

# LEGO® Education WeDo 2.0 Computational Thinking

Handleiding voor  
de leerkracht



WeDo 2.0

# Inhoud

**Introductie van WeDo 2.0  
Computational Thinking**

**3-11**

**WeDo 2.0 in het leerplan**

**12-28**

**Evalueren met WeDo 2.0**

**29-41**



# Vaardig worden in computational thinking met WeDo 2.0

In dit hoofdstuk ontdek je hoe je WeDo 2.0 kunt gebruiken om computational thinking-vaardigheden te ontwikkelen in een wetenschappelijke context.





# Vaardig worden in computational thinking met LEGO® Education WeDo 2.0-projecten

LEGO® Education presenteert je met trots deze projecten, die speciaal zijn ontwikkeld om in de basisschool te gebruiken om de computational thinking-vaardigheden van de leerlingen te ontwikkelen.

Computational thinking heeft betrekking op een set vaardigheden die iedereen kan toepassen om alledaagse problemen op te lossen. In WeDo 2.0 worden deze vaardigheden tijdens de verschillende fasen van elk project ontwikkeld. In elk project zijn mogelijkheden voor ontwikkeling geïdentificeerd. Jij bepaalt zelf welke het meest relevant zijn voor jou en je leerlingen.

In elk project van WeDo 2.0 wordt het gebruik van LEGO stenen gecombineerd met een programmeertaal met pictogrammen, waardoor je leerlingen problemen leren op te lossen en kennismaken met programmeerprincipes.

Met WeDo 2.0 worden de computational thinking-vaardigheden van je leerlingen ontwikkeld via codeeractiviteiten waarmee hun creaties tot leven worden gebracht, die zorgen voor veel leerplezier en zin om meer te ontdekken.





## Informatica, computational thinking, programmeren

Wetenschap en techniek bestaan al sinds het begin van de mensheid, maar informatica is een veel jongere discipline. Deze jonge discipline heeft niet alleen een invloed gehad op de manier waarop we wetenschap en techniek benaderen, maar ook op onze levensstijl.

Informatica is een STEM-discipline en heeft veel van dezelfde eigenschappen als wetenschap, technologie, techniek en wiskunde.

Alle STEM-disciplines bieden de mogelijkheid om een bepaalde denkwijze en een set levenslang te gebruiken competenties te ontwikkelen. Onder deze competenties vinden we het vermogen om vragen te stellen, oplossingen te bedenken en resultaten te presenteren.

Eén van deze competenties is computational thinking. Het is een denkwijze en een manier om problemen op te lossen.

Computational thinking is te omschrijven als een groep vaardigheden, waarvan één algoritmisch denken is. Coderen betekent het creëren van een algoritme.

Coderen is daarom één van de manieren om computational thinking te ontwikkelen in een STEM-context.

## STEM-disciplines

Wetenschap, technologie, techniek,  
wiskunde en informatica

**Een bepaalde denkwijze en een set levenslang te gebruiken competenties ontwikkelen**

1. Vragen stellen en problemen oplossen.
2. Modellen gebruiken.
3. Prototypes ontwerpen.
4. Onderzoeken.
5. Gegevens analyseren en interpreteren.
6. Computational thinking gebruiken.

- a. Ontleden
- b. Abstractie
- c. Algoritmisch denken (code)
- d. Evalueren
- e. Veralgemeeniseren

7. Argumenteren op basis van bewijzen.
8. Informatie verkrijgen, beoordelen en doorgeven.



## Wat is computational thinking?

De term “computational thinking” werd als eerste gebruikt door Seymour Papert, maar kreeg echte bekendheid door professor Jeannette Wing. Zij definieerde computational thinking als volgt:

*“de denkprocessen waarmee problemen en hun oplossingen zodanig worden geformuleerd dat ze kunnen worden gepresenteerd in een vorm die effectief kan worden uitgevoerd door een informatieverwerkende tussenpersoon.” (Wing, 2011)*

Computational thinking wordt op diverse gebieden, in verschillende situaties en in het dagelijks leven gebruikt. Je vindt computational thinking-vaardigheden terug in de wetenschap, techniek, wiskunde en informatica. De onderstaande vaardigheden zijn computational thinking-vaardigheden:

### Ontleden

Ontleden is de vaardigheid een probleem in kleinere onderdelen op te delen om gemakkelijker een oplossing te vinden. Op die manier wordt het gemakkelijker het probleem aan iemand anders uit te leggen of het in taken op te delen. Ontleden leidt vaak tot veralgemeniseren.

Voorbeeld: Wanneer je op vakantie gaat, kan de voorbereiding (of het project) in subtaken worden opgedeeld: een vlucht boeken, een hotel reserveren, je koffers inpakken, enz.

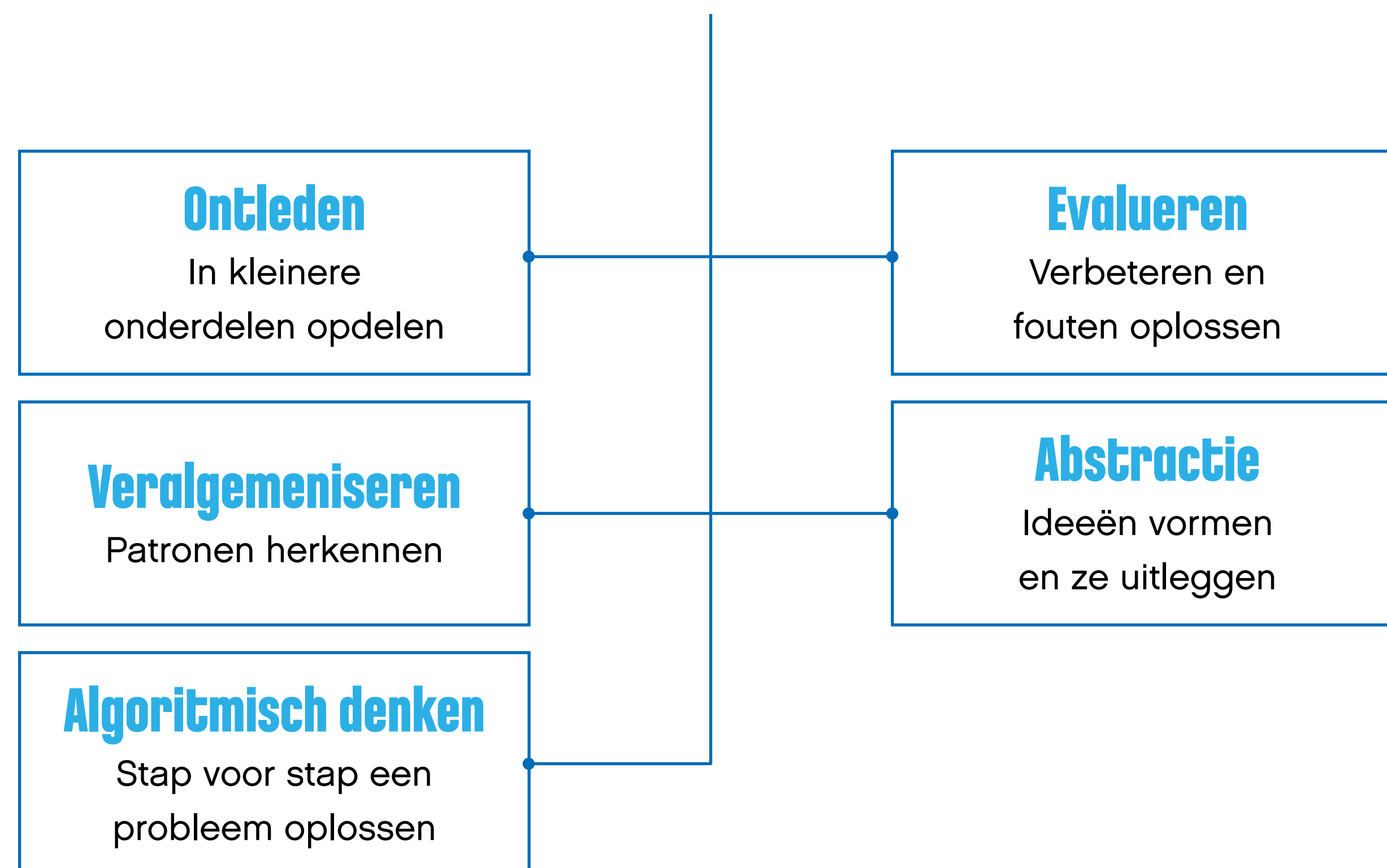
### Veralgemeniseren (patroonherkenning)

Veralgemeniseren is de vaardigheid om bekende delen van een taak te herkennen, evenals delen die ergens anders in voorkwamen. Dit leidt vaak tot eenvoudiger manieren om algoritmen te creëren.

Voorbeeld: Verkeerslichten werken door het eindeloos herhalen van dezelfde reeks acties.

## Computational thinking

Manieren waarop we problemen oplossen





## Wat is computational thinking?

### Algoritmisch denken

Algoritmisch denken is de vaardigheid om een geordende reeks stappen te creëren met als doel een probleem op te lossen.

Voorbeeld één: als we iets koken aan de hand van een recept, volgen we een reeks stappen om een maaltijd te bereiden.

Voorbeeld twee: als we met een computer werken, kunnen we een opeenvolging van handelingen programmeren die de computer opdraagt wat hij moet doen.

### Evalueren of fouten oplossen

Dit is de vaardigheid om na te gaan of een proefmodel al dan niet werkt zoals je had bedoeld en zo niet, de vaardigheid om te bepalen wat moet worden verbeterd. Dit is ook de procedure die een computerprogrammeur volgt om fouten in een programma te zoeken en deze op te lossen.

Voorbeeld één: tijdens het koken proeven we regelmatig van het gerecht om na te gaan of het genoeg gekruid is.

Voorbeeld twee: als we een tekst controleren op spelfouten en ontbrekende leestekens, lossen we de fouten op zodat de tekst weer klopt.

### Abstractie

Abstractie is de vaardigheid om een probleem of een oplossing uit te leggen en de onbelangrijke details achterwege te laten. Met andere woorden: in staat zijn een idee te conceptualiseren.

Voorbeeld: Als we een fiets beschrijven, doen we dit aan de hand van enkele details. We vermelden bijvoorbeeld het type en de kleur, en eventueel meer details voor iemand die echt geïnteresseerd is in fietsen.



