Het belang van *computationeel denken* in het onderwijs

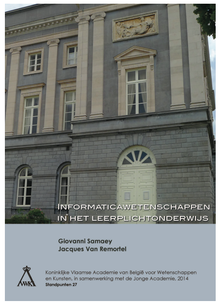
Prof. dr. Frank Neven

UHASSELT
KNOWLEDGE IN ACTION

Wie?

UHASSELT
KNOWLEDGE IN ACTION

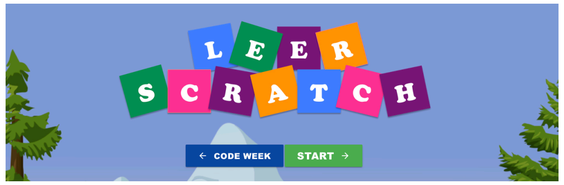
i22N
FORUM VOOR
INFORMATICA
WETENSCHAPPEN



CoderDojo

Progra-MEER

MEER dan enkel programmeren



Overzicht

1. Computationeel Denken (CD) uitgelegd
2. Plaats van CD in het leerplichtonderwijs
3. Relevantie van CD

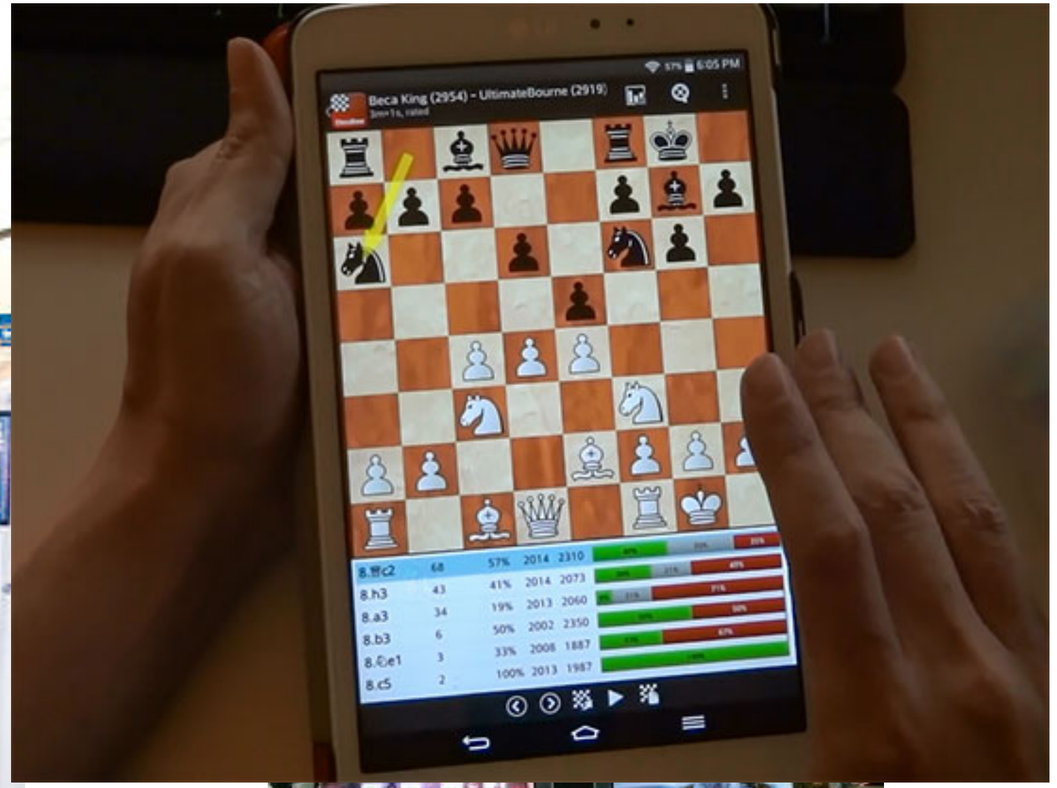
Overzicht

1. **Computationeel Denken (CD) uitgelegd**
2. Plaats van CD in het leerplichtonderwijs
3. Relevantie van CD

Zijn computer slim?

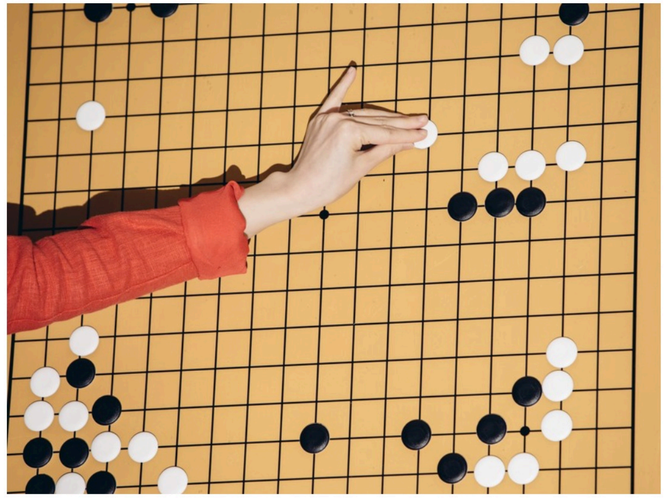
Garry Kasparov


IBM


Beca King (2954) - UltimateBourne (2919)
8.♕c2 68 57% 2014 2310
8.h3 43 41% 2014 2073
8.a3 34 19% 2013 2060
8.b3 6 50% 2002 2350
8.♘e1 3 33% 2008 1887
8.c5 2 100% 2013 1987

