



## Programmeren in het PO

Auteurs  
Team

Wietse van Bruggen ; Kennisnet LleG  
Stichting Kennisnet  
j.devreede@kennisnet.nl

Laatst gewijzigd  
Licentie  
Webadres

18 mei 2022  
CC Naamsvermelding 3.0 Nederland licentie  
<https://maken.wikiwijs.nl/74282/>



Dit lesmateriaal is gemaakt met Wikiwijs van Kennisnet. Wikiwijs is hét onderwijsplatform waar je leermiddelen zoekt, maakt en deelt.

# Inhoudsopgave

Inleiding .....	2
Verantwoording .....	3
1. Algoritmes .....	5
Onderbouw .....	5
Middenbouw .....	6
Bovenbouw .....	7
2. Decompositie .....	8
Onderbouw .....	8
Middenbouw .....	9
3. Patronen .....	10
Onderbouw .....	10
Middenbouw .....	11
4. Herhaling .....	12
Onderbouw .....	12
Bovenbouw .....	13
5. Fouten .....	14
Onderbouw .....	14
Middenbouw .....	15
6. Voorwaarden .....	16
Onderbouw .....	16
Middenbouw .....	17
Bovenbouw .....	18
7. Abstractie .....	19
Middenbouw .....	19
Bovenbouw .....	20
8. Functie .....	21
Middenbouw .....	21
Bovenbouw .....	22
9. Variabele .....	24
Middenbouw .....	24
Bovenbouw .....	25
10. Representatie .....	26
Middenbouw .....	26
Bovenbouw .....	27
Over dit lesmateriaal .....	28

# Inleiding

## Welkom bij de leerlijn programmeren in het basisonderwijs.

Deze leerlijn programmeren is bedoeld voor onderbouw, middenbouw, bovenbouw in het basisonderwijs. In de leerlijn komen begrippen en principes aan de orde die horen bij programmeren. Het bestaat uit uitleg over basisbegrippen van programmeren, doelen en activiteiten voor in de klas voor de leerlingen.

## Begrippen, doelen en activiteiten

De activiteiten die bij de begrippen staan zijn laagdrempelig en grotendeels zogenaamde "unplugged" activiteiten. Dat wil zeggen dat deze activiteiten geen gebruik maken van ICT en devices. Ze bieden een introductie tot de begrippen en in de hogere leerjaren worden ze verder uitgediept.

De activiteiten zijn suggesties en voorbeelden van activiteiten om aan de doelen te werken die bij het begrip horen. Je kunt uiteraard ook andere en aanvullende activiteiten en doen dan deze doen om aan de begrippen te werken.

## Workshop

Kennisnet heeft een workshop voor leraren gemaakt om bekend te raken met programmeren en computational thinking. Deze workshop is hier te vinden:

[http://maken.wikiwijs.nl/70012/Workshop\\_Computational\\_thinking#!page-1652293](http://maken.wikiwijs.nl/70012/Workshop_Computational_thinking#!page-1652293)

## Computational thinking

De activiteiten zijn gericht op programmeren en bevorderen computational thinking en het probleemoplossend vermogen bij leerlingen. De definitie van SLO van computational thinking is:

"Computational thinking is het procesmatig (her)formuleren van problemen op een zodanige manier dat het mogelijk wordt om met computertechnologie het probleem op te lossen. Het gaat daarbij om een verzameling van denkprocessen waarbij probleemformulering, gegevensorganisatie, -analyse en -representatie worden gebruikt voor het oplossen van problemen met behulp van ICT-technieken en -gereedschappen."

Computational thinking is onderdeel geworden van de 21<sup>e</sup> eeuwse vaardigheden. Programmeren is een manier waarop leerlingen aan hun computational thinking vaardigheden kunnen werken.

Meer informatie over computational thinking is te vinden op de website Curriculum van de Toekomst van SLO:

<http://curriculumvandetoekomst.slo.nl/21e-eeuwse-vaardigheden/computational-thinking>

Bij programmeren komen ook andere 21<sup>e</sup> eeuwse vaardigheden aan de orde. Creatief denken en handelen, ICT-basisvaardigheden, kritisch denken, mediawijsheid, probleemoplossend denken en handelen zijn vaardigheden die hieraan relateren. In de leerlijn richten we ons echter op computational thinking, geoperationaliseerd middels programmeren.

