzamelen.

### Aanwijzingen:

1. Projecteer de introductiepagina van het hoofdstuk 'For-lussen' van de playground 'Leren programmeren 1' op een scherm.
2. Inleiding:
  - Lees de pagina's met de hele klas door en stop zo nodig voor vragen.
3. Lussen gebruiken:
  - Laat leerlingen zien hoe portals werken en bekijk de opdrachten `moveForward()`, `turnLeft()`, `turnRight()` en `collectGem()`.
  - Laat de leerlingen experimenteren met manieren om Byte van de startpijl naar de edelstenen te leiden en ze te verzamelen, waarbij ze goed opletten welke opdrachten worden herhaald. Ze noteren de commando's op het werkblad of op een apart vel papier.
  - Verzamel ideeën uit de klas en schrijf de code in de Swift Playgrounds-app om Byte de opdracht te geven de eerste edelsteen te verzamelen en naar de portal te lopen.
  - Vraag de leerlingen hoeveel edelstenen er zijn en voeg dat getal toe aan de lus. Klik of tik op 'Voer mijn code uit'.
  - Probeer verschillende oplossingen.
  - Vier het met Byte!
4. Doorlussen:
  - Laat de leerlingen experimenteren met manieren om alle edelstenen te verzamelen, waarbij ze goed opletten welke opdrachten worden herhaald.
  - U voegt een `for`-lus toe met behulp van de codesuggesties onderaan de editor of door bovenaan het scherm op '+' te tikken.
  - Verzamel ideeën uit de klas en schrijf de code in Swift Playgrounds om de puzzel te voltooien. Klik of tik op 'Voer mijn code uit'.
  - Probeer verschillende ideeën.
  - Vier het met Byte!




### Leren programmeren 1

#### Materiaal voor de begeleider:

- iPad of Mac
- Swift Playgrounds-app
- Playground 'Leren programmeren 1'
- Projector of scherm

#### Materiaal voor de leerling:

- Werkbladen 'Lussen gebruiken' en 'Doorlussen'
- Potloden
- Extra papier (optioneel)

 [Download de werkbladen van 'Leren programmeren 1'](#)



## Verkennen

**Doelstelling:** Verkennen waarom een lus altijd een specifiek eindpunt moet hebben.

**Discussie:** Vraag de leerlingen zich een reuzenrad voor te stellen of een andere attractie die ze kennen. Wat zou er gebeuren als de machinist na vijf rondjes niet op de knop zou drukken om de rit te stoppen? Laat de leerlingen andere voorbeelden bedenken van wat er zou gebeuren als een lus niet wordt gestopt.

**Wat steken ze hiervan op:** Help leerlingen in te zien dat als ze geen eindpunt instellen voor een lus, de lus eeuwig wordt herhaald.

## Ontdekken

**Doelstelling:** Leerlingen ontdekken hoe lussen werken door een zelfontworpen hindernisbaan te doorlopen.

### Materiaal:

- Genoeg ruimte om lichamelijke activiteit te doen
- Rekwisieten voor een hindernisbaan
- Dobbelsteen

### Aanwijzingen:

1. Maak in de klas of buiten een korte hindernisbaan.
2. Werp een dobbelsteen en laat de leerlingen de hindernisbaan zo vaak herhalen als het aantal ogen van de worp.

### Alternatief:

Leerlingen bedenken een reeks bewegingen (bijvoorbeeld raak je tenen aan, spring, schop met je been). Werp een dobbelsteen. De leerlingen herhalen de reeks bewegingen zo vaak als het aantal ogen van de worp.



## Spelen

**Doelstelling:** Leerlingen kunnen een puzzel met een herhalend patroon creëren en de puzzel vervolgens als groep oplossen.

**Vorbereiding:** De leerlingen gaan werken in groepjes van drie. Maak met afplaktape voor elk groepje een raster van vier bij vier op de vloer.

### Aanwijzingen:

1. Deel het materiaal uit en deel de leerlingen in in groepjes van drie.
2. Lees alle rollen door en wijs aan elke leerling van het groepje een rol toe voor het eerste spel.
3. Laat de leerlingen het spel spelen, te beginnen met de rol van ontwerper.
4. Speel drie keer, waarbij de rolkaarten telkens rouleren.

### Rollen:

- Ontwerper: Plaats met de hulp van je medeleerlingen drie edelstenen in een herhalend patroon op het raster. Plaats de startpijl op het raster.
- Programmeur: Plaats met de hulp van je medeleerlingen de opdrachtkaarten op of naast het rooster om Byte naar de edelstenen te leiden en ze te verzamelen. Gebruik de luskaarten om de tester te vertellen hoe vaak hij/zij de opdrachten moet doorlopen.
- Tester: Begin met Byte op de pijl en volg de commandokaarten om Byte over het raster te verplaatsen. Als je alle edelstenen verzamelt, ben je goed bezig! Als het niet lukt, moeten jullie de code samen verbeteren.

### Alternatief:

Als leerlingen individueel met u werken of thuis leren, kunnen ze dit spel in hun eentje spelen met behulp van de downloadbare alternatieve Keynote-activiteit.