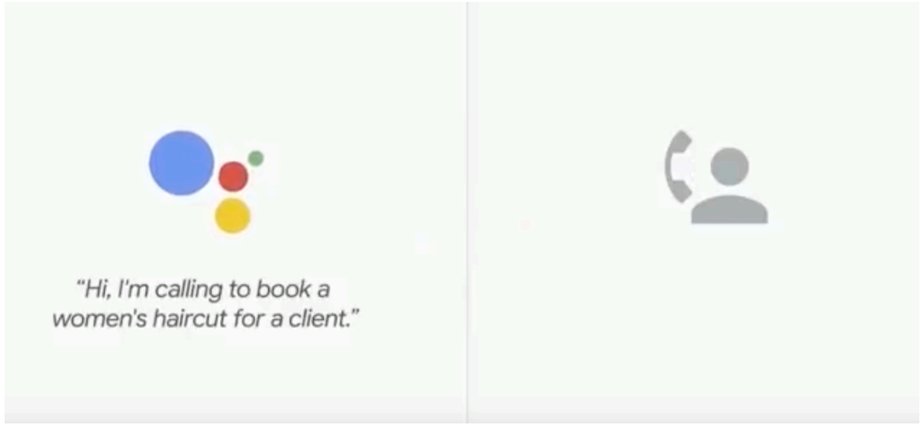
2011

GOOGLE'S AI WINS FIFTH AND FINAL GAME AGAINST GO GENIUS LEE SEDOL



2016

2019



2018

2019

Vertonen computers intelligent gedrag?

Zijn computer slim?