Deze leerlijn is tot stand gekomen vanuit een versnellingsvraag van het PO Doorbraakproject Slimmer Leren met ICT en is tot stand gekomen door de inzet van de besturen Oponoa en FIER, met hulp van SLO en met ondersteuning van Kennisnet.

# Verantwoording

## Verantwoording Leerlijn programmeren, Wikiwijs arrangement

Deze leerlijn is tot stand gekomen vanuit een versnellingsvraag van het PO Doorbraakproject Slimmer Leren met ICT 2015-2016. Dankzij de inzet van de besturen [OPONOA](#) (Gelderland) en onderwijsgroep [FIER](#) (Friesland), met hulp van [SLO](#) en met ondersteuning van [Kennisnet](#).

Bij de totstandkoming van deze leerlijn stonden de besturen aan het roer. Op deze pagina wordt een aantal van de belangrijkste keuzes beschreven die zijn gemaakt tijdens het inhoudelijke proces om te komen tot een gestructureerd aanbod.

Met als doel: iedereen kan aan de slag met programmeren in de klas!

### Leerlijn programmeren als arrangement, startpunt of eindpunt

Deze leerlijn is als Wikiwijs arrangement beschikbaar gesteld. Dat stelt iedereen die dat maar wil in de gelegenheid om een eigen invulling te geven aan de lijn. En waar dat wenselijk is, aan te vullen, uit te breiden etc. Iedere school kan hierin eigen keuzes maken.

De activiteiten dienen als voorbeeld en inspiratie. Het arrangement is een verzameling begrippen, doelen en activiteiten en biedt de mogelijkheid om uit te breiden, met andere activiteiten en met online tools en software. Dit aanbod groeit dagelijks.

Iedere school kan hiermee eigen keuzes maken, afhankelijk van aanwezige voorkennis over programmeren, infrastructuur, devices en ambities. Bij tools en activiteiten kun je kijken welke elementen uit de leerlijn in de tools aan bod komen, om zo tot zonder enige verplichting tot een eigen invulling te kunnen komen.

### Computational thinking

SLO definieert computational thinking als volgt:

"Computational thinking is het procesmatig (her)formuleren van problemen op een zodanige manier dat het mogelijk wordt om met computertechnologie het probleem op te lossen. Het gaat daarbij om een verzameling van denkprocessen waarbij probleemformulering, gegevensorganisatie, -analyse en -representatie worden gebruikt voor het oplossen van problemen met behulp van ICT-technieken en -gereedschappen."

In het zgn. *voorbeeldmatig leerplankader* van SLO is een verdere, *voorbeeldmatige* uitwerking te vinden van de begrippen en doelen voor computational thinking. Deze is te vinden op:

<http://curriculumvandetoekomst.slo.nl/21e-eeuwse-vaardigheden/computational-thinking/voorbeeldmatig-leerplankader>

Voor de leerlijn is gekozen om niet deze begrippen te kopiëren, maar om "eigen" begrippen te gebruiken. Deze keuze komt voort uit de gedachte om de leerlijn goed praktisch toepasbaar te maken voor leerkrachten, met eenduidige en herkenbare begrippen.

Wel hebben we bij elk begrip een verbinding proberen te leggen met de SLO computational thinking doelen. Programmeren is hier een onderdeel van. Bij elk begrip dat we behandelen is een pagina opgenomen waarin we aangeven welke computational thinking doelen verbonden zijn aan het begrip. Zo werken leerlingen ook aan die begrippen en doelen, maar hanteert de leerlijn een eigen set begrippen toegespitst op programmeren.

## Begrippen

De leerlijn programmeren is opgebouwd aan de hand van tien begrippen. Deze begrippen zien we als de "grammatica" van het programmeren. Met kennis en ervaring over deze concepten kun je met programmeren aan de slag. Bij elk begrip staat een korte toelichting ter verduidelijking van wat het begrip inhoudt.

We hebben daarbij nagedacht over de volgorde van de begrippen. De begrippen nemen qua complexiteit toe, onder andere door toenemende abstractie en doordat de begrippen meer en meer met elkaar in verbinding komen te staan.