## Een proces voor de ontwikkeling van computational thinking-vaardigheden

### Een technisch ontwerproces gebruiken

Als technici oplossingen voor een probleem zoeken, maken ze gebruik van een ontwerproces. Ze doorlopen verschillende fasen die hen naar een oplossing leiden. Tijdens elk van deze fasen passen de leerlingen een aantal van hun vaardigheden toe of ontwikkelen ze deze. Deze vaardigheden noemen we “computational thinking-vaardigheden”.

In WeDo 2.0 volgen leerlingen een soortgelijke procedure:

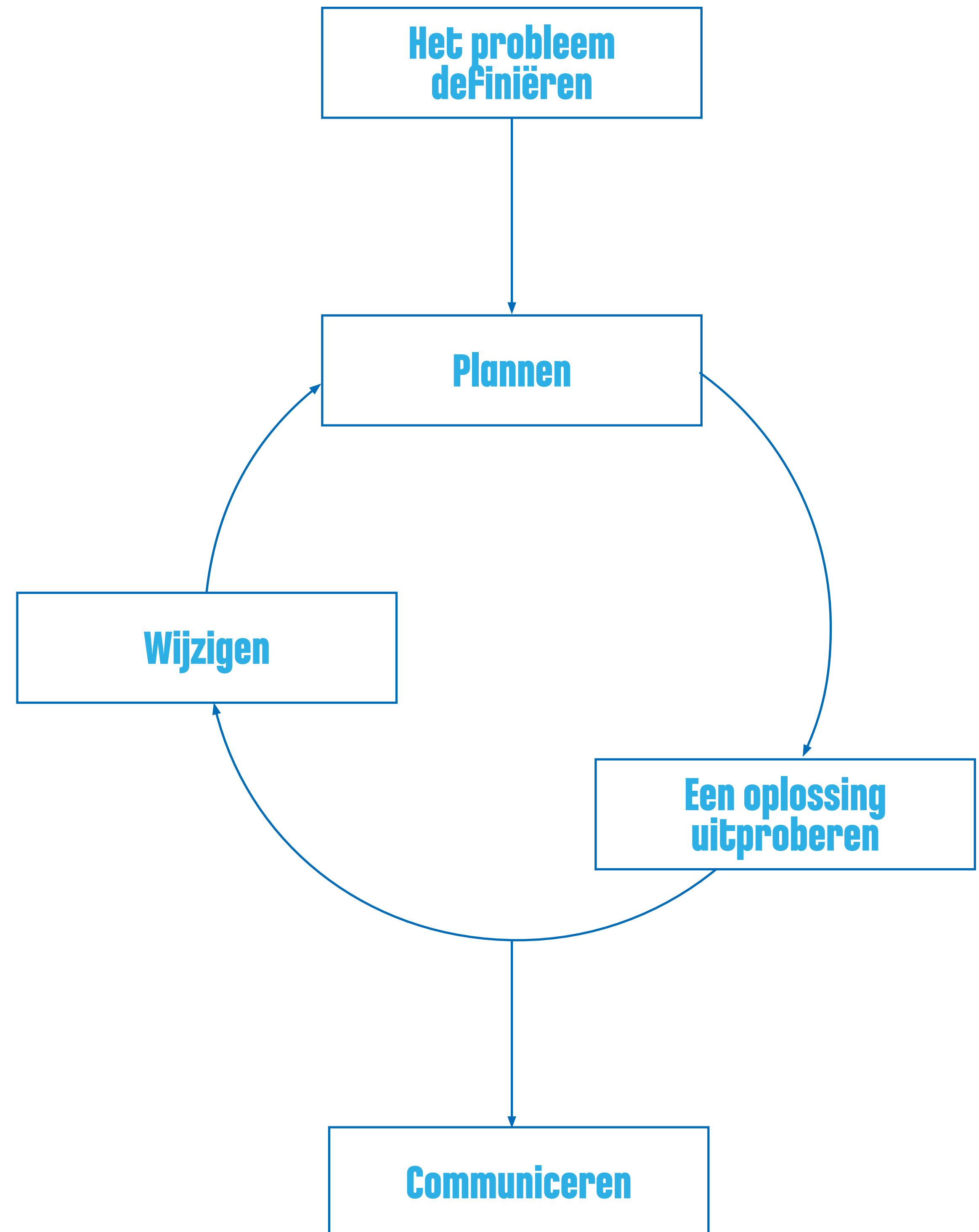
### Het probleem definiëren

De leerlingen krijgen een onderwerp aangeboden met een probleem dat ze willen oplossen of een situatie die ze willen verbeteren. Soms bevat een probleem veel details. Het probleem kan dan in kleinere onderdelen worden opgedeeld om gemakkelijker een oplossing te vinden.

Door het probleem op een eenvoudige manier te definiëren en door een aantal criteria voor succes te bepalen, ontwikkelen de leerlingen een vaardigheid die “Ontleden” wordt genoemd.

Met andere woorden:

- Is de leerling in staat om het probleem zelf uit te leggen?
- Is de leerling in staat om te beschrijven op welke manier ze gaan evalueren, ongeacht of ze erin zijn geslaagd het probleem op te lossen?
- Is de leerling in staat om het probleem in kleinere en beter beheersbare onderdelen op te delen?







## Een proces voor de ontwikkeling van computational thinking-vaardigheden

### Plannen

De leerlingen moeten de tijd nemen om verschillende oplossingen voor het probleem te bedenken en vervolgens een gedetailleerd plan maken om één van hun ideeën uit te voeren. Ze bepalen zelf de stappen die ze moeten doorlopen om de oplossing te vinden. Door te bepalen welke onderdelen van de taak ze misschien al eerder hebben gezien, ontwikkelen ze een vaardigheid die “Veralgemeniseren” wordt genoemd.

Met andere woorden:

- Is de leerling in staat om een lijst met acties op te stellen voor het programmeren?
- Is de leerling in staat om onderdelen van programma's te herkennen die hij of zij zou kunnen gebruiken?
- Is de leerling in staat om onderdelen van programma's opnieuw te gebruiken?

### Uitproberen

Elke leerling krijgt daarna de opdracht de eindversie van zijn of haar oplossing te creëren. In deze fase van het proces gebruiken leerlingen een programmeertaal met pictogrammen om hun LEGO® modellen te activeren. Wanneer de leerlingen hun ideeën coderen, ontwikkelen ze hun vaardigheden voor algoritmisch denken.

Met andere woorden:

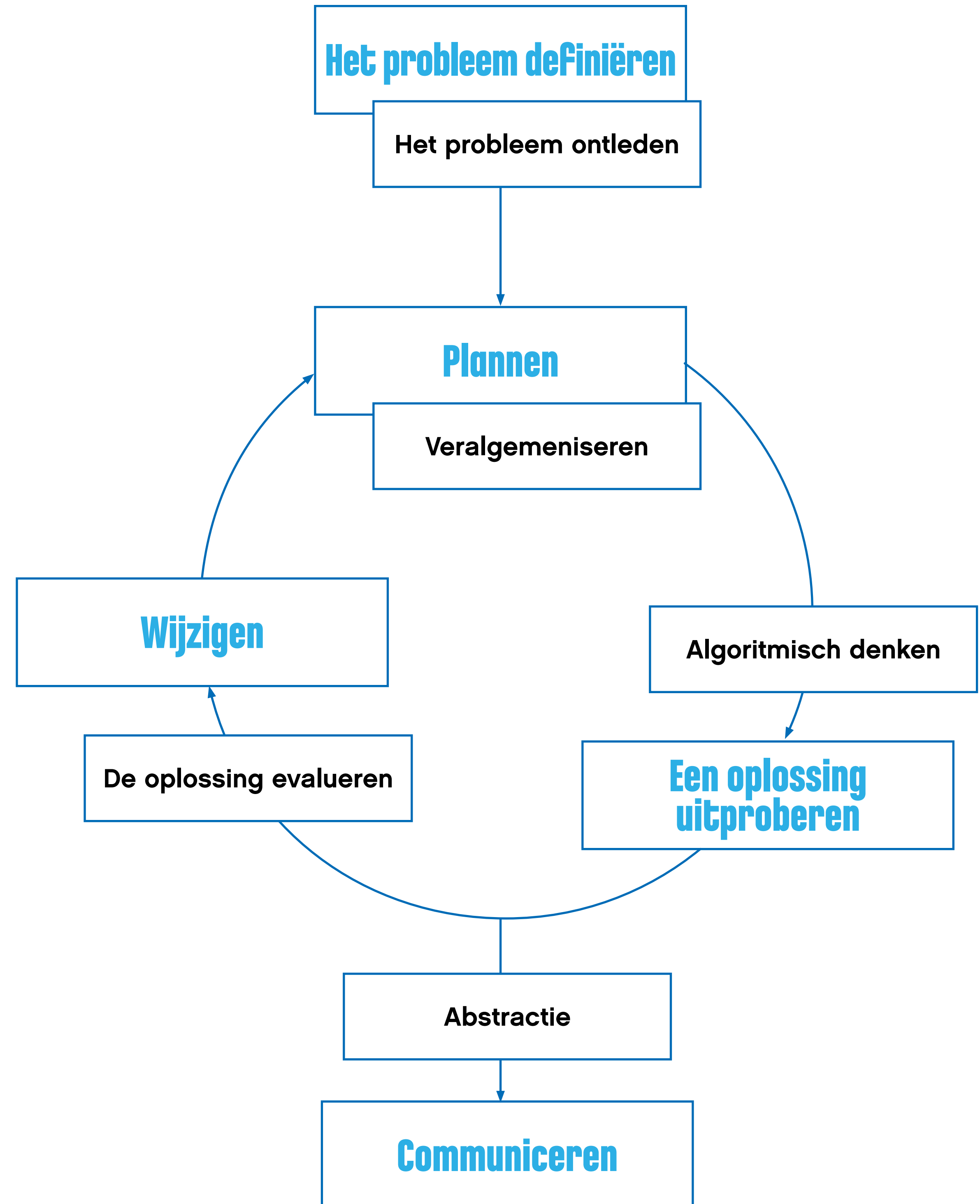
- Is de leerling in staat een oplossing voor een programma te programmeren?
- Is de leerling in staat om opeenvolgingen, herhalingen, voorwaarden, enz. te gebruiken?