### Materiaal voor de begeleider:

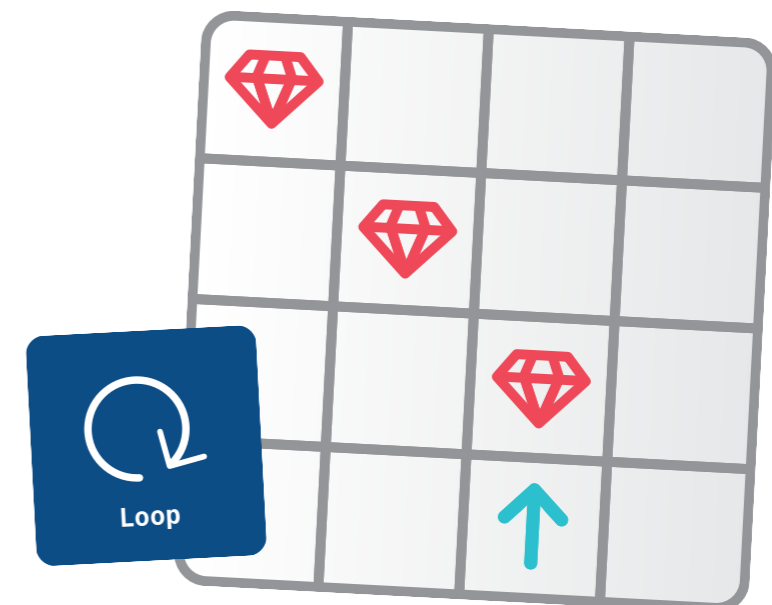
- Afplaktape

### Materiaal voor de leerling:

- Rolkaarten
- Commandokaarten: `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` en `Loop`
- Edelstenen
- Byte
- Pijl

↓ [Download het materiaal](#)

↓ [Download de alternatieve activiteit](#)



## Verkennen

**Doelstelling:** Herhalende patronen verkennen in muziek.

**Discussie:** Laat de leerlingen vertellen over instrumenten die ze spelen of liedjes die ze hebben gezongen. Vraag of ze ooit een beat of een refrein herhalen tijdens het spelen of zingen. Kunnen ze nog meer delen van een nummer of muziekstuk bedenken die zich herhalen?

### **Wat steken ze hiervan op:**

Versterking van het idee dat lussen uit twee delen bestaan:

- De commando's
- Het aantal keren dat ze moeten worden herhaald



## Ontdekken

**Doelstelling:** Leerlingen kunnen een drumpatroon herhalen, waardoor een verband wordt gelegd tussen code met lussen en een fysiek voorbeeld uit de echte wereld.

### **Materiaal:**

- Iets om op te drummen, zoals de vloer, je dijen of boeken
- Genoeg ruimte om in een kring te zitten

### **Aanwijzingen:**

1. Laat de leerlingen in een kring zitten.
2. Vertel de leerlingen dat ze de drumbeat die u maakt, net zo vaak moeten herhalen als het aantal vingers dat u opsteekt. Als u bijvoorbeeld vier vingers opsteekt, moeten de leerlingen de drumbeat vier keer herhalen en dan stoppen.
3. Ga om de beurt de cirkel rond of maak kleinere groepjes, zodat elke leerling de kans krijgt om de hoofddrummer te zijn.

### **Als er tijd over is:**

Laat de leerlingen trommels maken.

## Spelen

**Doelstelling:** Leerlingen roepen een aantal verschillende commando's binnen een lus aan en bepalen hoe vaak de lus moet worden aangeroepen.

### Aanwijzingen:

1. Projecteer de playground 'Leren programmeren 1' op een scherm. Ga naar de derde pagina, 'Naar de rand en terug', van het hoofdstuk 'For-lussen'.
2. Naar de rand en terug:
  - Neem de commando's `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` en `toggleSwitch()` door.
  - Laat leerlingen experimenteren met manieren om Byte van de startpijl naar elke gesloten schakelaar te leiden en deze om te zetten.
  - U voegt een `for`-lus toe met behulp van de codesuggesties onderaan de editor of door bovenaan het scherm op '+' te tikken.
  - Verzamel ideeën uit de klas en schrijf de code in de Swift Playgrounds-app om de puzzel te voltooien. Klik of tik op 'Voer mijn code uit'.
  - Probeer verschillende ideeën.
  - Vier het met Byte!
3. Sluit 'Leren programmeren 1' en open de playground 'MeeBot Leer Dansen'. Ga hierin naar de pagina 'Dans loop'. (Er is geen werkblad voor leerlingen voor deze pagina van de playground.)
4. Danslussen:
  - Laat de leerlingen als groep, in paren of individueel op hun eigen iPad werken om de lus te voltooien en de robot te zien dansen.
  - Vraag de leerlingen om hun dansen te delen, of maak er een paar met de hele klas.
  - Dans met de robot!



**Leren  
programmeren 1**



**MeeBot  
Leer dansen**

### Materiaal voor de begeleider:

- iPad of Mac
- Swift Playgrounds-app
- Playground 'Leren programmeren 1'
- Playground 'MeeBot Leer Dansen'
- Projector of scherm

### Materiaal voor de leerling:

- Werkblad 'Naar de rand en terug'
- Potloden
- iPad-devices (optioneel)
- Extra papier (optioneel)

↓ [Download het werkblad van 'Leren programmeren'](#)



# Variabelen



## Overzicht

### Les 1: Zinken of drijven

- Verkennen: Discussie over het bijwerken van een variabele
- Ontdekken: Activiteit 'Zinken of drijven'
- Spelen: 'Op de hoogte blijven' en 'Voorbeeldspel'

### Les 2: Woordspel

- Verkennen: Discussie over soorten antwoorden op vragen
- Ontdekken: Activiteit 'Woordspel'
- Spelen: Vloerpuzzelspel