## Doelen

Voor de tien begrippen zijn doelen geformuleerd. De onderverdeling die daarbij is gehanteerd: onderbouw, middenbouw en bovenbouw. Per domein zijn waar relevant door ons specifieke doelen geformuleerd. Deze doelen borduren voort op de voorgaande doelen per begrip en per "bouw".

Na onderbouw is het logisch om de doelen voor respectievelijk midden- en bovenbouw te behandelen. In de volgorde van de begrippen is gekeken of verdieping nodig is in elke "bouw". Voor sommige begrippen belichten we uitsluitend doelen voor bijvoorbeeld onderbouw en middenbouw. Of alleen middenbouw-bovenbouw. Dit komt voort uit het idee dat sommige begrippen aan de orde komen bij andere doelen of geen verdieping meer behoeven. *Decompositie* bijvoorbeeld is een begrip dat vaker aan de orde zal komen bij de behandeling van de andere begrippen.

Uiteraard staat het iedereen vrij om hier eigen keuzes in te maken, door bijvoorbeeld te kiezen om in de onderbouw de activiteiten rond programmeren nog niet aan te bieden. En deze door te schuiven naar de middenbouw. Of om ervoor te kiezen om met leerlingen uit de middenbouw die meer willen doen eerder te gaan werken aan bovenbouw doelen.

## Activiteiten

Bij de verschillende doelen en leeftijdsgroepen (onderbouw, middenbouw en bovenbouw) staan voorbeeldactiviteiten waarmee je werkt aan de geformuleerde doelen. Deze voorbeeldactiviteiten zijn zogenaamde "unplugged" activiteiten, waarmee je kunt werken aan het leren programmeren/ computational thinking zonder daarbij veel voorkennis, bepaalde hardware, software en infrastructuur bij nodig te hebben.

Hierdoor zijn ze door iedereen te gebruiken met zo min mogelijk drempels.

Met dit materiaal leg je de basis voor programmeren, ongeacht welke programmeertaal of -tool men hierna zou willen gebruiken. De ontwikkelingen gaan snel: wat nu nog beschikbaar is, zou morgen bijvoorbeeld achterhaald en niet meer beschikbaar kunnen zijn.

Veel van deze voorbeeldactiviteiten zijn afkomstig van het code.org platform, een Amerikaanse website met activiteiten die beschikbaar zijn gesteld onder een CC-BY-NC-SA 4.0 licentie en zijn vertaald naar het Nederlands. Bij sommige activiteiten zijn aanpassingen gedaan om ze aan te passen aan de Nederlandse context, zoals bijvoorbeeld liedjes die gebruikt worden in een activiteit.

# 1. Algoritmes

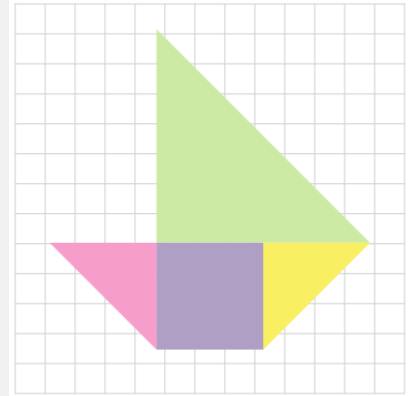
## Omschrijving

Een **reeks instructies**, die wanneer deze stap voor stap in de juiste volgorde uitgevoerd worden, tot een vooraf vastgesteld doel leiden.

## Koppeling met doelen SLO:

### Algoritmes en procedures

- Kan door algoritmisch redeneren oplossingen genereren
- Kan oplossingen automatiseren door middel van algoritmisch denken
- Kan een computerprogramma schrijven in code
- Kan een proces om problemen op te lossen generaliseren, zodat het ook bij andere problemen toegepast kan worden