### Wijzigen

De leerlingen evalueren hun oplossing en gaan na of hun programma en model al dan niet aan de criteria voor succes voldoen. Door hun evaluatievaardigheden te gebruiken, bepalen ze of ze een deel van hun programma moeten veranderen, herstellen of verbeteren of dat ze er fouten uit moeten halen.

Met andere woorden:

- Gebruikt de leerling delen van het programma opnieuw?
- Lost de leerling problemen in het programma op?
- Is de leerling in staat om te bepalen of de oplossing verband houdt met het probleem?





## Een proces voor de ontwikkeling van computational thinking-vaardigheden

### Communiceren

De leerlingen presenteren de eindversie van hun oplossing voor de klas en leggen hierbij uit waarom hun oplossing aan de criteria voor succes voldoet. Door hun oplossing op het juiste detailniveau uit te leggen, ontwikkelen ze hun abstractie- en communicatievaardigheden.

Met andere woorden:

- Legt de leerling het belangrijkste deel van de oplossing uit?
- Geeft de leerling genoeg details om de begrijpelijkheid te vergroten?
- Zorgt de leerling ervoor dat deze uitlegt hoe de oplossing aan de succesvereisten voldoet?





## Vaardigheden voor computational thinking ontwikkelen via coderen

De leerlingen maken kennis met een aantal programmeerprincipes om hun algoritmisch denken te stimuleren. Tijdens het ontwerpen van hun oplossingen zetten ze een reeks acties en structuren op waarmee hun modellen tot leven komen.

De meest voorkomende WeDo 2.0-programmeerprincipes die de leerlingen gaan gebruiken zijn:

### 1. Output

Output is alles wat wordt bestuurd door het programma dat de leerlingen schrijven. Voorbeelden van output in WeDo 2.0 zijn geluiden, verlichting, beeld en het in- en uitschakelen van motoren.

### 2. Input

Input is de informatie die een computer of apparaat ontvangt. Deze informatie kan door het gebruik van sensoren in de vorm van een numerieke waarde of een tekst worden ingevoerd. Een sensor die bijvoorbeeld iets waarneemt of meet (zoals een afstand) zet die waarde om in een digitaal inputsignaal zodat het in een programma kan worden gebruikt.

### 3. (Wachten op) gebeurtenissen

De leerlingen kunnen hun programma laten weten dat er moet worden gewacht tot er iets gebeurt voordat het de opeenvolging van acties hervat. Programma's kunnen gedurende een vooraf ingestelde tijd wachten of wachten tot een sensor iets detecteert.

### 4. Herhaling

De leerlingen kunnen programmeren dat de acties ofwel eindeloos, ofwel voor een bepaalde duur worden herhaald.

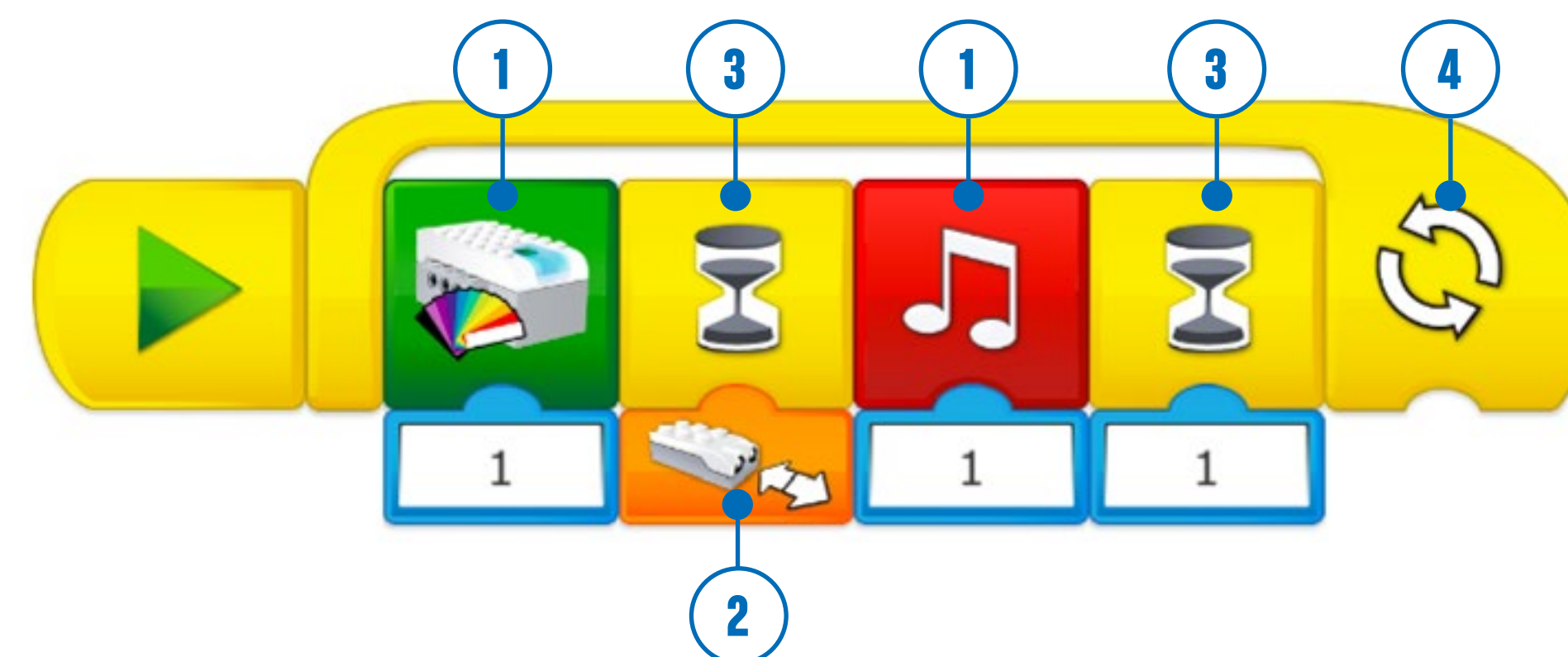
### 5. Functies

Functies zijn een groep acties die in specifieke situaties samen moeten worden gebruikt.

Bijvoorbeeld: de groep stenen die kan worden gebruikt om een licht te laten knipperen, zou de knipperfunctie kunnen worden genoemd.

### 6. Voorwaarden

Voorwaarden worden door de leerlingen gebruikt voor het programmeren van acties die enkel onder bepaalde omstandigheden moeten worden uitgevoerd. Het creëren van voorwaarden in een programma betekent dat een deel van het programma nooit wordt uitgevoerd als niet aan de voorwaarde wordt voldaan. Bijvoorbeeld: als de kantelsensor naar links kantelt, start de motor en als de sensor naar rechts kantelt, stopt de motor. Als de kantelsensor nooit naar links kantelt, dan start de motor nooit en als de sensor nooit naar rechts kantelt, dan stopt de motor nooit.



# WeDo 2.0 in het leerplan

In de LEGO® Education WeDo 2.0-projecten worden LEGO® stenen gecombineerd met 21e eeuwse vaardigheden. Alle WeDo 2.0-projecten zijn ontworpen om de computational thinking-vaardigheden van de leerlingen te ontwikkelen.





# Computational thinking in het leerplan

De wereld verandert en of we ons daar nu van bewust zijn of niet, technologie en informatica geven vorm aan bijna elk aspect van ons leven. Leerlingen worden in een snel tempo actieve burgers en het is één van de topprioriteiten van het land geworden om hen uit te rusten met de juiste vaardigheden.

Computational thinking is een set vaardigheden die zich wereldwijd verspreidt en wordt een belangrijke competentie om te ontwikkelen met betrekking tot technologie. Computational thinking is erkend als essentieel onderdeel van de 21e eeuwse vaardigheden, en wordt momenteel verwerkt in leerplannen in binnen- én buitenland.

Computational thinking is de basis geworden voor de normen vastgesteld door de Computer Science Teacher Association (CSTA) en andere organisaties zoals ISTE, Code.org en Computing at School (de Britse organisatie die verantwoordelijk is voor een wereldwijd erkend leerplan voor informatica). Al deze instellingen hebben hun leerplannen op elkaar afgestemd, waarbij de nadruk ligt op de ontwikkeling van computational thinking-vaardigheden. Meer informatie hierover vind je in het leerplankader Computational thinking van SLO.

Deze belangrijke vaardigheden kunnen worden ontwikkeld via boeiende activiteiten of projecten die zijn gebaseerd op problematische situaties uit het dagelijkse leven. Om deze ontwikkeling te ondersteunen, voegt LEGO® Education een gespecialiseerde reeks projecten voor computational thinking toe aan de wetenschapsprojecten die al beschikbaar zijn in WeDo 2.0.



## Visueel overzicht van de geleide projecten

### 1. Maanbasis

In dit project ontwerpen je leerlingen een oplossing waarbij een robot een basis op de maan kan bouwen.

### 2. Objecten vastpakken

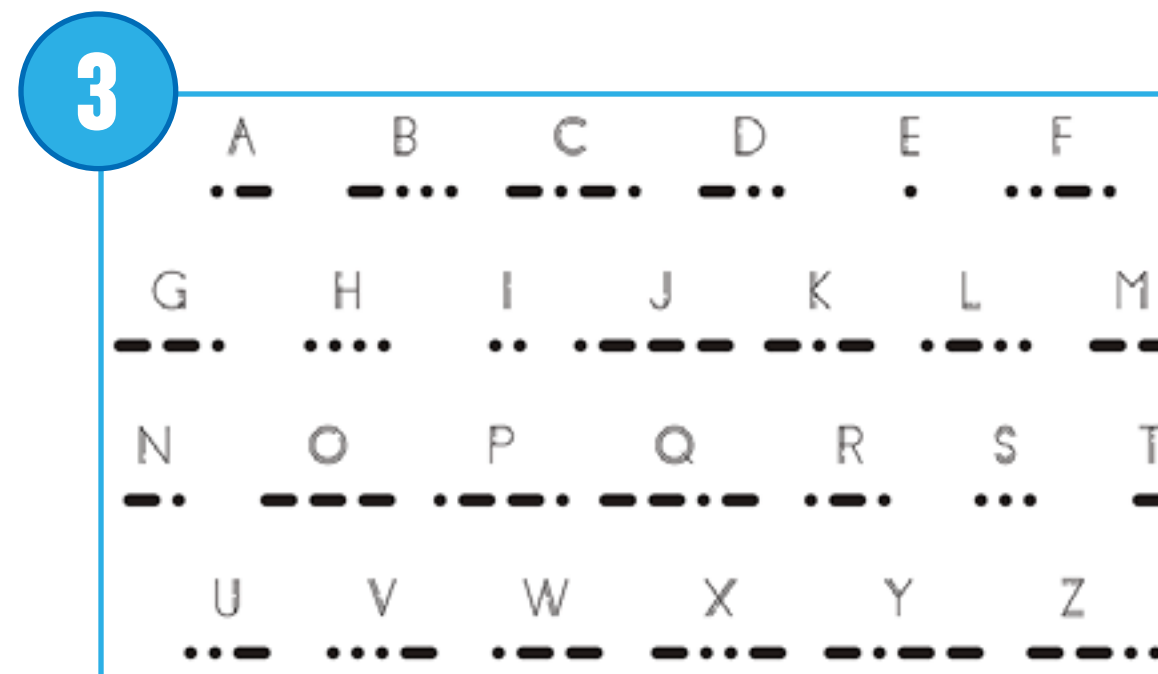
In dit project ontwerpen je leerlingen een oplossing voor een armprothese waarmee kleine voorwerpen kunnen worden bewogen.