### Les 3: Alles over mij

- Verkennen: Discussie over vragen beantwoorden met een lijst
- Ontdekken: Activiteit 'Alles over mij'
- Spelen: Een lus gebruiken

## Leerlingen kunnen

- Een variabelenaam koppelen aan een gegeven waarde
- De waarde wijzigen die aan een variabele is toegewezen
- Weten welke Swift-typen je aan een variabele kunt toewijzen, inclusief true/false (Booleaanse waarden), getallen (Ints), woorden (Strings), kleuren (kleurliterals) en afbeeldingen (afbeeldingsliterals)
- Instructies en code testen en debuggen

## Termen

- **Variabele:** Een benoemde container waarin een waarde is opgeslagen die kan worden gewijzigd
- **Gegevens:** Informatie
- **Booleaanse waarde:** Een type dat de waarde true of false heeft

## Verkennen

**Doelstelling:** Het begrip ‘variabelen’ verkennen door objecten te tellen en het variabelegetal bij te werken.

### Materiaal voor de begeleider:

- Whiteboard
- Stift
- Gum
- Bakje
- Vijf potloden (of vijf stuks van hetzelfde soort voorwerp)

### Aanwijzingen:

1. Begin met een variabele-instructie op het whiteboard te schrijven om de objecten bij te houden.
  - Voorbeeld: `var numberOfPencils = 0`
2. Houd een leeg bakje omhoog en vertel de leerlingen dat het bakje de variabele `numberOfPencils` voorstelt.
3. Leg één potlood in het bakje en vraag de leerlingen wat nu de waarde van de variabele is. Als ze correct antwoorden, veegt u de `0` uit en schrijft u `1` op.
4. Ga door totdat alle potloden zijn toegevoegd en de code luidt: `var numberOfPencils = 5`.
5. Haal nu potloden uit het bakje en werk de variabele telkens bij.

**Wat steken ze hiervan op:** De leerlingen begrijpen dat variabelen een stukje informatie opslaan. In dit geval is de informatie een getal en het getal geeft aan hoeveel potloden er in het bakje liggen.

## Ontdekken

**Doelstelling:** Met behulp van voorwerpen die ze hebben gevonden, voeren leerlingen experimenten uit om te bepalen of iets zinkt of blijft drijven, waarna ze de gegevens vastleggen met behulp van afbeeldingen (afbeeldingsliterals) en true/false-waarden (Booleaanse waarden).

### Materiaal voor de leerling:

- iPad-devices
- Keynote-app
- Werkblad ‘Zinken of drijven’
- Emmer met water
- Verschillende voorwerpen om te testen

### Aanwijzingen:

1. Deel de leerlingen in in kleine groepjes.
2. Laat ze verschillende voorwerpen verzamelen om te testen.
3. Vraag de leerlingen voor elk voorwerp om:
  - Een foto te maken van het voorwerp en die toe te voegen aan het werkblad.
  - Het voorwerp te testen in het water.
  - Het resultaat te noteren op het werkblad door een kringetje te zetten om waar of niet waar.



[Download het werkblad ‘Zinken of drijven’](#)

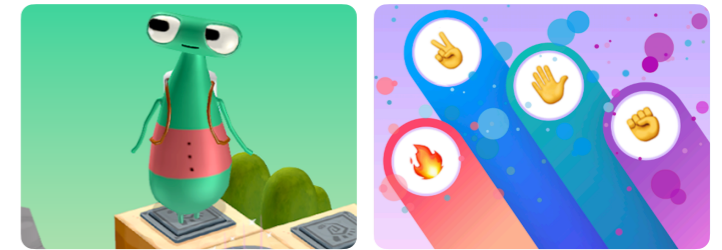
## Spelen

**Doelstelling:** Leerlingen kunnen variabelen maken en bijwerken in twee verschillende codeercontexten.

### Aanwijzingen:

1. Projecteer de playground ‘Leren programmeren 2’ op een scherm. Ga naar het hoofdstuk ‘Variabelen’.
2. Inleiding:
  - Lees de pagina’s met de hele klas door en stop zo nodig voor vragen.
3. Op de hoogte blijven:
  - Laat de leerlingen experimenteren met manieren om Hopper van de startpijl naar de edelsteen te leiden en deze te verzamelen. Ze noteren de commando’s op het werkblad of op een apart vel papier.
  - Verzamel ideeën uit de klas en schrijf de code in de Swift Playgrounds-app om de puzzel te voltooien. Klik of tik op ‘Voer mijn code uit’.
  - Probeer verschillende ideeën.
  - Vier het met Hopper!
4. Sluit ‘Leren programmeren 2’ en ga verder met de laatste pagina van de playground ‘Steen, papier, schaar’, getiteld ‘Voorbeeldgame’. (Er is geen werkblad voor leerlingen voor deze pagina van de playground.)
5. Voorbeeldspel:
  - Klik of tik op ‘Voer mijn code uit’ om het spel te spelen voordat u iets wijzigt.
  - Beslis als groep welke delen van het spel u wilt aanpassen. Leuk om aan te passen zijn bijvoorbeeld `game.roundsToWin`, `game.challenger.emoji`, `game.addOpponent` en `game.roundPrize`.
  - Speel het spel meerdere keren en verander telkens iets anders.

**Als er tijd over is:** Veel variabelen zijn vastgelegd in het bestand `Game.swift`. Als leerlingen nieuwsgierig zijn waarom sommige variabelen niet voorafgegaan worden door `var`, kunt u het bestand `Game.swift` voor ze openen om te laten zien waar de eigenschappen van de game zijn aangemaakt.