### Problemen (her)fomuleren

- Kan mogelijke oplossingen analyseren om de meest kansrijke richting te bepalen

### Probleem decompositie

- Kan een taak opdelen in kleinere taken

### Abstractie

- Kan complexiteit reduceren en algemene concepten overbrengen

### Automatisering

- Kan door het opstellen van een serie van geordende stappen een probleem oplossen of een bepaald doel bereiken
- Kan effectieve en efficiënte stappen zetten en bronnen gebruiken om tot een uiteindelijke oplossing te komen
- Kan mogelijke oplossingen identificeren, analyseren en implementeren met als doel de meest effectieve en efficiënte oplossing te vinden

## Onderbouw

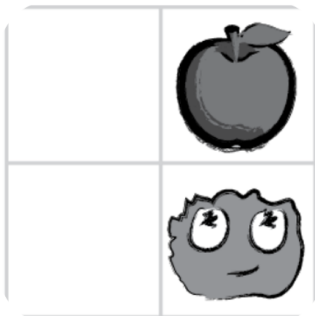
### Doelen voor de onderbouw

- Leert het begrip algoritme kennen
- Leert dat een algoritme een lijst van instructies is die leiden tot bepaald resultaat.
- Leren een simpel algoritme te maken in een concrete situatie met een vaste, van te voren bepaalde set instructies.

- Leert dat je een algoritme om kunt zetten naar code zodat het een programma wordt dat een computer uit kan voeren..

### Mogelijke activiteiten

#### 1: De goede weg (unplugged)



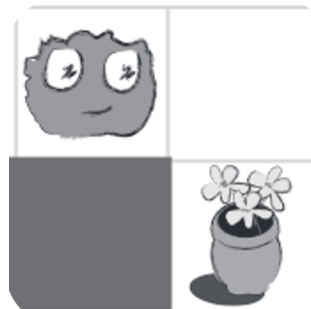
activiteit

leerkracht

Welke kant moet de Flurb  
op om bij het fruit de  
komen?

(unplugged)

#### 2: Stappenplan (unplugged)



activiteit

assessment

leerkracht

Help de Flurb bij haar  
bloemen te komen!

## Middenbouw

### Doelen voor de middenbouw

- Leert hoe complex het is om echte problemen om te zetten in programma's.
- Leert dat ideeën helder kunnen lijken, maar wellicht nog steeds verkeerd "begrepen" kunnen worden door een computer.
- Oefent hoe ideeën gecommuniceerd kunnen worden middels codes en symbolen.
- Leert het begrip algoritme te verbinden met alledaagse situaties.
- Leert grote activiteiten op te delen in deelopdrachten.
- Leert hoe deelopdrachten in een logische volgorde geplaatst kunnen worden.

### Mogelijke activiteiten

#### 1: Programmeren op

#### 2: Papieren vliegtuig

### ruitjes-papier (unplugged)



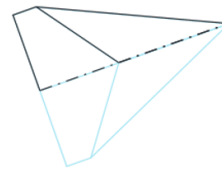
Volg de instructies op en  
maak een tekening op  
ruitjespapier.

activiteit

assessment

leerkracht

### (unplugged)



VOUW PAPIER WEER  
DOORMIDDEN

Vol de instructies en  
maak een vliegtuig.

activiteit

assessment

leerkracht

## Bovenbouw

### Doelen voor de bovenbouw

- Leert een algoritme te maken dat hergebruikt kan worden.
- Leert verschillende algoritmes te evalueren.
- Leert een probleem op te lossen door een eigen set instructies te formuleren.

### Mogelijke activiteiten

#### 1. Dobbelstenenrace (unplugged)



Denk als een computer  
tijdens een spelletje  
dobbelen.

activiteit

assessment

leerkracht

#### 2. Algoritmes (unplugged)



Leg een figuur op basis van  
aanwijzingen van je  
klasgenoot.

activiteit

assessment

leerkracht



## 2. Decompositie

### Omschrijving

Een probleem onderzoeken en uitsplitsen in deelproblemen om het op te kunnen lossen. (Stappenplan met kleine probleempjes oplossen.)

### Koppeling met doelen SLO:

#### Algoritmes en procedures

- Kan door algoritmisch redeneren oplossingen genereren
- Kan een proces om problemen op te lossen generaliseren, zodat het ook bij andere problemen toegepast kan worden



#### Automatisering

- Kan mogelijke oplossingen identificeren, analyseren en implementeren met als doel de meest effectieve en efficiënte oplossing te vinden

#### Probleem decompositie

- Kan een taak opdelen in kleinere taken
- Kan een lange lijst met opdrachten opdelen in subcategorieën

## Onderbouw

### Doelen voor de onderbouw

- Leert een simpele activiteit op te delen in deelopdrachten.
- Leert hoe deelopdrachten in een logische volgorde geplaatst kunnen worden.