### 3. Berichten verzenden

In dit project ontwerpen je leerlingen een oplossing om informatie uit te wisselen met behulp van een systeem van signalen die in patronen zijn geordend.

### 4. Vulkaanalarm

In dit project ontwerpen je leerlingen een apparaat waarmee vulkanische activiteit beter kan worden gecontroleerd, om op die manier de wetenschappelijke exploratie te bevorderen.





## Visueel overzicht van de open projecten

### 5. Inspectie

In dit project ontwerpen je leerlingen een oplossing waarbij een robot smalle ruimtes gaat onderzoeken. De beweging van de robot wordt hierbij geleid door sensoren.

### 6. Emotioneel ontwerp

In dit project ontwerpen je leerlingen een oplossing waarbij een robot positieve emoties kan tonen tijdens zijn interactie met mensen.

### 7. Veiligheid in de stad

In dit project ontwerpen je leerlingen een oplossing om de veiligheid in de stad te verbeteren.

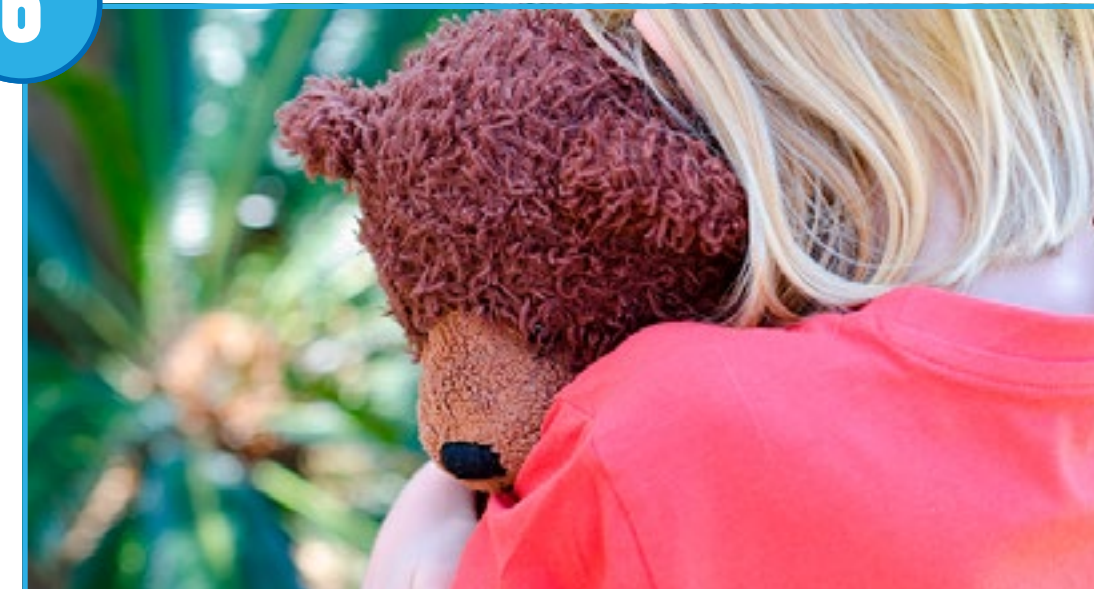
### 8. Dierenzintuigen

In dit project geven je leerlingen vorm aan de manier waarop dieren hun zintuigen gebruiken om met hun omgeving om te gaan.

5



6



7



8





## Mogelijk verloop bij het ontwikkelen van computational thinking-vaardigheden

Je kunt de projecten in de gewenste volgorde uitvoeren. In elk project worden de mogelijkheden om computational thinking-vaardigheden te ontwikkelen benadrukt. Jij bepaalt zelf welke projecten voor jou en je leerlingen het meest relevant zijn. De onderstaande volgorde wordt aanbevolen, omdat hierbij de programmeerconcepten langzaam steeds complexer worden:

### Aan de slag

Plan twee lessen van elk 45 minuten om je leerlingen kennis te laten maken met WeDo 2.0.

Les 1: Milo, de wetenschapsrover

Les 2: Milo's bewegingssensor, Milo's kantelsensor en samenwerken combineren

### Geleide projecten

Plan twee lessen van elk 45 minuten, zodat je leerlingen de tijd krijgen om een reeks acties te programmeren.

Les 3: Maanbasis (onderzoeks- en creatiefase)

Les 4: Maanbasis (test- en deelfase)

Plan twee lessen van elk 45 minuten waarin je leerlingen sensoren (input) gebruiken.

Les 5: Objecten vastpakken (onderzoeks- en creatiefase)

Les 6: Objecten vastpakken (test- en deelfase)

Plan twee lessen van elk 45 minuten, waarin je je leerlingen leren werken met sensoren (input), herhalingen en parallel programmeren.

Les 7: Berichten verzenden (onderzoeks- en creatiefase)

Les 8: Berichten verzenden (test- en deelfase)

Plan twee lessen van elk 45 minuten om je leerlingen te laten kennismaken met voorwaarden en de manier waarop ze alle andere programmeerprincipes kunnen integreren.

Les 9: Vulkaanalarm (onderzoeks- en creatiefase)

Les 10: Vulkaanalarm (test- en deelfase)

### Open projecten

Plan twee of drie lessen van elk 45 minuten om je eigen project te creëren, gebaseerd op één van de voorgestelde open projecten. Dit project zou alle programmeerprincipes moeten omvatten, evenals de computational thinking-vaardigheden die tijdens de geleide projecten zijn ontwikkeld.





## Mogelijk verloop bij het ontwikkelen van computational thinking-vaardigheden

### Aan de slag

Laat je leerlingen kennismaken met WeDo 2.0



45 minuten

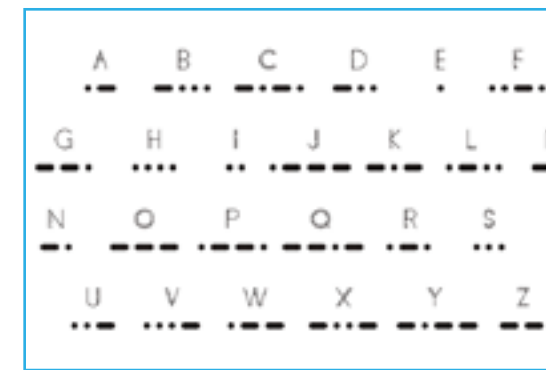


45 minuten



### Geleid project - Berichten verzenden

De leerlingen gebruiken sensoren (input), herhalingen en parallel programmeren.



Verkorte lesverloop gebruiken  
2 x 45 minuten



### Geleid project - Maanbasis

De leerlingen programmeren reeksen van acties.

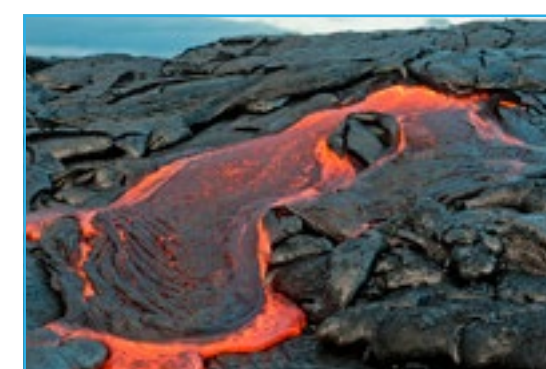


Verkorte lesverloop gebruiken  
2 x 45 minuten



### Geleid project - Vulkaanalarm

De leerlingen maken kennis met voorwaarden en andere programmeerprincipes.



Verkorte lesverloop gebruiken  
2 x 45 minuten



### Geleid project - Objecten vastpakken

De leerlingen gebruiken sensoren (input).