**Leren  
programmeren 2**

**Steen, papier,  
schaar**

### Materiaal voor de begeleider:

- iPad of Mac
- Swift Playgrounds-app
- Playground ‘Leren programmeren 2’
- Playground ‘Steen, papier, schaar’
- Projector of scherm

### Materiaal voor de leerling:

- Werkblad ‘Op de hoogte blijven’
- Potloden
- Extra papier (optioneel)

↓ [Download het werkblad van ‘Leren programmeren’](#)

## Verkennen

**Doelstelling:** Verken verschillende soorten antwoorden in de echte wereld en relateer ze aan verschillende Swift-typen, zoals ja/nee of true/false (Booleaanse waarden), getallen (Ints), woorden (Strings), kleuren (kleurliterals) en afbeeldingen (afbeeldingsliterals).

### Materiaal voor de begeleider:

- Whiteboard
- Stiften

**Discussie:** Bedenk klassikaal een aantal vragen die verschillende typen antwoord vereisen, en schrijf ze op het bord.

### Voorbeelden:

- Welke kleur hebben je ogen? → kleur
- Heb je een huisdier? → ja/nee
- Heb je broertjes of zusjes? → ja/nee
- Hoe oud ben je? → getal
- Hoe heet je? → woord

**Wat steken ze hiervan op:** Leg uit dat variabelen ook verschillende typen hebben, waaronder getallen, woorden, kleuren, afbeeldingen en ja/nee-antwoorden. Afhankelijk van hoe een variabele wordt aangemaakt, moet de variabele hetzelfde type behouden, ook als die wordt bijgewerkt naar iets nieuws. Bijvoorbeeld: `var mijnLeeftijd = 8` kan veranderen in 9, maar niet in "negen".

## Ontdekken

**Doelstelling:** Leerlingen kunnen een woordspel voltooien door het juiste antwoordtype in te vullen.

### Materiaal voor de leerling:

- 'Woordspel'-werkbladen
- Potloden
- Kleurpotloden

### Aanwijzingen:

Laat de leerlingen in kleine groepjes een of meer woordspelletjes uitwerken. Idealiter zou in elke groepje ten minste één leerling moeten zitten die kan lezen of die kan helpen. Als de leerlingen geen van allen kunnen lezen, doet u een paar spelletjes samen met de hele groep.

**Als er tijd over is:** Als leerlingen daartoe in staat zijn, laat u ze een woordspel maken dat door een partner wordt ingevuld. Moedig ze aan om voor de lege plekken cijfers, woorden, kleuren, afbeeldingen en ja/nee-antwoorden te gebruiken.



[Download de Woordspel-werkbladen](#)



## Spelen

**Doelstelling:** Leerlingen kunnen Byte verschillende edelstenen laten verzamelen, alle edelstenen in een container leggen en een variabele bijwerken.

**Vorbereiding:** De leerlingen gaan werken in groepjes van drie. Maak met afplaktape voor elk groepje een raster van vier bij vier op de vloer.

### Aanwijzingen:

1. Deel het materiaal uit en deel de leerlingen in in groepjes van drie.
2. Lees alle rollen door en wijs aan elke leerling van het groepje een rol toe voor het eerste spel.
3. Laat de leerlingen het spel spelen, te beginnen met de rol van ontwerper.
4. Speel drie keer, waarbij de rolkaarten telkens rouleren.

### Rollen:

- Ontwerper: Leg meerdere edelstenen en de startpijl op het raster.
- Programmeur: Plaats met de hulp van je medeleerlingen de commandokaarten op of naast het raster om Byte naar de edelstenen te leiden en ze te verzamelen.
- Tester: Begin met Byte op de pijl en volg de commando's om Byte over het raster te verplaatsen en de verzamelde edelstenen aan de container toe te voegen. Als je alle edelstenen hebt verzameld, werk je de variabele `numberOfGems` op de container bij, en vier je een feestje! Als je ze niet allemaal hebt verzameld, moeten jullie als team de code samen verbeteren.

### Alternatief:

Als leerlingen individueel met u werken of thuis leren, kunnen ze dit spel in hun eentje spelen met behulp van de downloadbare alternatieve Keynote-activiteit.

### Materiaal voor de begeleider:

- Afplaktape

### Materiaal voor de leerling:

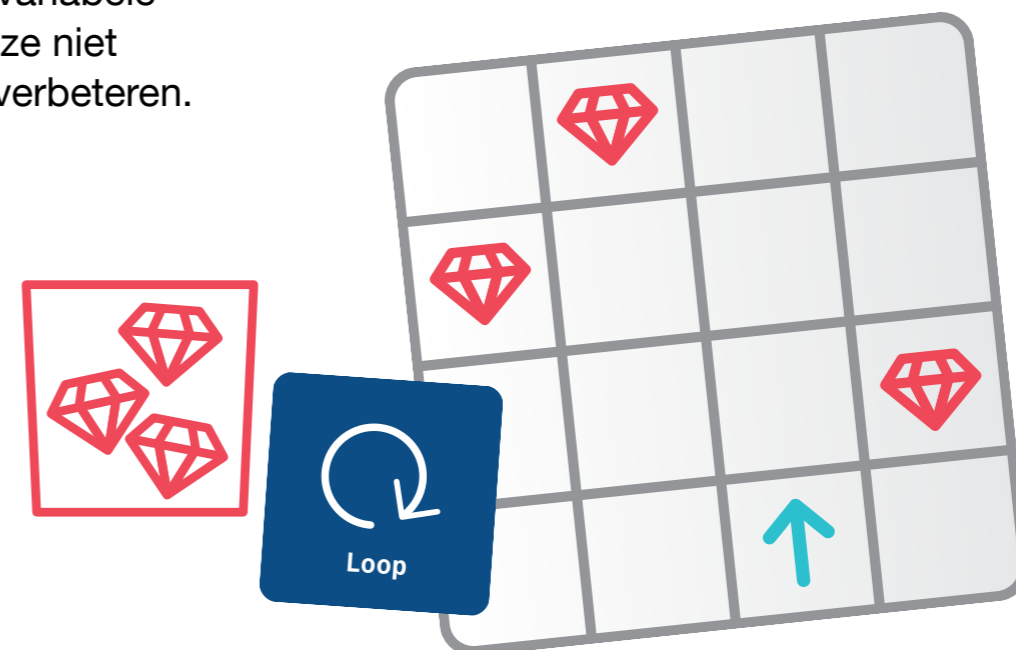
- Rolkaarten
- Commandokaarten: `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` en `Loop`
- Edelstenen
- Byte
- Pijl
- Bakje met label: `var numberOfGems = _____`
- Pen