### Mogelijke activiteiten

**Een zaadje planten  
(unplugged)**

**activiteit**



leerkracht

Help elkaar om zaadjes te planten

## Middenbouw

### Doelen voor de middenbouw

- Leert een grote activiteit op te delen in deelopdrachten,
- Leert een concreet probleem op te delen in deelopdrachten
- Leert deze deelopdrachten in een logische volgorde te plaatsen, in een algoritme te vertalen

**Knikkerbaan  
(unplugged)**

leerkracht



Uit welke regels en ontwerpkeuzes bestaat een knikkerbaan?

# 3. Patronen

## Omschrijving

Patronen (model, vorm, kleur, voorbeeld, herhalingen) herkennen.

## Koppeling met doelen SLO:

### Gegevens analyseren

- Kan gegevens logisch ordenen en begrijpen
- Kan patronen vinden en conclusies trekken

### Algoritmes en procedures

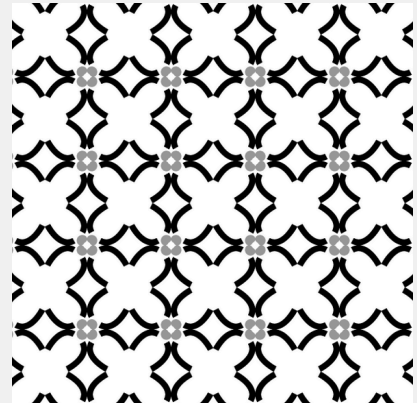
- Kan door algoritmisch redeneren oplossingen genereren
- Kan een proces om problemen op te lossen generaliseren, zodat het ook bij andere problemen toegepast kan worden

### Probleem decompositie

- Kan een aantal taken combineren tot één taak

### Automatisering

- Kan door het opstellen van een serie van geordende stappen een probleem oplossen of een bepaald doel bereiken
- Kan repetitieve taken laten uitvoeren door computers



# Onderbouw

## Doelen voor de onderbouw

- leert wat patronen zijn in relatie tot computational thinking
- leert (voornamelijk visuele) patronen te herkennen

## Mogelijke activiteiten

**Patronen in de  
omgeving (unplugged)**



leerkracht

Welke patronen kun je in je omgeving vinden?

## Middenbouw

### Doelen voor de middenbouw

- leert patronen te herkennen in abstracte situaties (zoals bijvoorbeeld cijfer reeksen)
- leert hoe patronen gebruikt kunnen worden binnen het maken van algoritmes en programma's

### Mogelijke activiteiten

#### Patronen in cijferreeksen (unplugged)



leerkracht

Welke patronen vind je in abstractere situaties zoals cijferreeksen?

## 4. Herhaling

### Omschrijving

Een lus of loop zorgt ervoor dat iets herhaald wordt.

### Koppeling met doelen SLO:

#### Algoritmes en procedures

- Kan oplossingen automatiseren door middel van algoritmisch denken
- Kan een computerprogramma schrijven in code
- Kan een proces om problemen op te lossen generaliseren, zodat het ook bij andere problemen toegepast kan worden



#### Probleem decompositie

- Kan een taak opdelen in kleinere taken

#### Automatisering

- Kan door het opstellen van een serie van geordende stappen een probleem oplossen of een bepaald doel bereiken
- Kan repetitieve taken laten uitvoeren door computers
- Kan effectieve en efficiënte stappen zetten en bronnen gebruiken om tot een uiteindelijke oplossing te komen
- Kan mogelijke oplossingen identificeren, analyseren en implementeren met als doel de meest effectieve en efficiënte oplossing te vinden
- Kan repetitieve taken laten uitvoeren door computers

## Onderbouw

### Doelen voor de onderbouw

- Leert herhalingen herkennen
- Leert dat herhalingen activiteiten zijn die je meerdere keren uit kunt voeren.
- Leert dat herhalingen uitermate geschikt zijn om automatisch uit te laten voeren door een computer.