Verkorte lesverloop gebruiken  
2 x 45 minuten



### Open projecten





## Leerplanoverzicht van de geleide projecten

	1 Maanbasis	2 Objecten vastpakken	3 Berichten verzenden	4 Vulkaanalarm
<b>Natuur en techniek</b>	K42, 45 C1, 2	K34, 41, 42, 44, 45 C6, 7, 9, 10, 12	K44, 45 C7	K42, 44, 45 C2, 3, 11
<b>Aardrijkskunde</b>	K49, 50 C1, 2, 3			K49, 50 C2, 3
<b>Geschiedenis</b>	K53		K51, 52, 53	
<b>Mens en samenleving</b>		K34, 41, 44, 45 C 7, 9, 10	K35, 37 C7, 9	K39, 49 C2, 3, 9

Dit overzicht geeft inzicht in de koppeling van de WeDo 2.0-projecten met schoolvakken, kerndoelen (K) en concepten uit Oriëntatie op jezelf en de wereld. Concepten: Aarde (C1 Positie in het heelal, C2 Bouw van de aarde, C3 Processen van de aarde, C4 Weer en seizoenen, C5 Klimaat), Levende natuur (C6 Biologische eenheid, C7 Interactie, C8 Voortplanting, C9 Evenwicht), Niet-levende natuur (C10 Materiaal, C11 Energie, C12 Kracht en beweging)



## Leerplanoverzicht van de open projecten

	5 Inspectie	6 Emotioneel ontwerp	7 Veiligheid in de stad	8 Dierenzintuigen
<b>Natuur en techniek</b>	K45	K40, 41, 45 C6, 7, 10	K44, 45 C7, 10	K40, 41, 45 C6, 7
<b>Aardrijkskunde</b>				
<b>Geschiedenis</b>				
<b>Mens en samenleving</b>		K40 C7, 10	K40, 44 C7, 10	

Dit overzicht geeft inzicht in de koppeling van de WeDo 2.0-projecten met schoolvakken, kerndoelen (K) en concepten uit Oriëntatie op jezelf en de wereld. Concepten: Aarde (C1 Positie in het heelal, C2 Bouw van de aarde, C3 Processen van de aarde, C4 Weer en seizoenen, C5 Klimaat), Levende natuur (C6 Biologische eenheid, C7 Interactie, C8 Voortplanting, C9 Evenwicht), Niet-levende natuur (C10 Materiaal, C11 Energie, C12 Kracht en beweging)



## 21e eeuwse vaardigheden

Naast werken aan schoolvakken, kerndoelen en concepten uit Oriëntatie op jezelf en de wereld, werk je met WeDo 2.0 ook aan de ontwikkeling van de 21e-eeuwse vaardigheden. WeDo 2.0 is uitermate geschikt voor het aanleren van een groot aantal generieke vaardigheden, kennis en houdingen die nodig zijn om te kunnen functioneren in en om bij te dragen aan de 21e-eeuwse samenleving. Je vindt hier een omschrijving van de vaardigheden, die WeDo 2.0 helpt te ontwikkelen bij leerlingen. Een volledig overzicht van de 21e-eeuwse vaardigheden vind je op de website van SLO.

### 1. Communicatie

1. Kan doelgericht informatie uitwisselen met anderen (spreken, luisteren, de kern van de boodschap herkennen).
2. Kan omgaan met verschillende communicatiemiddelen en de verschillende strategieën daarbij.
3. Heeft inzicht in bepaalde mogelijkheden die ICT biedt om effectief te communiceren.

### 2. Creatief denken en Handelen

1. Heeft een onderzoekende en ondernemende houding.
2. Kan denken buiten de gebaande paden en kan nieuwe samenhangen zien.
3. Kan creatieve technieken gebruiken.
4. Durft risico's te nemen en kan fouten zien als leermogelijkheden.

### 3. Samenwerken

1. Kan verschillende rollen bij zichzelf en anderen (h)erkennen.
2. Kan hulp vragen, geven en ontvangen.
3. Heeft een positieve en open houding ten aanzien van andere ideeën.
4. Heeft respect voor culturele verschillen.
5. Kan onderhandelen en afspraken maken met anderen in een team.
6. Kan functioneren in heterogene groepen.
7. Kan effectief communiceren binnen de groep.

### 4. Kritisch denken

1. Kan effectief redeneren en formuleren.
2. Kan informatie interpreteren, analyseren en synthetiseren.



## Verbinding met referentiekaders Taal en Rekenen

De bijdrage aan twee vaardigheden lichten we hier verder toe: communicatie, met een verwijzing naar het referentiekader Taal (1F), een koppeling naar het referentiekader rekenen (1F) (en Computational Thinking, voor CT-deel). We geven alleen die onderdelen die bij het gebruik van WeDo 2.0 aan bod komen. Voor een volledige, uitgebreide beschrijving van de referentiekaders verwijzen we je naar de website van SLO en het Expertisecentrum Nederlands. Taal Het werken aan een goede communicatie in de groep tijdens het uitvoeren van een project bevordert het resultaat daarvan. En versterkt het delen van de uitkomst met anderen.

### 1. Mondelinge taalvaardigheid

1. Kan eenvoudige gesprekken voeren over vertrouwde onderwerpen in het dagelijks leven in en buiten school.
2. Kan luisteren naar eenvoudige teksten over alledaagse, concrete onderwerpen of over onderwerpen die aansluiten bij de leefwereld van de leerling.
3. Kan in eenvoudige bewoordingen een beschrijving geven, informatie geven, verslag uitbrengen, uitleg en instructie geven in alledaagse situaties in en buiten school.

### 2. Lezen

1. Kan eenvoudige teksten lezen over alledaagse onderwerpen en over onderwerpen die aansluiten bij de leefwereld.

### 3. Schrijven

1. Kan korte eenvoudige teksten schrijven over alledaagse onderwerpen of over onderwerpen uit de leefwereld.

### 4. Begrippen en taalverzorging

1. Is bekend met regels en begrippen die ten dienste staan van correct taalgebruik.
2. Is bekend met de regels en moeilijkheden in alfabetische spelling (dezelfde klank heeft dezelfde letter), orthografische spelling (autonome regels over de grens van lettergrepen heen), morfologische spelling (woorden die gevormd

worden door toevoeging van voorof achtervoegsels, werkwoordspelling waarvan een deel zuiver morfologisch is (tegenwoordige tijd meervoud, verleden tijd van werkwoorden met stam op -d, hele werkwoord), leestekens, afbreekregels en grammaticale begrippen voor werkwoordspelling.

### Rekenen

1. Weet eenvoudige getallen, bewerkingen en symbolen correct te noteren en te gebruiken.
2. Kan getallen lezen en uitleggen hoe getallen uit cijfers opgebouwd zijn;
3. Kan hoofdrekenen met en zonder notatie van tussenresultaten;
4. Kan hoofdbewerkingen (+, -, x, :) met gehele en eenvoudige decimale getallen op papier uitvoeren, evenals bewerkingen met eenvoudige breuken;
5. Kan berekeningen uitvoeren om problemen op te lossen en de rekenmachine op verstandige wijze inzetten;
6. Kan in de context van verhoudingen eenvoudige berekeningen uitvoeren, ook met procenten en verhoudingen;
7. Kan veel voorkomende en eenvoudige meetinstrumenten gebruiken en aflezen, met maateenheden rekenen en in eenvoudige gevallen maateenheden in elkaar omzetten;
8. Heeft een gevoel ontwikkeld voor standaardmaten in veel voorkomende situaties;
9. Kent namen van enkele meetkundige figuren en begrippen en kan deze gebruiken om situaties in de ruimte te beschrijven;
10. Kan eenvoudige tabellen, diagrammen en grafieken gebruiken bij het oplossen van problemen, ook om eenvoudige berekeningen uit te voeren.



## Onderzoekend en ontwerpend leren

De wetenschapprojecten in WeDo 2.0 zijn gecategoriseerd in drie groepen: onderzoeken, modelleren en ontwerpen. Daarmee zijn ze uitermate geschikt om invulling te geven aan onderzoekend en ontwerpend leren. Een onderzoekende en ontwerpende manier van leren stimuleren houding, vaardigheden, denkwijzen en kennis in samenhang. Een uitgebreidere beschrijving van onderzoekend en ontwerpend leren vind je op de website van SLO.

Onderzoekend en ontwerpend: onderzoekend en ontwerpend leren kennen zeven duidelijk onderscheiden stappen. De stappen in de WeDo 2.0-projecten volgen datzelfde proces en vragen de zelfde basisvaardigheden.

### Onderzoekend leren stappen

- Confronteren met object, verschijnsel of situatie
- Verkennen en vragen formuleren
- Onderzoek opzetten en bronnen, materiaal en instrumenten verzamelen
- Onderzoek uitvoeren en resultaten verwerken
- Conclusie(s) trekken en verslagleggen
- Onderzoek presenteren
- Verdiepen en verbreden

### Ontwerpend leren stappen

- Confronteren met vraag of probleem
- Verkennen en oplossingen bedenken
- Ontwerp schetsen en materiaal en gereedschap verzamelen
- Ontwerp realiseren
- Testen en bijstellen
- Product presenteren
- Verdiepen en verbreden



## Leerplanoverzicht van de geleide projecten, gekoppeld aan W&T-vaardigheden

	1 Maanbasis	2 Objecten vastpakken	3 Berichten verzenden	4 Vulkaanalarm
<b>Vaardigheid één:</b> Vragen stellen en problemen definiëren	●	●	●	●
<b>Vaardigheid twee:</b> Modellen ontwikkelen en gebruiken				
<b>Vaardigheid drie:</b> Een onderzoek plannen en uitvoeren				
<b>Vaardigheid vier:</b> Gegevens analyseren en interpreteren				
<b>Vaardigheid vijf:</b> Gebruikmaken van wiskunde en computational thinking	●	●	●	●
<b>Vaardigheid zes:</b> Verklaringen opstellen en oplossingen uitwerken	●	●	●	●
<b>Vaardigheid zeven:</b> Argumenteren op basis van bewijzen	●	●	●	●
<b>Vaardigheid acht:</b> Informatie verkrijgen, beoordelen en doorgeven	●	●	●	●



# Leerplanoverzicht van de open projecten, gekoppeld aan W&T-vaardigheden

	5 Inspectie	6 Emotioneel ontwerp	7 Veiligheid in de stad	8 Dierenzintuigen
<b>Vaardigheid één:</b> Vragen stellen en problemen definiëren	●	●	●	●
<b>Vaardigheid twee:</b> Modellen ontwikkelen en gebruiken				●
<b>Vaardigheid drie:</b> Een onderzoek plannen en uitvoeren				
<b>Vaardigheid vier:</b> Gegevens analyseren en interpreteren				
<b>Vaardigheid vijf:</b> Gebruikmaken van wiskunde en computational thinking	●	●	●	●
<b>Vaardigheid zes:</b> Verklaringen opstellen en oplossingen uitwerken	●	●	●	
<b>Vaardigheid zeven:</b> Argumenteren op basis van bewijzen	●	●	●	●
<b>Vaardigheid acht:</b> Informatie verkrijgen, beoordelen en doorgeven	●	●	●	●