[Download het materiaal](#)



[Download de alternatieve activiteit](#)











## Verkennen

**Doelstelling:** Verkennen hoe je lijsten (of *arrays*) gebruikt bij het aanmaken van variabelen.

**Discussie:** Wat zou er gebeuren als een werkblad de leerlingen zou vragen naar de naam van hun broer of zus en ze hebben er meer dan één? Verzamel ideeën uit de klas. Als leerlingen voorstellen om een lijst te maken, vertel ze dan dat programmeurs dat ook doen! Als een variabele meer dan één antwoord heeft, moeten leerlingen een lijst maken.

Laat de leerlingen vragen bedenken die meerdere antwoorden kunnen hebben.

### Voorbeelden:

- Namen van vrienden/vriendinnen -> Rosa, Sam, Joy
- Leeftijden van de leerlingen —> 7, 8, 7, 8, 7, 8, 9, 7, 8, 9, 8
- Favoriete kleuren —>  ,  ,  ,  , 
- Favoriete dieren —>  ,  ,  , 

**Wat steken ze hiervan op:** Lijsten die leerlingen in code maken, zijn net als lijsten in een zin.

## Ontdekken

**Doelstelling:** De leerlingen kunnen variabelen invullen om dingen over zichzelf en een partner te beschrijven. Misschien kunnen leerlingen zelfs een array gebruiken als variabeletype.

### Materiaal voor de leerling:

- Werkbladen ‘Alles over mij’ en ‘Alles over jou’
- Potloden
- Kleurpotloden

### Aanwijzingen:

1. Laat de leerlingen het werkblad ‘Alles over mij’ invullen.
  - Als leerlingen meer dan één broer of zus of huisdier hebben, laat ze dan een lijst maken met items gescheiden door komma's.
2. Deel de leerlingen in in paren om het werkblad ‘Alles over jou’ in te vullen.

**Alternatief:** Leerlingen kunnen hun iPad en Keynote gebruiken om het werkblad in te vullen, foto's te maken voor de beeldantwoorden en de kleurliterals in te kleuren met behulp van de opmaakopties.



[Download de werkbladen bij ‘Alles over ...’](#)

## Spelen

**Doelstelling:** Leerlingen kunnen een variabele in code identificeren en ontdekken manieren waarop ze arrays kunnen gebruiken in combinatie met lussen.

### Aanwijzingen:

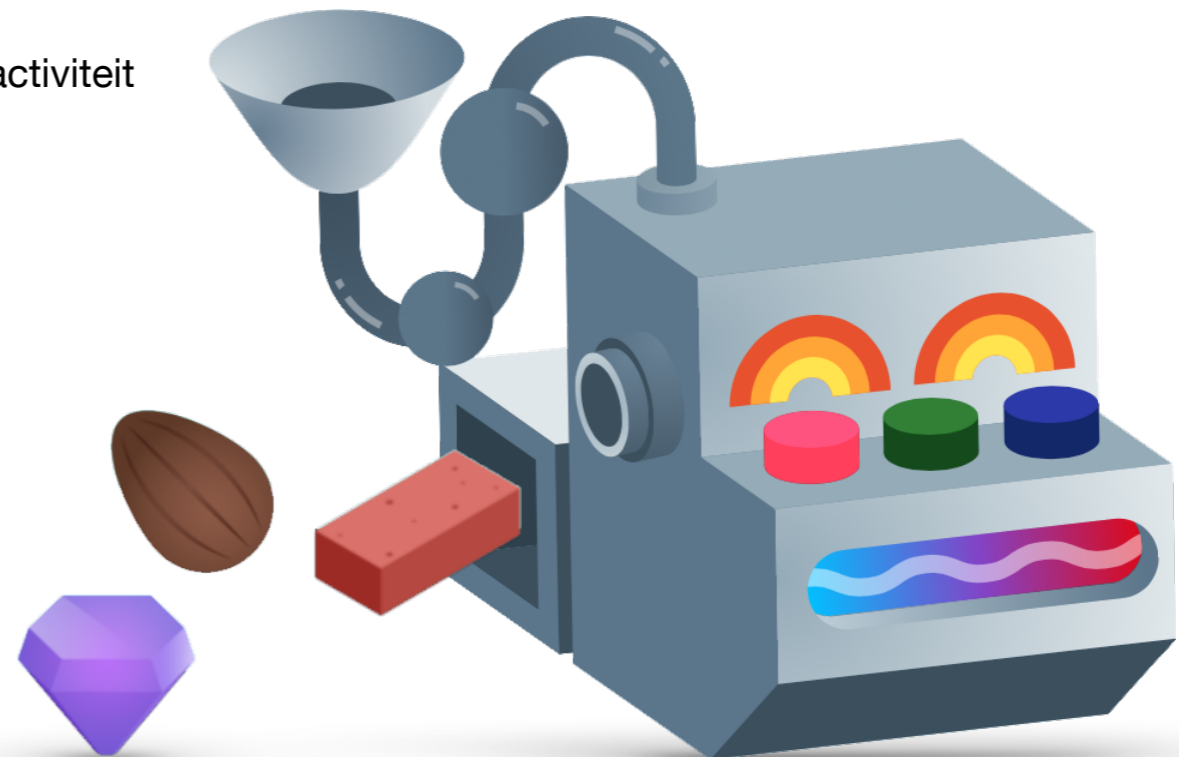
1. Projecteer de playground 'Programmeermachine' op een scherm.
2. Inleiding:
  - Lees de pagina's met de hele klas door en stop zo nodig voor vragen.
  - Optioneel: Speel de eerste twee pagina's ('De machine verkennen' en 'Combineren met kleuren').
3. Een lus gebruiken:
  - Op deze pagina combineren leerlingen hun kennis van lussen met variabelen.
  - Kijk of leerlingen in de code de variabele kunnen identificeren die een array gebruikt.
  - Klik of tik op 'Voer mijn code uit' om te zien wat de machine maakt.
  - Ga verder met de tweede stap in de instructies en werk de code bij met een tweede variabele, items en een geneste lus. Klik of tik nogmaals op 'Voer mijn code uit' om te zien wat de machine maakt.
  - Opmerking: Probeer deze pagina eerst zelf voordat u de activiteit met leerlingen doet.