### Mogelijke activiteiten

**Doe de loop (unplugged)**



activiteit

assessment

leerkracht

Als je danst, maak je vaak dezelfde bewegingen. Soms is het handig te weten hoe vaak.

## Bovenbouw

### Doelen voor de bovenbouw

- Leert dat herhalingen in combinatie met voorwaarden gebruikt kunnen worden

### Mogelijke activiteiten

Deze activiteit gaat zowel over herhaling als over voorwaarden.

**For loop plezier  
(unplugged)**



activiteit

assessment

leerkracht

Starten en stoppen als een bepaalde waarde bereikt is.

## 5. Fouten

### Omschrijving

Fouten opsporen en fouten oplossen. Dit noemen we ook wel debuggen. (Probleemoplossend vermogen ondersteunen /ontwikkelen)

### Koppeling met doelen SLO:

#### Problemen (her)formuleren

- Kan op een zodanige manier problemen formuleren dat het mogelijk wordt om het probleem op te lossen door gebruik van een computer of ander gereedschap
- Kan mogelijke oplossingen analyseren om de meest kansrijke richting te bepalen



#### Automatisering

- Kan door het opstellen van een serie van geordende stappen een probleem oplossen of een bepaald doel bereiken
- Kan een proces om problemen op te lossen generaliseren, zodat het ook bij andere problemen toegepast kan worden

## Onderbouw

### Doelen voor de onderbouw

- Leert het begrip debuggen kennen.
- Leert simpele fouten te herkennen in een simpel proces/aantal achtereenvolgende stappen (lego/tangram)
- Leert hoe ze fouten kunnen herkennen met een aantal strategieën.

### Mogelijke activiteiten

**Puzzel 10: Nectar verzamelen**



leerkracht

Herstel de fouten om alle nectar te verzamelen.

[Nectar verzamelen](#)

## Middenbouw

### Doelen voor de middenbouw

- Leert fouten te herkennen in abstracte situaties.
- Leert fouten op te lossen nadat ze herkend zijn.

### Mogelijke activiteiten

#### Programmeer-estafette (unplugged)



activiteit

assessment

leerkracht

Welke groep is het eerst klaar?



## 6. Voorwaarden

### Omschrijving

Iets dat gedaan wordt als er aan bepaalde voorwaarden wordt voldaan. Als..., dan .....

### Koppeling met doelen SLO:

#### Algoritmes en procedures

- Kan oplossingen automatiseren door middel van algoritmisch denken
- Kan een computerprogramma schrijven in code
- Kan een proces om problemen op te lossen generaliseren, zodat het ook bij andere problemen toegepast kan worden



#### Automatisering

- Kan twee verschillende concepten vergelijken en deze logisch verbinden
- Kan door algoritmisch redeneren oplossingen genereren

#### Parallelization

- Kan taken gelijktijdig laten uitvoeren door computers

## Onderbouw

### Doelen voor de onderbouw

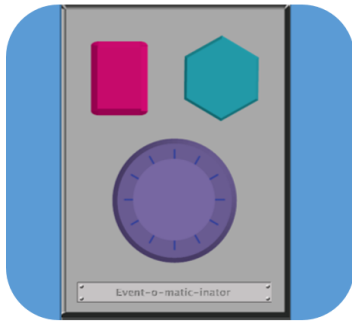
- Leert het begrip voorwaarde kennen.
- Leert op basis van te voren bepaalde tekens en acties met voorwaarden te werken.

### Mogelijke activiteiten

Hee, een event  
(unplugged)

activiteit

assessment



leerkracht

Laat de klas reageren als je op de knop drukt van de event-o-matic-inator

## Middenbouw

### Doelen voor de middenbouw

- Leert voorwaarden toe te passen binnen een programma waardoor gedeeltes van het programma wel of niet zullen worden uitgevoerd.
- Kan uitleggen onder welke voorwaarden een gedeelte van het programma wel of niet zal worden uitgevoerd.

### Mogelijke activiteiten

**Conditionele statements met kaarten (unplugged)**



activiteit

assessment

leerkracht

ALS de kaart rood is,  
DAN...