## Leerplanoverzicht van de open projecten, geordend volgens de CSTA-normen

Identificator			Tussentijdse norm CSTA K-12 CS							
			1 Maanbasis	2 Objecten vastpakken	3 Berichten verzenden	4 Vulkaanalarm	5 Inspectie	6 Emotioneel ontwerp	7 Veiligheid in de stad	8 Dierenzintuigen
K-2	1A-A-3-7	Algoritmes ontwikkelen en uitvoeren (sets van stap-voor-stap instructies). Deze algoritmes bevatten reeksen en eenvoudige herhalingen om een taak zowel zelfstandig als gezamenlijk en met of zonder computerapparaat te volbrengen.	●	●	●	●	●	●	●	●
K-2	1A-A-6-8	Een algoritme analyseren dat reeksen en eenvoudige herhalingen bevat, met of zonder computerapparaat, en de fouten oplossen (herstellen).	●	●	●	●	●	●	●	●
K-2	1A-C-7-9	Software die computerapparaten bestuurt herkennen en gebruiken (bv. een app gebruiken om op het scherm te tekenen of software gebruiken om een verhaal te schrijven of robots te besturen).	●	●	●	●	●	●	●	●
K-2	1A-C-7-10	De juiste terminologie gebruiken bij het benoemen en beschrijven van de functie van veelvoorkomende computerapparaten en -onderdelen (bv. desktopcomputer, laptop, tablet, beeldscherm, toetsenbord, muis, printer).								
K-2	1A-C-6-11	Eenvoudige hardware- of softwareproblemen herkennen die zich tijdens gebruik kunnen voordoen (bv. een app of programma werkt niet zoals verwacht, geen geluid, apparaat kan niet worden ingeschakeld) en hiervoor de juiste terminologie gebruiken.	●	●	●	●	●	●	●	●
K-2	1A-D-7-12	Gedurende een bepaalde periode gegevens verzamelen en deze ordenen in een tabel of grafiek om een voorspelling te kunnen doen.								
K-2	1A-D-4-13	Een computerapparaat gebruiken om informatie op te slaan, te zoeken, op te vragen, te wijzigen en te verwijderen en om de informatie die als gegevens wordt opgeslagen, te definiëren.								
K-2	1A-D-4-14	Een model van een object of proces creëren om patronen en essentiële elementen te herkennen (bv. waterkringloop, levenscyclus van een vlinder, weerpatronen van de seizoenen).	●	●	●	●	●	●	●	●



# Leerplanoverzicht van de open projecten, geordend volgens de CSTA-normen

Identificator		Tussentijdse norm CSTA K–12 CS	1 Maanbasis	2 Objecten vastpakken	3 Berichten verzenden	4 Vulkaanalarm	5 Inspectie	6 Emotioneel ontwerp	7 Veiligheid in de stad	8 Dierenzintuigen
3–5	1B-A-2-1	Strategieën voor samenwerking toepassen om in de ontwerpcyclus probleemoplossing te ondersteunen.	●	●	●	●	●	●	●	●
3–5	1B-A-7-2	Op een correcte manier citaten en documenten gebruiken wanneer ideeën worden overgenomen en aangepast voor eigen gebruik (bv. foto's gebruiken die door anderen zijn genomen, muziek gebruiken die door anderen is gemaakt, programmeringsprojecten bewerken).	●	●	●	●	●	●	●	●
3–5	1B-A-5-3	Een plan opstellen als onderdeel van het iteratieve ontwerpproces (bv. storyboard, stroomschema, pseudocode, story map), zowel zelfstandig als met verschillende teams.	●	●	●	●	●	●	●	●
3–5	1B-A-5-4	Programma's bouwen om een probleem op te lossen of je creativiteit uit te drukken. Deze programma's bevatten reeksen, gebeurtenissen, herhalingen, voorwaarden, parallelisme en variabelen en kunnen zowel zelfstandig als in een groep (bv. pair programming) worden gebouwd met een visuele programmeertaal met blokken of een op tekst gebaseerde taal.	●	●	●	●	●	●	●	●
3–5	1B-A-5-5	Wiskundige bewerkingen gebruiken om de waarde van een variabele te veranderen.				●				
3–5	1B-A-3-6	Een groter probleem in kleinere subproblemen opdelen (ontleden), zelfstandig of in een groep.	●	●	●	●	●	●	●	●



# Leerplanoverzicht van de open projecten, geordend volgens de CSTA-normen

Identificator		Tussentijdse norm CSTA K-12 CS	1 Maanbasis	2 Objecten vastpakken	3 Berichten verzenden	4 Vulkaanalarm	5 Inspectie	6 Emotioneel ontwerp	7 Veiligheid in de stad	8 Dierenzintuigen
3-5	1B-A-3-7	Algoritmes ontwikkelen en uitvoeren (sets van stap-voor-stap instructies). Deze algoritmes bevatten reeksen, herhalingen en voorwaarden om een taak zowel zelfstandig als gezamenlijk en met of zonder computerapparaat te volbrengen.	●	●	●	●	●	●	●	●
3-5	1B-A-6-8	Een algoritme analyseren dat reeksen, gebeurtenissen, herhalingen, voorwaarden en parallelisme bevat en de fouten oplossen (herstellen).	●	●	●	●	●	●	●	●
3-5	1B-C-7-9	Een model maken van de manier waarop een computersysteem werkt. (Toelichting: alleen met de basiselementen van een computersysteem, zoals input, output, processor, sensoren en geheugen.)								
3-5	1B-C-7-10	De juiste terminologie gebruiken bij het benoemen van interne en externe onderdelen van computerapparaten en hun verbanden, mogelijkheden en beperkingen beschrijven.								
3-5	1B-C-6-11	Eenvoudige hardware- of softwareproblemen die zich tijdens gebruik kunnen voordoen herkennen, de juiste terminologie ervoor gebruiken en probleemoplossende strategieën toepassen (bv. het apparaat herstarten, controleren of er stroom en een netwerk beschikbaar zijn, de app sluiten en opnieuw openen).								
3-5	1B-D-5-12	Een computational artefact bouwen dat vorm geeft aan de eigenschappen en gedragingen die met een begrip worden geassocieerd (bv. het zonnestelsel, de levenscyclus van een plant).	●	●	●	●	●	●	●	●
3-5	1B-D-5-13	Een vraag beantwoorden met behulp van een computer om gegevens die door de klas of een leerling zijn verzameld te bewerken (bv. sorteren, optellen en/of het gemiddelde berekenen, een tabel of grafiek maken) en te analyseren.								



## Leerplanoverzicht van de open projecten, geordend volgens de CSTA-normen

Identificator		Tussentijdse CSTA-norm	1 Maanbasis	2 Objecten vastpakken	3 Berichten verzenden	4 Vulkaanalarm	5 Inspectie	6 Emotioneel ontwerp	7 Veiligheid in de stad	8 Dierenzintuigen
3-5	1B-D-4-14	Numerieke waarden gebruiken om niet-numerieke ideeën op de computer weer te geven (binair stelsel, ASCII, eigenschappen van pixels zoals RGB).			●					
3-5	1B-I-7-15	Positieve en negatieve gevolgen van de alomtegenwoordigheid van computers en informatica in het dagelijks leven evalueren en beschrijven (bv. filmpjes en audiobestanden downloaden, elektronische apparaten, draadloos internet, mobiele computerapparaten, gps-systemen, draagbare computerapparaten).								
3-5	1B-I-7-16	Voorbeelden geven van de manier waarop informatica de maatschappij kan beïnvloeden en van de manier waarop maatschappelijke waarden de keuzes op het gebied van informatica kunnen vormen.								
3-5	1B-I-1-17	Verschillende invalshoeken, synchroon of asynchroon, zoeken om een project te verbeteren en deze invalshoeken met elkaar vergelijken.								
3-5	1B-I-1-18	Brainstormen over manieren waarop computerapparaten voor alle gebruikers toegankelijker kunnen worden gemaakt.								
3-5	1B-I-1-19	Problemen uitleggen die te maken hebben met het gebruik van computerapparaten en netwerken (bv. uitloggen om te vermijden dat anderen jouw account gebruiken, cyberpesten, privacy van persoonlijke informatie en eigenaarschap).								
3-5	1B-N-7-20	Voorbeelden van sterke wachtwoorden opstellen, uitleggen waarom het belangrijk is sterke wachtwoorden te gebruiken en een correct gebruik en correcte beveiliging van persoonlijke wachtwoorden laten zien.								
3-5	1B-N-4-21	Een model maken van de manier waarop een apparaat op een netwerk een bericht verzendt van het ene apparaat (zender) naar het andere (ontvanger) en hiervoor specifieke regels volgt.			●					

# Competenties voor computational thinking evalueren

Je kunt de voortgang van je leerlingen in een WeDo 2.0-project op verschillende manieren volgen en evalueren.

Dit deel biedt de volgende hulpmiddelen voor evaluatie:

- Documentatiepagina's
- Zelfevaluatie
- Notitieblad voor anekdotes
- Schema met observatierubrieken





## Evaluatie door de leerling

### Documentatiepagina's

Bij elk project moeten de leerlingen documenten opstellen om hun werk samen te vatten.

Om een volledig wetenschappelijk verslag te verkrijgen, is het noodzakelijk dat de leerlingen:

- Hun werk vastleggen met behulp van verschillende soorten media
- Elke stap van het proces vastleggen
- De tijd nemen om hun document te ordenen en samen te stellen

Het is zeer waarschijnlijk dat het eerste document dat je leerlingen maken niet zo goed is als de documenten die daarop volgen. Je kunt hen helpen door:

- Hen feedback te geven en hen de tijd te geven om te ontdekken waar en hoe ze bepaalde delen van hun document kunnen verbeteren.
- Hen de mogelijkheid te geven om hun documenten met elkaar te delen. Door hun wetenschappelijke bevindingen over te brengen, leren de leerlingen hoe wetenschappers te werk gaan.

### Zelfevaluatie

Na elk project moeten de leerlingen nadenken over het werk dat ze hebben verricht. Gebruik de volgende pagina om je leerlingen aan te zetten tot nadenken en om ze doelen te laten stellen voor het volgende project.





# Rubriek voor zelfevaluatie

Naam: \_\_\_\_\_

Klas: \_\_\_\_\_

Project: \_\_\_\_\_

Instructies: omcirkel de steen die weergeeft hoe goed je het hebt gedaan. Hoe groter de steen, hoe beter je het hebt gedaan.				
Ik heb de vraagstelling of het probleem gedefinieerd.				
Ik heb een LEGO® model gebouwd en een oplossing geprogrammeerd.				
Ik heb mijn oplossing uitgeprobeerd en verbeteringen aangebracht.				
Ik heb mijn ideeën vastgelegd en gedeeld.				