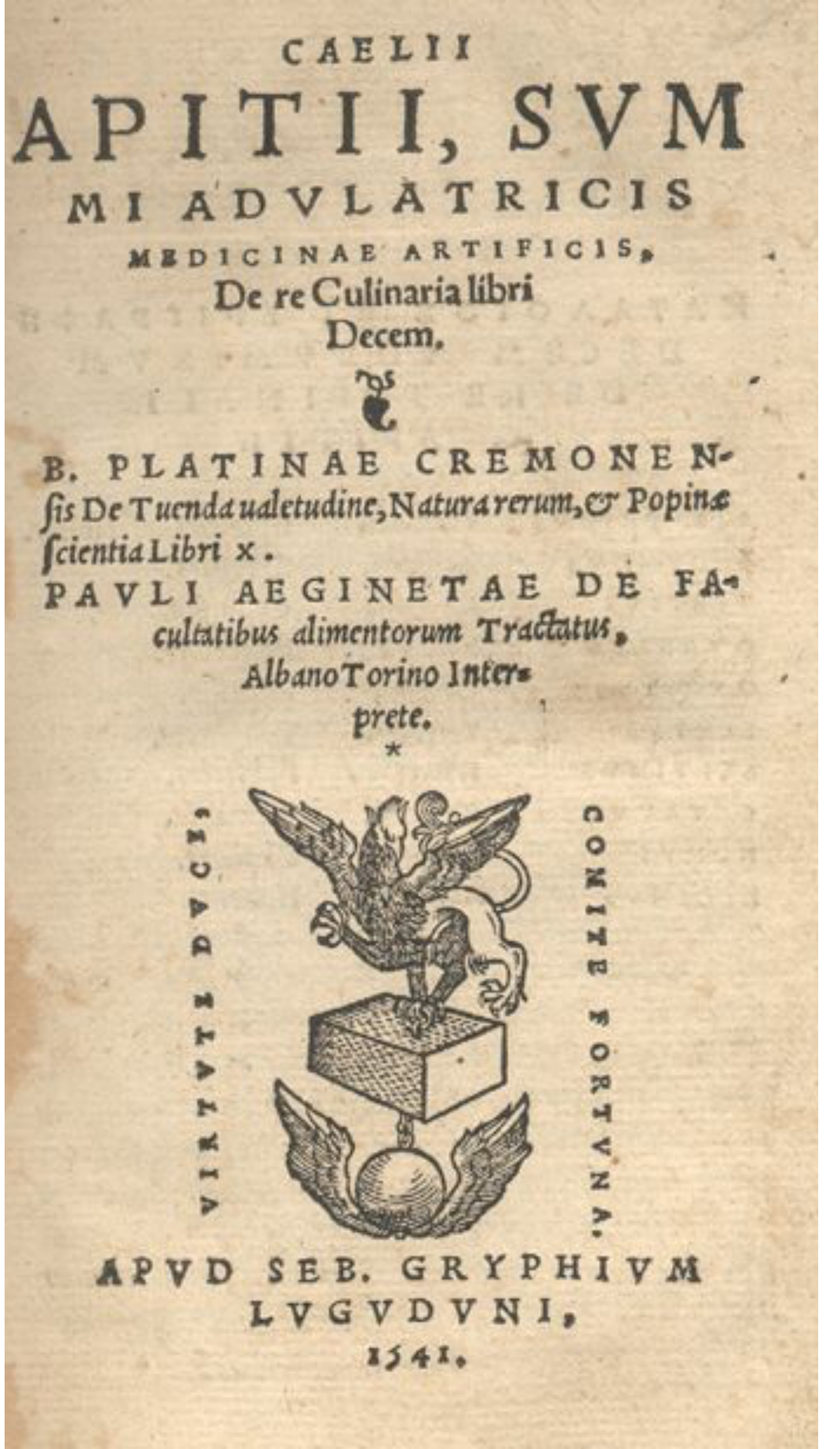
Computers voeren instructies uit

Overzicht

1. Computationeel Denken (CD) uitgelegd
    1. Zijn computers slim?
    2. **Algoritmen**
    3. Concepten en aanpakken
2. Plaats van CD in het leerplichtonderwijs
3. Relevantie van CD

ALGORITME

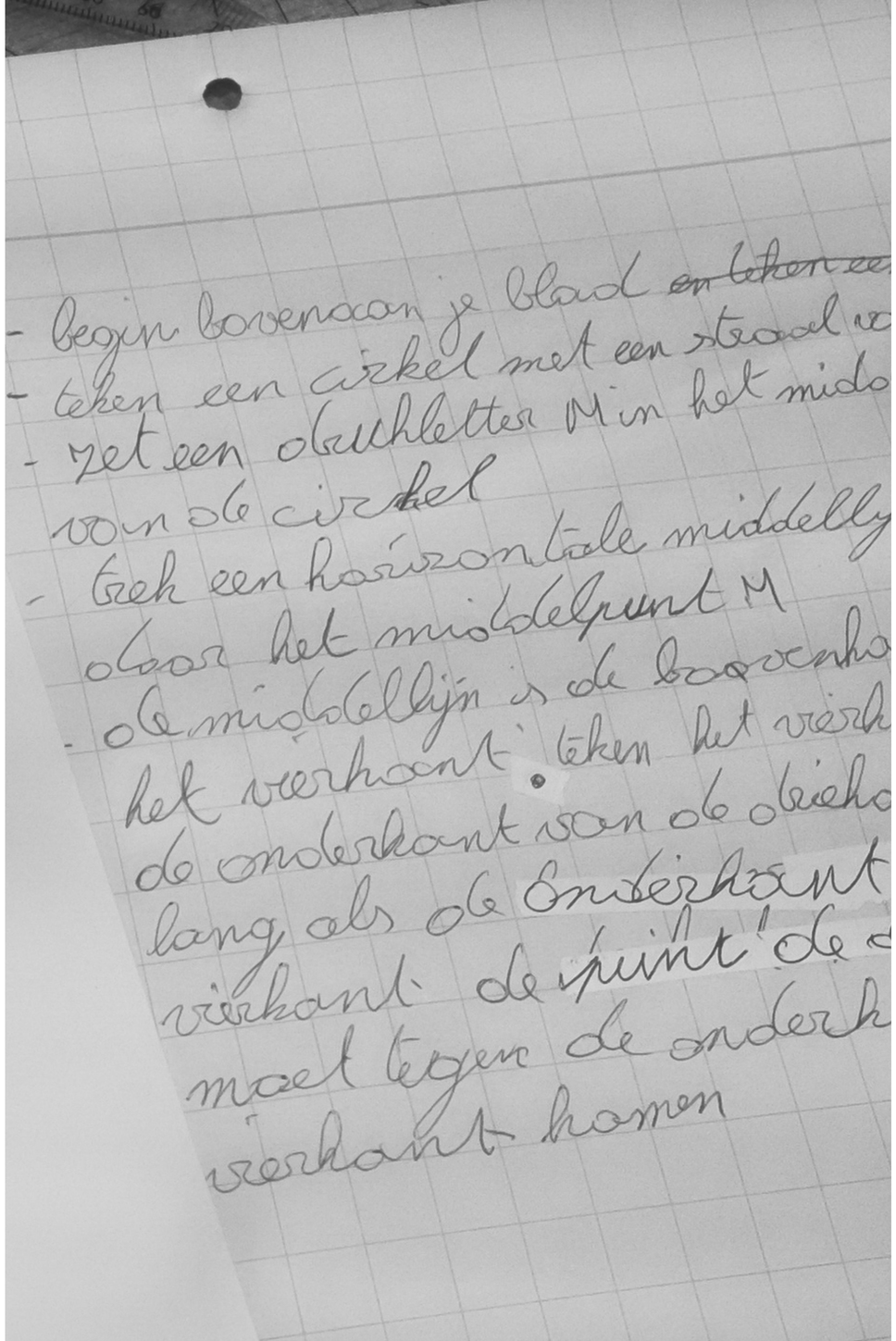
Geheel van instructies die stapsgewijs uitgevoerd kunnen worden om een probleem op te lossen


CAELII
APITII, SVM
MI ADVLATRICIS
MEDICINAE ARTIFICIS,
De re Culinaria libri
Decem.