# Bovenbouw

## Doelen voor de bovenbouw

- Leert zelf bedachte condities toe te passen in een programma. (voortbouwen op vorige opdracht middenbouw voorwaarde)

## Mogelijke activiteiten

Deze activiteit gaat zowel over herhaling als over voorwaarden.

**For loop plezier  
(unplugged)**



activiteit

assessment

leerkracht

Starten en stoppen als  
een bepaalde waarde  
bereikt is.

# 7. Abstractie

## Omschrijving

Het veralgemeniseren van verschillende problemen, door de verschillen weg te laten. Een oplossing vinden die bruikbaar is voor verschillende problemen door specifieke verschillen weg te laten.

## Koppeling met doelen SLO:

### Problemen (her)formuleren

- Kan op een zodanige manier problemen formuleren dat het mogelijk wordt om het probleem op te lossen door gebruik van een computer of ander gereedschap



### Probleem decompositie

- Kan een taak opdelen in kleinere, behapbare taken
- Kan een aantal taken combineren tot één taak
- Kan complexiteit reduceren en algemene concepten overbrengen

### Abstractie

- Kan complexiteit reduceren en algemene concepten overbrengen
- Kan twee verschillende concepten vergelijken en deze logisch verbinden

## Middenbouw

### Doelen voor de middenbouw

- Leert het begrip abstractie kennen.
- Leert abstraheerbare elementen te herkennen.
- Leert verschillen te identificeren in vergelijkbare situaties en deze verschillen te abstraheren.

### Mogelijke activiteiten

Hee, een event!  
(Unplugged)





assessment

leerkracht

Achterhaal de regels van  
een onbekend spel

## Bovenbouw

### Doelen voor de bovenbouw

- Leert abstraheren toe te passen om zelf iets nieuws te creëren.

### Mogelijke activiteiten

#### Gekke Glibs (unplugged)



activiteit

assessment

leerkracht

Maak een verhaal met 'gaten'  
af.

# 8. Functie

## Omschrijving

Een "hulp"programma binnen het "hoofd"programma dat hergebruikt kan worden. Wordt ook wel genoemd sub routine.

## Koppeling met doelen SLO:

### Automatisering

- Kan door het opstellen van een serie van geordende stappen een probleem oplossen of een bepaald doel bereiken
- Kan repetitieve taken laten uitvoeren door computers

### Algoritmes en procedures

- Kan een proces om problemen op te lossen generaliseren, zodat het ook bij andere problemen toegepast kan worden

### Probleem decompositie

- Kan een taak opdelen in kleinere taken
- Kan een lange lijst met opdrachten opdelen in subcategorieën
- Kan een aantal taken combineren tot één taak

### Algoritmes en procedures

- Kan door algoritmisch redeneren oplossingen genereren

### Problemen (her)formuleren

- Kan op een zodanige manier problemen formuleren dat het mogelijk wordt om het probleem op te lossen door gebruik van een computer of ander gereedschap

### Simulatie en modellering

- Kan een proces representeren of een experiment uitvoeren op basis van modellen

### Parallelization

- Kan taken gelijktijdig laten uitvoeren door computers

## Middenbouw

### Doelen voor de middenbouw

- Leert het begrip functies kennen.
- Leert wat het doel van functies zijn.
- Leert acties te herkennen die in een functie kunnen worden beschreven.
- Leert een functie te definiëren
- Leert een functie aan te roepen.

### Mogelijke activiteiten

#### Functionele instructies (unplugged)



Vang het licht met kralen  
en draad

activiteit

assessment

leerkracht

#### Liedjes schrijven (unplugged)



Zoek de herhalingen in  
een lied.

activiteit

assessment

leerkracht

## Bovenbouw

### Doelen voor de bovenbouw

- Leert functies te combineren met parameters en variabelen.

### Mogelijke activiteiten

#### Liedjes schrijven met parameters (unplugged)

activiteit



**assessment**

**leerkracht**

Leer parameters kennen  
aan de hand van liedjes



# 9. Variabele

## Omschrijving

Een variabele is een waarde die varieert en moet worden opgeslagen om te kunnen vergelijken met andere waarden.