**Programmeermachine**

### Materiaal voor de begeleider:

- iPad of Mac
- Swift Playgrounds-app
- Playground 'Programmeermachine'
- Projector of scherm



# App-ontwerp



## Verkennen

**Doelstelling:** Bekende apps verkennen op verschillende apparaten.

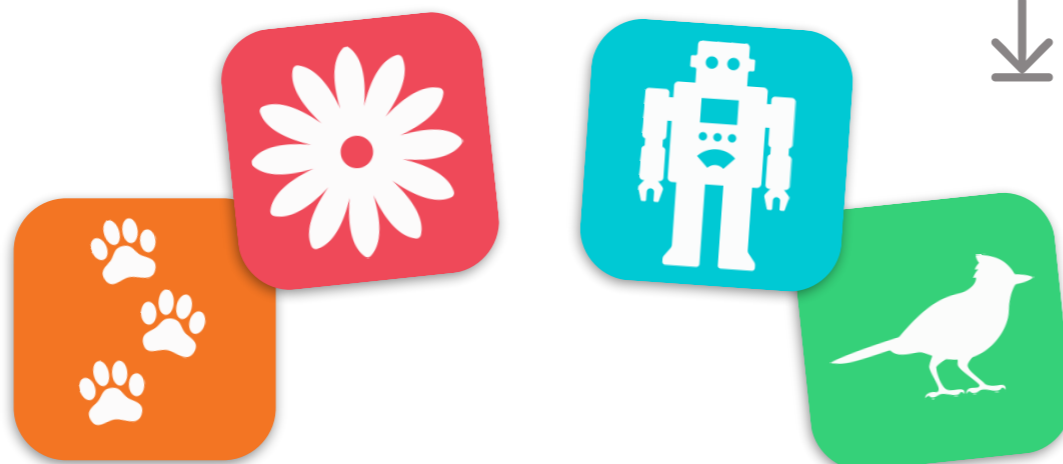
**Aanwijzingen:** Start een discussie over apps die leerlingen thuis of op school op de iPad gebruiken. Praat vervolgens over apps die zij of hun ouders of verzorgers gebruiken op devices thuis.

**Wat steken ze hiervan op:** Leerlingen worden er nogmaals op gewezen dat apps niet alleen op telefoons staan, maar ook op horloges, tablets, computers en zelfs tv.

**Als er tijd over is:** Ga dieper in op enkele voorbeelden van apps en vraag leerlingen voor wie de app is ontworpen, wat hij doet en waarom ze denken dat hij gemaakt is.

### Voorbeeld:

- App: Swift Playgrounds
- Voor wie: Mensen die meer willen weten over Swift
- Wat het doet: Helpt mensen te leren programmeren door middel van puzzels en lessen
- Waarom het is gemaakt: Om mensen met weinig of geen programmeerkennis te leren programmeren



## Ontdekken

**Doelstelling:** Leerlingen voorbereiden op het ontwerpen van eigen apps door een bekende app te analyseren.

### Materiaal voor de leerling:

- iPad-devices
- Wat is een app? (werkblad)
- Potloden
- Kleurpotloden of -pennen

### Aanwijzingen:

1. Deel de leerlingen in in kleine groepjes of laat ze individueel werken.
2. Vraag de leerlingen om een iPad-app te kiezen.
3. Laat ze de app verkennen aan de hand van het werkblad 'Wat is een app?'.
4. Vraag de leerlingen om hun bevindingen over de app te delen met de hele groep of met hun partners.

**Tip voor begeleiders:** Hoe jonger de leerlingen zijn, hoe meer hulp ze nodig hebben om dit werkblad in te vullen. Voor leerlingen van 5 tot 6 jaar zou u twee of drie apps met de hele groep kunnen doen.