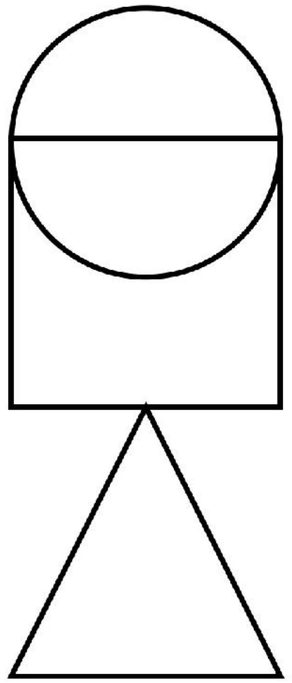
B. PLATINAE CREMONENsis De Tuenda ualetudine, Natura rerum, & Popinæ scientia Libri x.
PAVLI AEGINETAE DE FAcultatibus alimentorum Tractatus,
Albano Torino Interprete.

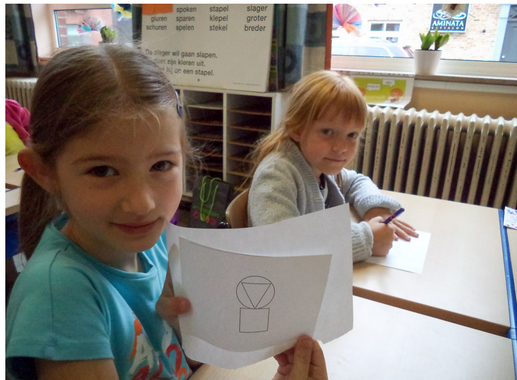
VIRTVTI DVCIS, COMITE FORTVNA.

APVD SEB. GRYPHIVM
LVGVDVNI,
1541.