### Reflectie over het project

Iets wat ik heel goed deed was: \_\_\_\_\_

Iets wat ik wil verbeteren voor de volgende keer is: \_\_\_\_\_



## Evaluatie door de leerkracht

Om ervoor te zorgen dat de leerlingen hun vaardigheden wat betreft wetenschap, techniek en computational thinking kunnen ontwikkelen, is het noodzakelijk dat zij daarvoor voldoende tijd krijgen en op feedback kunnen rekenen. Net zoals in de ontwerpcyclus, waarin leerlingen moeten leren dat mislukkingen deel uitmaken van het proces, is het belangrijk dat met een evaluatie de leerlingen feedback krijgen met betrekking tot wat ze goed deden en wat ze kunnen verbeteren. Probleemgestuurd onderwijs gaat niet om slagen of mislukken. Het gaat om actief leren en voortdurend op ideeën voortbouwen en deze uitproberen.

Je kunt je leerlingen op verschillende manieren feedback geven om ze te helpen hun vaardigheden te ontwikkelen. Bij elke fase van de WeDo 2.0-projecten bieden we voorbeelden van rubrieken die kunnen worden gebruikt om:

- Het gedrag, de reactie en de strategieën van de leerlingen te observeren
- Vragen te stellen over hun denkproces

Aangezien leerlingen vaak in groepen werken, is het goed om zowel op teamniveau als op individueel niveau feedback te geven.

### Notitieblad voor anekdotes

Het notitieblad voor anekdotes biedt je de mogelijkheid om voor elke leerling alle waarnemingen te noteren waarvan je denkt dat deze belangrijk zijn. Gebruik het sjabloon op de volgende pagina om de leerlingen waar nodig feedback te geven.






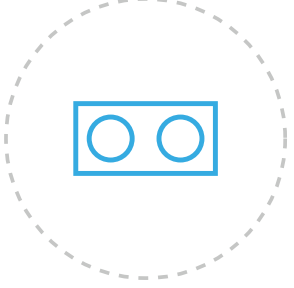
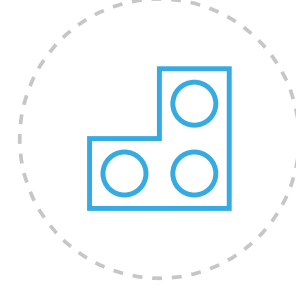
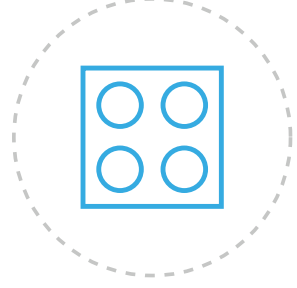


# Notitieblad voor anekdotes

Naam:

Klas:

Project:

1. Beginner	2. Gevorderd	3. Bedreven	4. Expert
			

Opmerkingen:



## Evaluatie door de leerkracht

### Observatierubrieken

Voor elk geleid project zijn er voorbeelden van rubrieken gegeven. Voor elke leerling of elke groep kun je de observatierubrieken gebruiken om:

- De prestaties van de leerlingen bij elke stap van het proces te evalueren
- De leerlingen constructieve feedback te geven om hen te helpen bij hun voortgang

De observatierubrieken in de geleide projecten kunnen aan je behoeften worden aangepast. De rubrieken zijn gebaseerd op deze progressieve fasen:

#### 1. Beginner

De leerling bevindt zich in de beginfase van ontwikkeling voor wat betreft inhoudelijke kennis, het vermogen om inhoud te begrijpen en toe te passen en/of om samenhangende gedachten over een bepaald onderwerp te uiten.

#### 2. Gevorderd

De leerling is alleen in staat om blijk te geven van basiskennis (bv. woordenschat) en kan de inhoudelijke kennis nog niet toepassen en de voorgestelde concepten nog niet begrijpen.

#### 3. Bedreven

De leerling heeft een bepaald begripsniveau van de inhoud en de concepten en kan de geleerde onderwerpen, inhoud of concepten voldoende demonstreren. Het vermogen om deze kennis buiten de vereiste opdracht te bespreken en toe te passen ontbreekt nog.

#### 4. Expert

De leerling kan de concepten en ideeën naar een hoger niveau tillen, de concepten op andere situaties toepassen en kennis combineren, toepassen en verruimen om gesprekken aan te gaan die uitbreidingen van ideeën bevatten.

### ► Suggestie

Gebruik het schema met observatierubrieken op de volgende pagina om de voortgang van je leerlingen bij te houden.





# Schema met observatierubrieken

Klas:		Project:			
Namen van de leerlingen		Onderzoeken	Creëren	Testen	Delen
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					



Te gebruiken in combinatie met de rubrieken die op de volgende pagina worden beschreven: (1) beginner, (2) gevorderd, (3) bedreven, (4) expert.



## De projectfasen evalueren - Algemene rubrieken

Je kunt de evaluatierubrieken gebruiken om aan het einde van elke fase van een project algemene feedback te geven op een schaal van 1 tot 4.

### Onderzoeksfase

In de onderzoeksfase dient de feedback betrekking te hebben op de al dan niet actieve betrokkenheid van de leerling in het gesprek door vragen te stellen en te beantwoorden, evenals op hun niveau van inzicht in het probleem.

1. De leerling is niet in staat om vragen te beantwoorden of voldoende deel te nemen aan gesprekken.
2. Na wat hulp is de leerling in staat om vragen te beantwoorden of voldoende deel te nemen aan gesprekken.
3. De leerling is in staat om vragen te beantwoorden of voldoende deel te nemen aan gesprekken.
4. De leerling is in staat om zijn project in een klassikaal gesprek toe te lichten.

### Testfase

Zorg er tijdens de testfase voor dat de leerling goed in teamverband werkt, zijn/haar beste oplossing verantwoordt en gebruikmaakt van de informatie die is verzameld tijdens de onderzoeksfase.

1. De leerling is niet in staat om goed in teamverband te werken, oplossingen te verantwoorden en verzamelde informatie voor verdere ontwikkeling te gebruiken.
2. De leerling is in staat om in teamverband te werken en om met begeleiding of hulp informatie te verzamelen en toe te passen om oplossingen te verantwoorden.
3. De leerling is in staat om in teamverband te werken en bij te dragen aan teambesprekingen, om oplossingen te verantwoorden en om inhoudelijke informatie te verzamelen en toe te passen.
4. De leerling kan oplossingen voor het verzamelen en toepassen van informatie verantwoorden en bespreken.

### Deelfase

Zorg er tijdens de deelfase voor dat de leerling in staat is zijn of haar oplossing met de juiste woordenschat en met het juiste detailniveau te beschrijven.

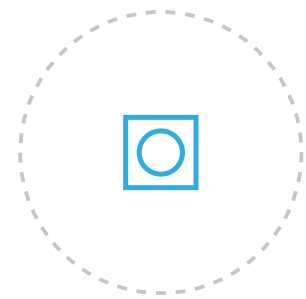

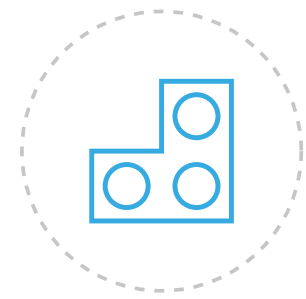
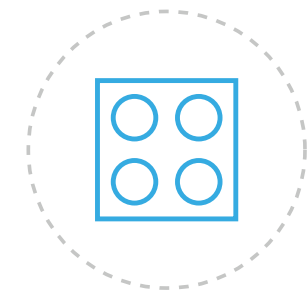
1. De leerling gebruikt geen bewijs uit zijn/haar bevindingen met betrekking tot ideeën die tijdens de presentatie zijn gedeeld en volgt de vastgestelde richtlijnen niet.
2. De leerling gebruikt enig bewijs uit zijn/haar bevindingen, maar de verantwoording is beperkt. De vastgestelde richtlijnen worden over het algemeen gevolgd, maar op bepaalde vlakken ook niet.
3. De leerling onderbouwt zijn/haar bevindingen met voldoende bewijzen en volgt de vastgestelde richtlijnen voor het presenteren van deze bewijzen.
4. De leerling bespreekt zijn/haar bevindingen volledig en maakt uitgebreid gebruik van passend bewijsmateriaal om zijn/haar redenering te verantwoorden en volgt hierbij alle vastgestelde richtlijnen.



# Vaardigheden voor computational thinking evalueren

Naam: \_\_\_\_\_

Klas: \_\_\_\_\_

Ontleden	1. Beginner	2. Gevorderd	3. Bedreven	4. Expert	Opmerkingen
					
Het probleem in eigen woorden beschrijven.	De leerling is niet in staat om het probleem in zijn eigen woorden te beschrijven.  <input type="checkbox"/>	De leerling is na wat hulp in staat om het probleem in zijn eigen woorden te beschrijven.  <input type="checkbox"/>	De leerling is in staat om het probleem in zijn eigen woorden te beschrijven.  <input type="checkbox"/>	De leerling is in staat om het probleem in zijn eigen woorden te beschrijven en begint het probleem in kleinere onderdelen op te delen.  <input type="checkbox"/>	
Beschrijven hoe je te weten komt of je al dan niet een succesvolle oplossing voor het probleem hebt gevonden.	De leerling is niet in staat om succescriteria te beschrijven.  <input type="checkbox"/>	De leerling is in staat om, na wat hulp, succescriteria te beschrijven.  <input type="checkbox"/>	De leerling is in staat om succescriteria te beschrijven.  <input type="checkbox"/>	De leerling is in staat om succescriteria gedetailleerd te beschrijven.  <input type="checkbox"/>	
Beschrijven hoe je het probleem in kleinere onderdelen kunt opdelen.	De leerling is niet in staat om het probleem op te delen.  <input type="checkbox"/>	Na wat hulp is de leerling in staat om het probleem in kleinere onderdelen op te delen.  <input type="checkbox"/>	De leerling is in staat om het probleem in kleinere onderdelen op te delen.  <input type="checkbox"/>	De leerling is in staat om het probleem in kleinere onderdelen op te delen en kan de verbanden tussen elk van de onderdelen beschrijven.  <input type="checkbox"/>	