## Koppeling doelen SLO:

### Algoritmes en procedures

- Kan een proces om problemen op te lossen generaliseren, zodat het ook bij andere problemen toegepast kan worden
- Kan door algoritmisch redeneren oplossingen genereren
- Kan oplossingen automatiseren door middel van algoritmisch denken
- Kan een computerprogramma schrijven in code



### Gegevens analyseren

- Kan gegevens logisch ordenen en begrijpen

### Abstractie

- Kan twee verschillende concepten vergelijken en deze logisch verbinden

### Parallelization

- Kan middelen op een dergelijke wijze organiseren dat het mogelijk wordt om ze simultaan in te zetten om een gezamenlijk doel te bereiken

## Middenbouw

### Doelen voor de middenbouw

- Leert het begrip variabele kennen.
- Leert het begrip toe te passen.

### Mogelijke activiteiten

**Variabelen in  
enveloppen  
(unplugged)**

**activiteit**



assessment

leerkracht

Leerlingen ontwerpen  
een robot. Daarna  
worden de details  
klassikaal in variabelen  
gevat

## Bovenbouw

### Doelen voor de bovenbouw

- Leert het begrip toe te passen in abstracte situaties.

Voor de bovenbouw zijn geen specifieke activiteiten opgenomen voor het begrip variabele. Het komt terug in de activiteiten voor andere begrippen in de bovenbouw.

# 10. Representatie

## Omschrijving

Het weergeven en organiseren van gegevens in passende grafieken, lijsten, teksten of plaatjes.

## Koppeling met doelen SLO:



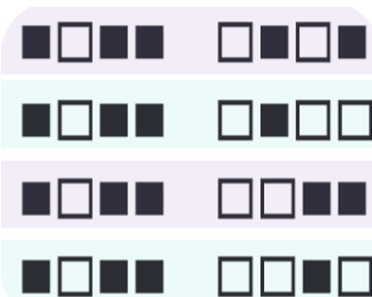
## Middenbouw

### Doelen voor de middenbouw

- Leert dat een computer uiteindelijk een machine is die werkt met nullen en enen om te werken.
- Leert hoe simpele informatie zoals cijfers en letters gerepresenteerd kunnen worden middels binaire codes.

### Mogelijke activiteiten

#### Binaire polsbandjes (unplugged)



activiteit

assessment

leerkracht

Maak je eigen armband met je naam in binaire code.

# Bovenbouw

## Doelen voor de bovenbouw

- Leert hoe verschillende soorten complexe gegevens gerepresenteerd kunnen worden middels binaire codes.

## Mogelijke activiteiten

### Afbeeldingen in binaire code (unplugged)



activiteit

assessment

leerkracht

Bedenk je eigen binaire code aan de hand van afbeeldingen.

# Over dit lesmateriaal

## Colofon

<b>Auteurs</b>	Wietse van Bruggen ; Kennisnet LleG
<b>Team</b>	Stichting Kennisnet j.devreede@kennisnet.nl
<b>Laatst gewijzigd</b>	18 mei 2022 om 16:41
<b>Licentie</b>	De Nederlandse Creative Commons 3.0 licentie waarbij de gebruiker het werk mag kopiëren, verspreiden en doorgeven en afgeleide werken mag maken onder de voorwaarde: Naamsvermelding, zie <a href="http://creativecommons.org/licenses/by/3.0/nl/">http://creativecommons.org/licenses/by/3.0/nl/</a> . <a href="#">Meer informatie over de CC Naamsvermelding 3.0 Nederland licentie licentie.</a>

## Aanvullende informatie over dit lesmateriaal

Van dit lesmateriaal is de volgende aanvullende informatie beschikbaar:

<b>Leerniveaus</b>	PO groep 3, PO groep 4, PO groep 1, PO groep 8, PO groep 7, PO groep 6, PO groep 5, PO groep 2
<b>Leerinhoud en doelen</b>	Computervaardigheden/ICT
<b>Eindgebruiker</b>	leraar
<b>Trefwoorden</b>	computational thinking, dgpo-401, dgpo-403, dgpo-404, dgpo-405, dgpo-406, dgpo-407, dgpo-408, dgpo-409, dgpo-410