[Download het werkblad 'Wat is een app?'](#)

## Spelen

**Doelstelling:** Leerlingen ontwerpen hun eigen apps!

### Materiaal voor de leerling:

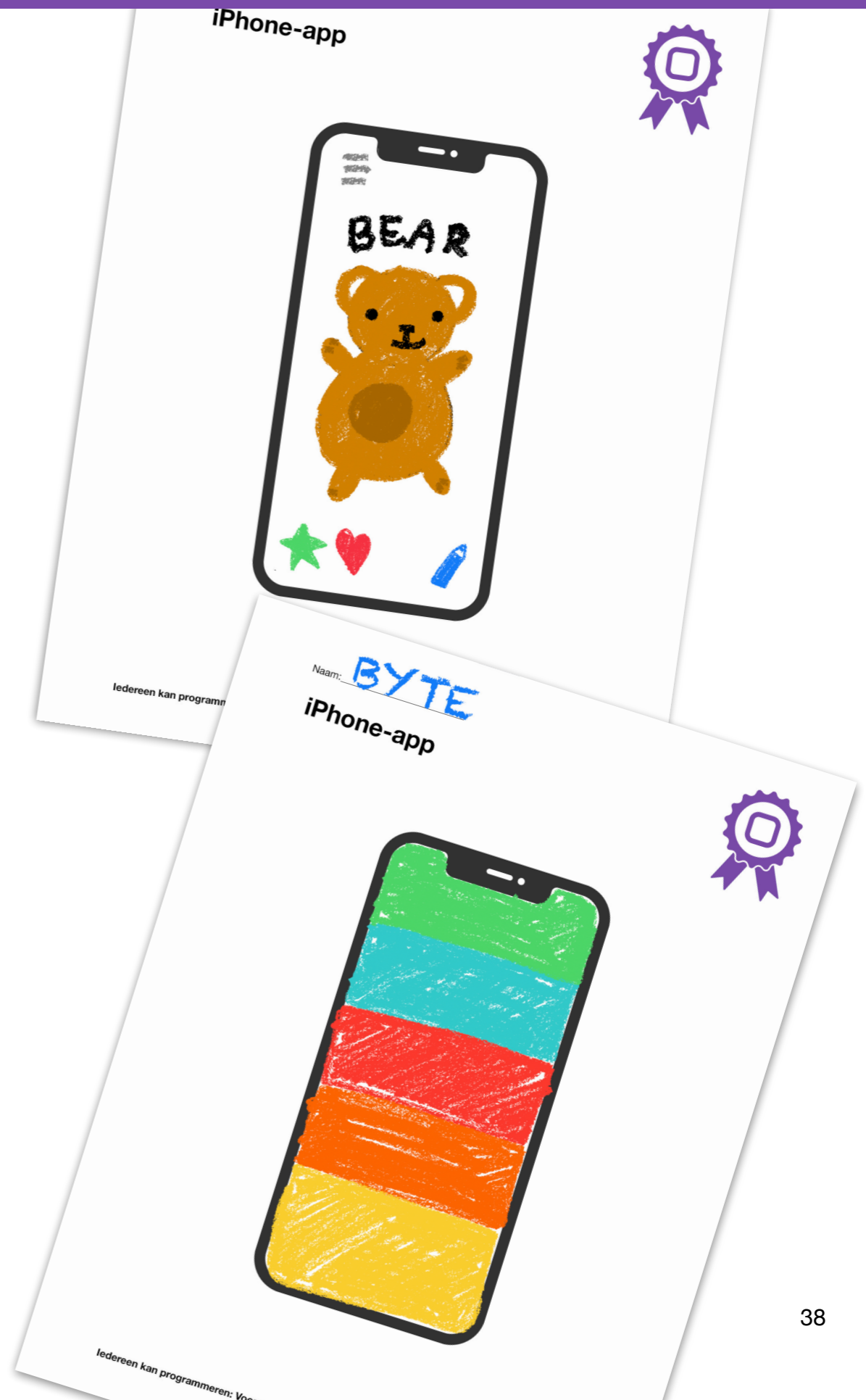
- Werkblad 'Mijn app-ontwerp'
- Apparaatsjablonen
- Extra papier
- Potloden
- Kleurpotloden of -pennen

### Aanwijzingen:

1. Deel de leerlingen in in kleine groepjes of laat ze individueel werken.
2. Neem het werkblad 'Mijn app-ontwerp' met de leerlingen door om ze te begeleiden bij hun eerste stappen op weg naar het ontwerpen van een app.
3. Laat leerlingen met behulp van extra papier of de apparaatsjablonen een prototype maken van de pagina's van hun apps.
4. Laat de leerlingen een definitieve versie van hun app-prototypes maken met behulp van de apparaatsjablonen.
5. Nodig alle leerlingen of groepjes uit om hun ideeën aan de hele groep te presenteren.