# Vaardigheden voor computational thinking evalueren

Naam: \_\_\_\_\_

Klas: \_\_\_\_\_

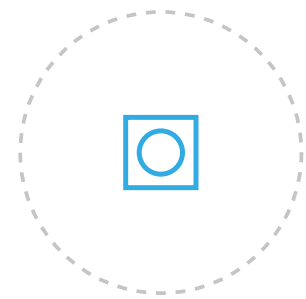


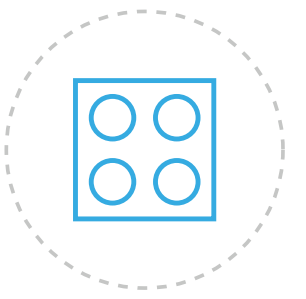
Veralgemeineren	1. Beginner	2. Gevorderd	3. Bedreven	4. Expert	Opmerkingen
Beschrijven welk programma uit de programmabibliotheek (of ergens anders) je hebt gebruikt en waarom.	De leerling is niet in staat om te beschrijven welk programma is gebruikt en waarom.  <input type="checkbox"/>	De leerling is in staat om aan te geven welk programma is gebruikt.  <input type="checkbox"/>	De leerling is in staat om te beschrijven welk programma is gebruikt en waarom.  <input type="checkbox"/>	De leerling is in staat om gedetailleerd te beschrijven welk programma is gebruikt en welke wijzigingen eraan zijn aangebracht.  <input type="checkbox"/>	
Observeren hoe je leerlingen patronen herkennen of concepten hergebruiken die ze al eerder hebben gezien.	De leerling is niet in staat om patronen te herkennen of concepten te hergebruiken die hij al eerder heeft gezien.  <input type="checkbox"/>	Na wat hulp is de leerling in staat om patronen te herkennen of concepten te hergebruiken die hij al eerder heeft gezien.  <input type="checkbox"/>	De leerling is in staat om patronen te herkennen of concepten te hergebruiken die hij al eerder heeft gezien opnieuw.  <input type="checkbox"/>	De leerling is in staat om patronen te herkennen of eigen concepten opnieuw te gebruiken.  <input type="checkbox"/>	



# Vaardigheden voor computational thinking evalueren

Naam: \_\_\_\_\_

Klas: \_\_\_\_\_

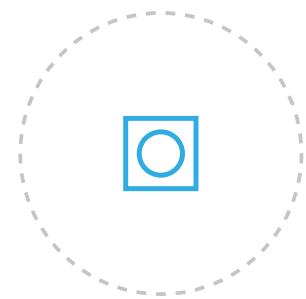
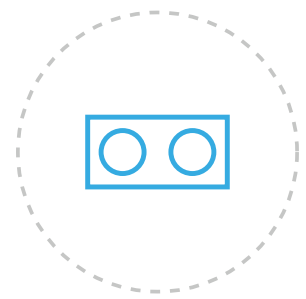

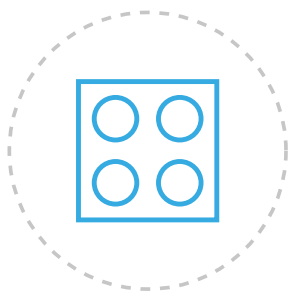
Algoritmisch denken	1. Beginner	2. Gevorderd	3. Bedreven	4. Expert	Opmerkingen
					
De lijst met acties voor het programmeren beschrijven.	De leerling is niet in staat om een lijst met acties op te stellen voor het programmeren.  <input type="checkbox"/>	Na wat hulp is de leerling in staat om een lijst met acties op te stellen voor het programmeren.  <input type="checkbox"/>	De leerling is in staat om een lijst met acties op te stellen.  <input type="checkbox"/>	De leerling is in staat om een gedetailleerde lijst met acties op te stellen om hem te helpen het programma te ontwerpen.  <input type="checkbox"/>	
Beschrijven hoe je jouw oplossing hebt geprogrammeerd.	De leerling is niet in staat om het programma te beschrijven.  <input type="checkbox"/>	Na wat hulp is de leerling in staat om het programma te beschrijven.  <input type="checkbox"/>	De leerling is in staat om het programma te beschrijven.  <input type="checkbox"/>	De leerling is in staat om het programma te beschrijven en geeft gedetailleerde informatie over elk onderdeel.  <input type="checkbox"/>	
De programmeerprincipes (bv. output, input, gebeurtenissen, herhalingen, enz.) beschrijven die in je oplossing zijn gebruikt.	De leerling is niet in staat de programmeerprincipes te beschrijven die in de oplossing zijn gebruikt.  <input type="checkbox"/>	Na wat hulp is de leerling in staat om de programmeerprincipes te beschrijven die in de oplossing zijn gebruikt.  <input type="checkbox"/>	De leerling is in staat de programmeerprincipes te beschrijven die in de oplossing zijn gebruikt.  <input type="checkbox"/>	De leerling is in staat om met groot begrip de programmeerprincipes te beschrijven die in de oplossing zijn gebruikt.  <input type="checkbox"/>	



## Vaardigheden voor computational thinking evalueren

Naam: \_\_\_\_\_

Klas: \_\_\_\_\_

Evaluatie	1. Beginner	2. Gevorderd	3. Bedreven	4. Expert	Opmerkingen
					
Beschrijven wat er gebeurde toen je je programma uitvoerde en of dit al dan niet was wat je had verwacht.	De leerling is niet in staat te beschrijven wat er gebeurde.  <input type="checkbox"/>	Na wat hulp is de leerling in staat om te beschrijven wat er gebeurde en om dit te vergelijken met wat er werd verwacht.  <input type="checkbox"/>	De leerling is in staat om te beschrijven wat er gebeurde en om dit te vergelijken met wat er werd verwacht.  <input type="checkbox"/>	De leerling is in staat om te beschrijven wat er gebeurde, om dit te vergelijken met wat er werd verwacht en om er oplossingen voor te vinden.  <input type="checkbox"/>	
Beschrijven hoe je de problemen in je programma hebt opgelost.	De leerling kan niet beschrijven hoe de problemen zijn opgelost.  <input type="checkbox"/>	Na wat hulp kan de leerling beschrijven hoe de problemen zijn opgelost.  <input type="checkbox"/>	De leerling kan beschrijven hoe de problemen zijn opgelost.  <input type="checkbox"/>	De leerling kan gedetailleerd beschrijven hoe de problemen zijn opgelost.  <input type="checkbox"/>	
Beschrijven hoe jouw oplossing verband houdt met het probleem.	De leerling is niet in staat te beschrijven hoe de oplossing verband houdt met het probleem.  <input type="checkbox"/>	Na wat hulp is de leerling in staat te beschrijven hoe de oplossing verband houdt met het probleem.  <input type="checkbox"/>	De leerling is in staat te beschrijven hoe de oplossing verband houdt met het probleem.  <input type="checkbox"/>	De leerling is in staat gedetailleerd te beschrijven hoe de oplossing verband houdt met het probleem.  <input type="checkbox"/>	
Beschrijven hoe je in de loop van het project nieuwe manieren hebt uitprobeerde om de problemen op te lossen	De leerling is niet in staat om andere manieren te beschrijven die hij in de loop van het project heeft uitprobeerde.  <input type="checkbox"/>	Na wat hulp is de leerling in staat om andere manieren te beschrijven die hij in de loop van het project heeft uitprobeerde.  <input type="checkbox"/>	De leerling is in staat om andere manieren te beschrijven die hij in de loop van het project heeft uitprobeerde.  <input type="checkbox"/>	De leerling is in staat om andere manieren te beschrijven die hij in de loop van het project heeft uitprobeerde en is in staat om te beschrijven waarom niet elke mogelijkheid is overwogen.  <input type="checkbox"/>	

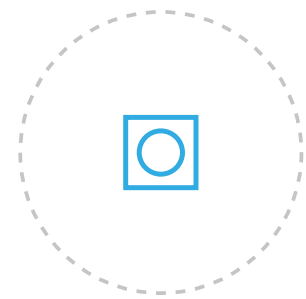

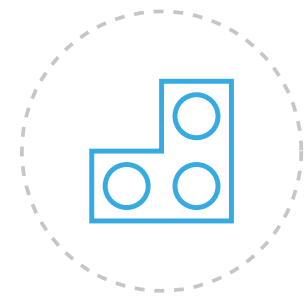
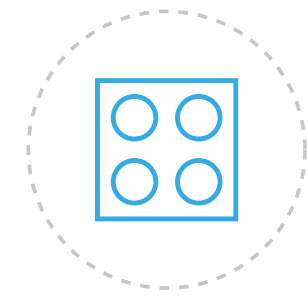




# Vaardigheden voor computational thinking evalueren

Naam: \_\_\_\_\_

Klas: \_\_\_\_\_

Abstractie	1. Beginner	2. Gevorderd	3. Bedreven	4. Expert	Opmerkingen
					
Het belangrijkste deel van je oplossing beschrijven.	De leerling is niet in staat om de oplossing te beschrijven. <input type="checkbox"/>	Na wat hulp is de leerling in staat om de oplossing te beschrijven. <input type="checkbox"/>	De leerling is in staat om de oplossing te beschrijven. <input type="checkbox"/>	De leerling is in staat om de oplossing te beschrijven en richt zich op het belangrijkste deel van de oplossing. <input type="checkbox"/>	
De belangrijkste details van je oplossing beschrijven.	De leerling is niet in staat om de details van de oplossing te beschrijven. <input type="checkbox"/>	Na wat hulp is de leerling in staat om de details van de oplossing te beschrijven. <input type="checkbox"/>	De leerling is in staat om de details van de oplossing te bespreken, maar sommige details zijn niet belangrijk. <input type="checkbox"/>	De leerling is in staat om de belangrijkste details van de oplossing te bespreken. <input type="checkbox"/>	
Beschrijven hoe je oplossing aan de oorspronkelijke criteria voldoet.	De leerling is niet in staat om te beschrijven hoe de oplossing aan de oorspronkelijke criteria voldoet. <input type="checkbox"/>	Na wat hulp is de leerling in staat om te beschrijven hoe de oplossing aan de oorspronkelijke criteria voldoet. <input type="checkbox"/>	De leerling is in staat om te beschrijven hoe de oplossing aan de oorspronkelijke criteria voldoet. <input type="checkbox"/>	De leerling is in staat om bijzonder duidelijk te beschrijven hoe de oplossing aan de oorspronkelijke criteria voldoet. <input type="checkbox"/>	

# LEGO® Education WeDo 2.0



LEGOeducation.com

LEGO and the LEGO logo are trademarks of the/son des marques de commerce du/son marcas registradas de LEGO Group. ©2017 The LEGO Group. 01.01.2017. - V1.