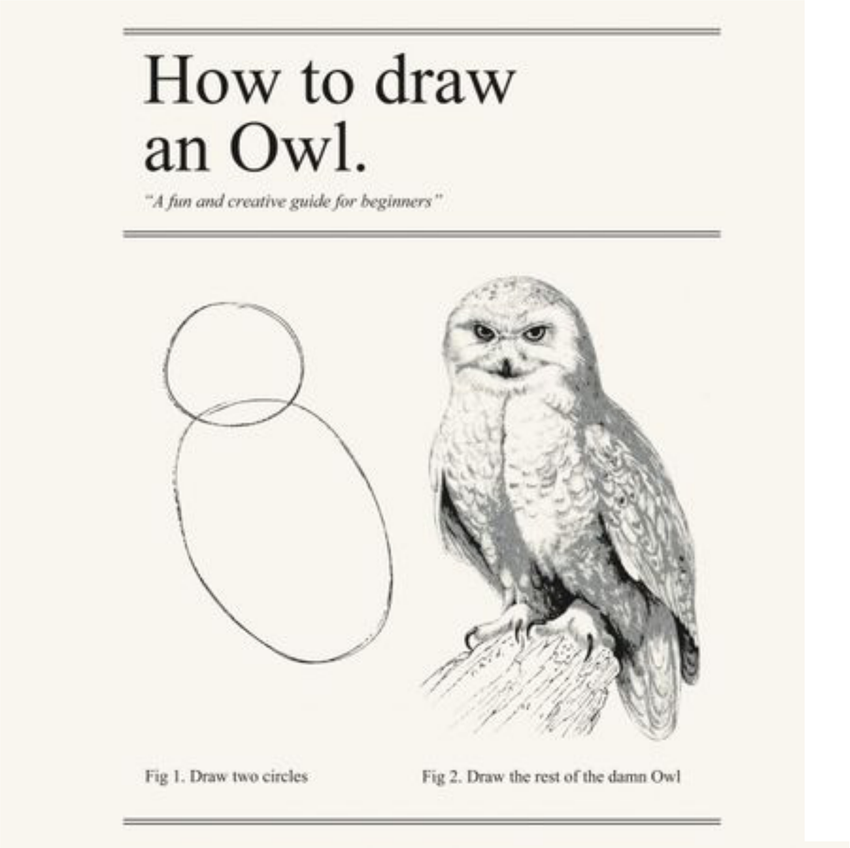


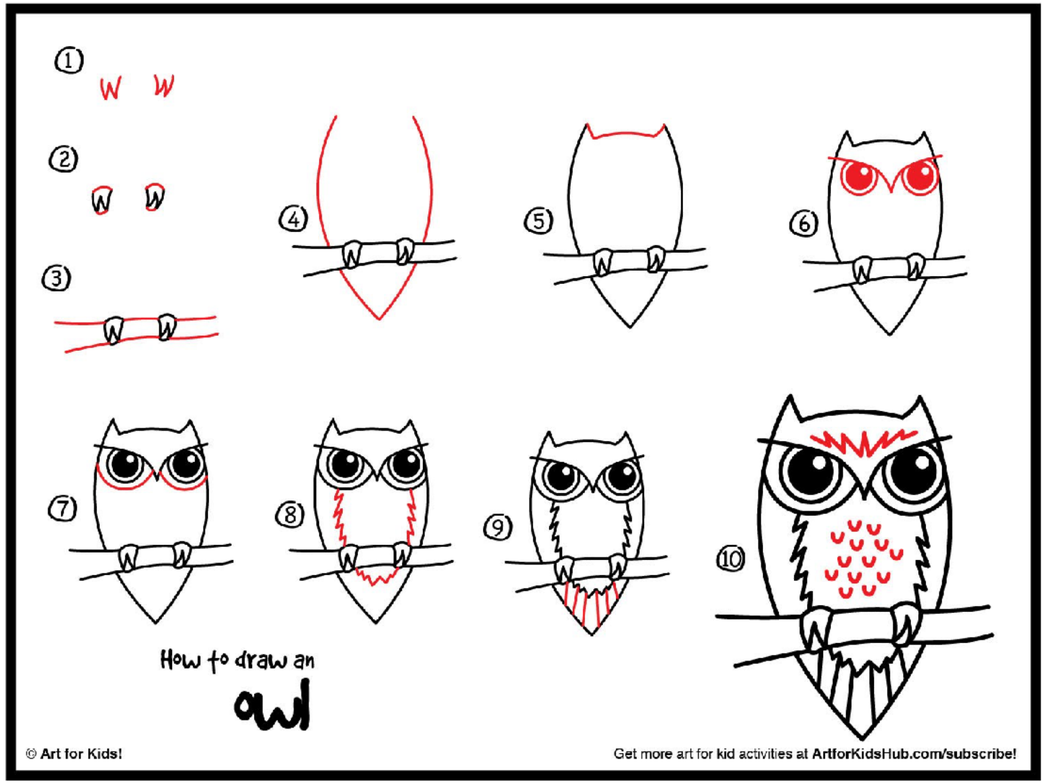
OPSTELLEN VAN ALGORITME





STAPSGEWIJZE PROCEDURE





Art for Kids

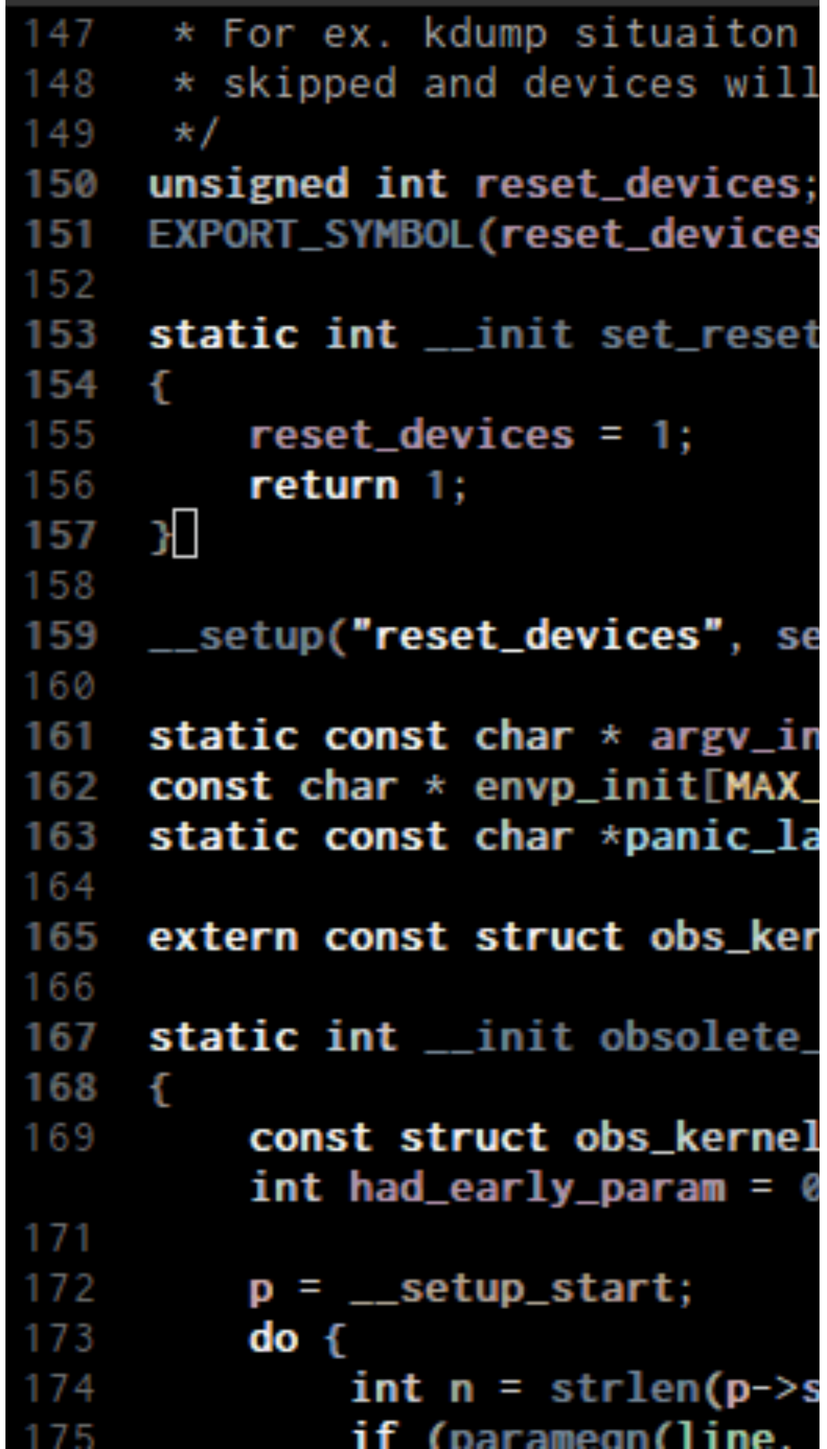
PROGRAMMEREN

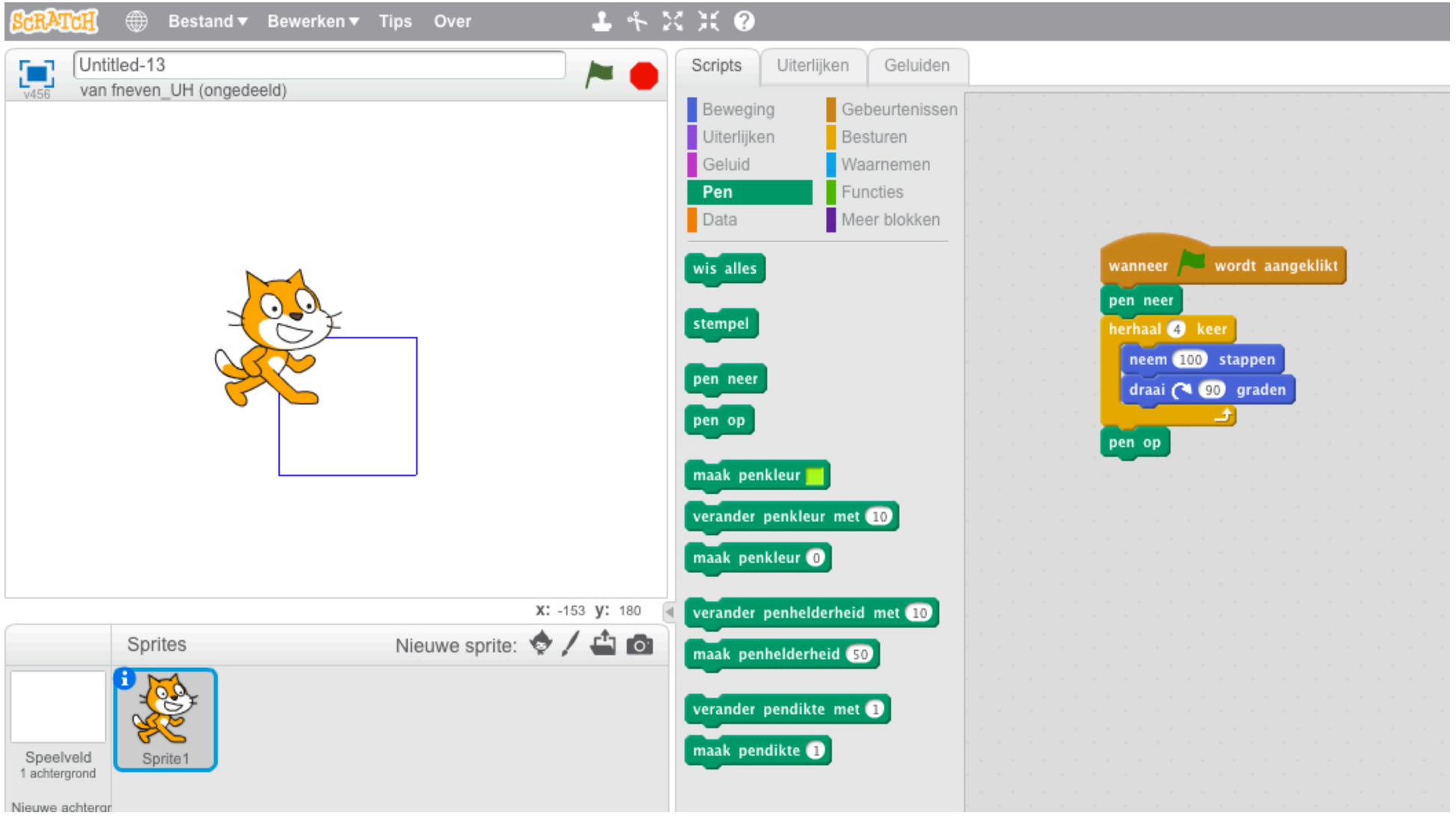
=

MAKEN VAN EEN ALGORITME DAT DE UITVOERDER BEGRIJPT

PROGRAMMA

Een algoritme dat rechtstreeks uitvoerbaar is door de computer




Bestand
Bewerken
Tips
Over
Untitled-13
van fneven_UH (ongedeeld)
Scripts
Uiterlijken
Geluiden
Beweging
Uiterlijken
Geluid
Pen
Data
Gebeurtenissen
Besturen
Waarnemen
Functies
Meer blokken
wis alles
stempel
pen neer
pen op
maak penkleur
verander penkleur met 10
maak penkleur 0
verander penhelderheid met 10
maak penhelderheid 50
verander pendikte met 1
maak pendikte 1
wanneer wordt aangeklikt
pen neer
herhaal 4 keer
neem 100 stappen
draai 90 graden
pen op
x: -153 y: 180
Sprites
Nieuwe sprite:
Speelveld
1 achtergrond
Sprite1