↓ [Download het werkblad 'Mijn app-ontwerp'](#)

↓ [Download de apparaatsjablonen](#)



# Informatiebronnen voor begeleiders



## Woordenlijst

- **Booleaanse waarde:** Een type dat de waarde true of false heeft
- **Bug:** Een programmeerfout
- **Commando:** Code die een app opdracht geeft om een bepaalde bewerking uit te voeren
- **Gegevens:** Informatie
- **Debuggen:** Het zoeken en corrigeren van programmeerfouten
- **Functie:** Een benoemde set commando's die op elk gewenst moment kan worden uitgevoerd
- **Lus:** Een blok code dat een bepaald aantal keren wordt herhaald
- **Wijzigen:** Veranderen
- **Sequentie:** De volgorde waarin dingen gebeuren
- **Stap:** Eén bewerking in een groter proces
- **Omzetten:** In- of uitschakelen
- **Variabele:** Een benoemde container waarin een waarde is opgeslagen die kan worden gewijzigd



## Voorbeeldantwoorden

De volgende pagina's bieden een mogelijke oplossing voor elke Swift Playgrounds-puzzel, maar de puzzels kunnen op meer dan één manier worden opgelost. Moedig leerlingen aan om verschillende manieren uit te proberen waarop ze Byte of de andere figuurtjes kunnen aansturen.

Leerlingen mogen trots zijn op alle soorten programmeerwerk en alle doelen die ze hebben. Sommige leerlingen willen misschien niet alleen de edelstenen verzamelen, maar de hele puzzelruimte verkennen, terwijl andere leerlingen misschien onderweg naar de edelstenen zo vaak mogelijk willen ronddraaien. Vergeet niet dat programmeren leuk hoort te zijn!



### Leren programmeren 1

#### Hoofdstuk 'Commando's'

Commando's geven

```
moveForward()
moveForward()
moveForward()
collectGem()
```

#### Hoofdstuk 'Commando's'

Commando toevoegen

```
moveForward()
moveForward()
turnLeft()
moveForward()
moveForward()
collectGem()
```

#### Hoofdstuk 'Functies'

Nieuw gedrag samenstellen

```
moveForward()
moveForward()
moveForward()
turnLeft()
turnLeft()
turnLeft()
moveForward()
moveForward()
moveForward()
collectGem()
```

#### Hoofdstuk 'Functies'

Nieuwe functie maken

```
func turnRight() {
    turnLeft()
    turnLeft()
    turnLeft()
}

moveForward()
turnLeft()
moveForward()
turnRight()
moveForward()
turnRight()
moveForward()
turnLeft()
moveForward()
toggleSwitch()
```



**Leren programmeren 1**

**Hoofdstuk 'Functies'**

Verzamelen, omzetten, herhalen

```
func collectToggle() {
  moveForward()
  collectGem()
  moveForward()
  toggleSwitch()
  moveForward()
}
```

```
collectToggle()
turnLeft()
collectToggle()
moveForward()
turnLeft()
collectToggle()
turnLeft()
collectToggle()
```

**Hoofdstuk 'Lussen'**

Lussen gebruiken

```
for i in 1 ... 5 {
  moveForward()
  moveForward()
  collectGem()
  moveForward()
}
```

**Hoofdstuk 'Lussen'**

Doorlussen

```
for i in 1 ... 4 {
  moveForward()
  collectGem()
  moveForward()
  moveForward()
  moveForward()
  turnRight()
}
```

**Hoofdstuk 'Lussen'**

Naar de rand en terug

```
for i in 1 ... 4 {
  moveForward()
  moveForward()
  toggleSwitch()
  turnLeft()
  turnLeft()
  moveForward()
  moveForward()
  turnLeft()
}
```

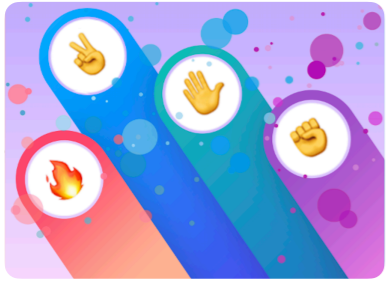


**Leren programmeren 2**

**Hoofdstuk 'Variabelen'**

Op de hoogte blijven

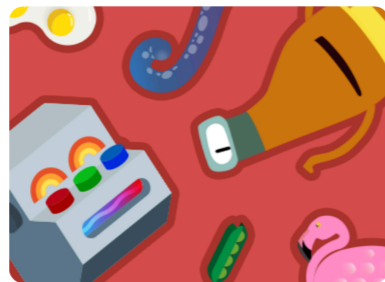
```
var gemCounter = 0
moveForward()
moveForward()
collectGem()
gemCounter += 1
```



**Steen, papier, schaar**

**Voorbeeldspel**

Er is geen voorbeeldoplossing voor deze pagina omdat het spel volledig aanpasbaar is – je kunt het spelen zoals je wilt!



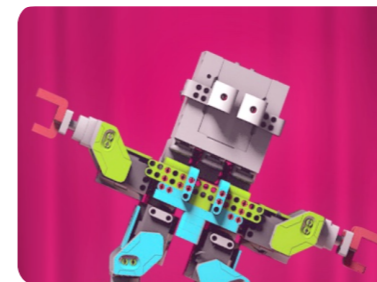
**Programmeermachine**

**Een lus gebruiken**

```
var colors = [Light.red, Light.green, Light.blue]

var items = [Item.metal, Item.stone, Item.cloth, Item.dirt, Item.DNA, Item.spring, Item.wire, Item.egg, Item.tree, Item.gear, Item.seed, Item.crystal, Item.mushroom, Item.unidentifiedLifeForm]

for item in items {
  setItemA(item)
  setItemB(.dirt)
  switchLightOn(.green)
  forgeItems()
}
```



**MeeBot Leer dansen**

**Basisbewegingen**

```
bendAndTwist()
happy()
moveBackward()
shake()
skip()
split()
swagger()
twist()
```

**Dans loop**

```
for i in 1 ... 5 {
  bend()
  bend(beats: 2)
  bendAndTwist()
  moveBackward(beats: 9)
}
```