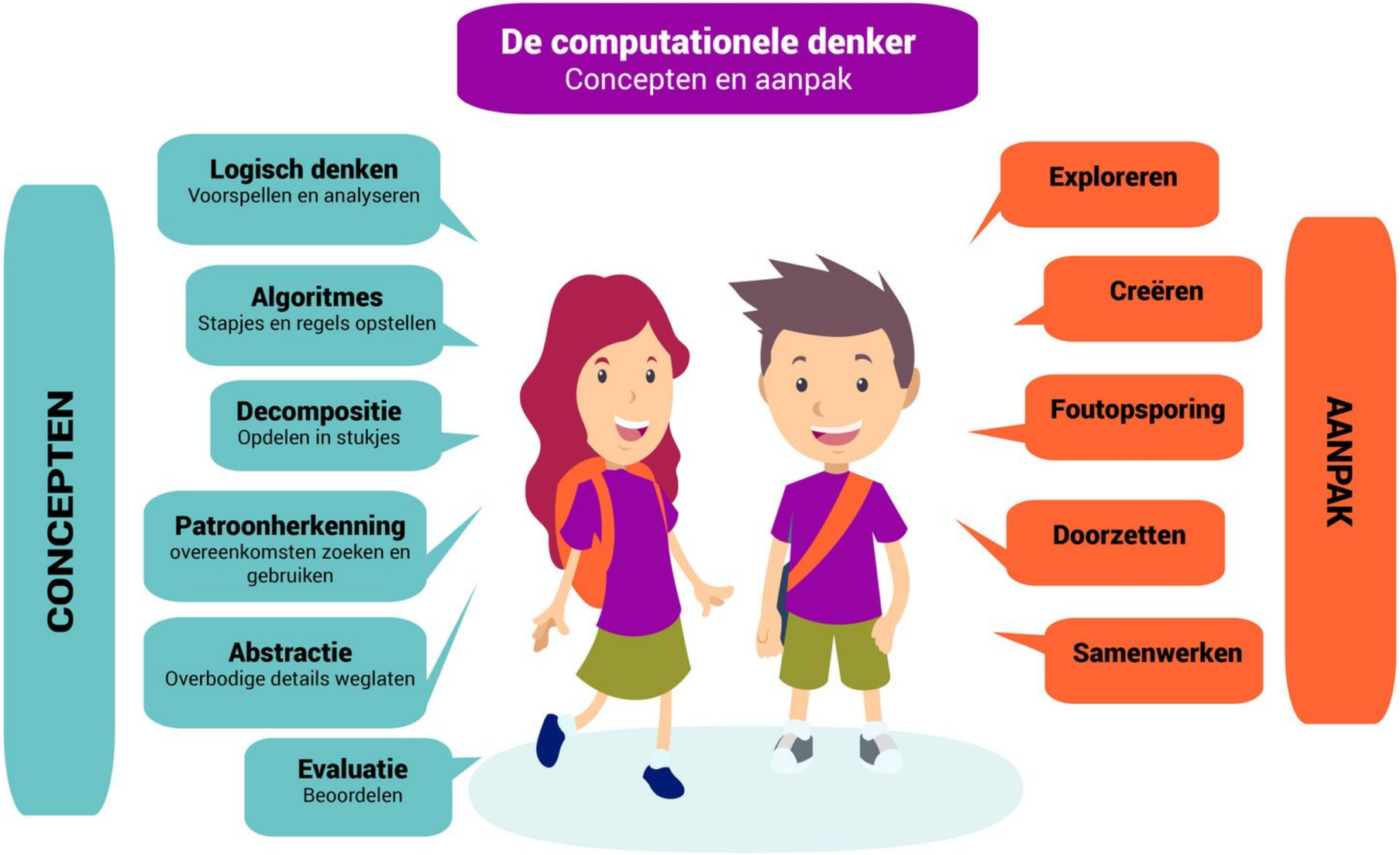
Overzicht

1. Computationeel Denken (CD) uitgelegd
    1. Zijn computers slim?
    2. Algoritmen
    3. **Concepten en aanpakken**
2. Plaats van CD in het leerplichtonderwijs
3. Relevantie van CD

“

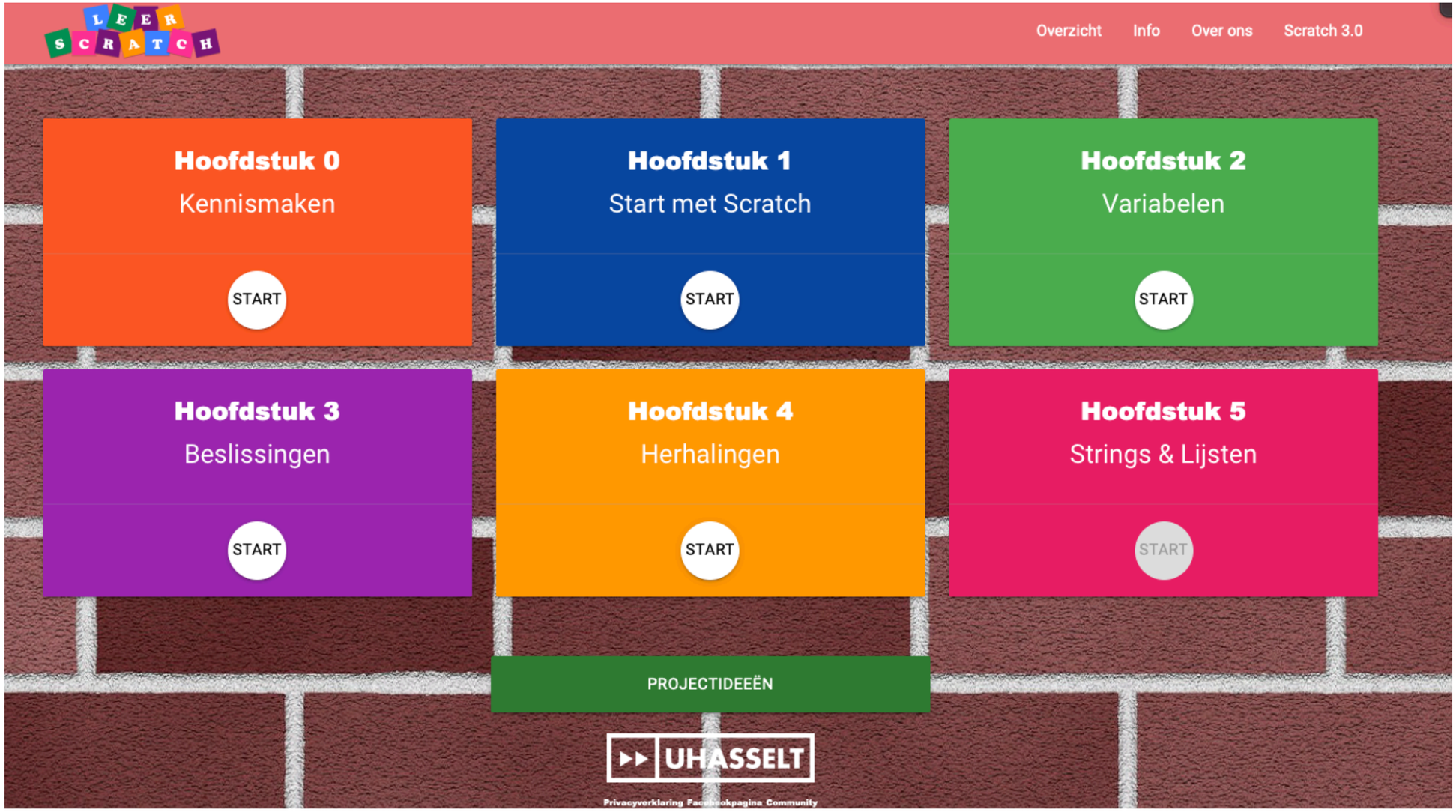
Computationeel denken is problemen op zo een manier benaderen dat computers kunnen gebruikt worden om ze op te lossen.

- Jeannette Wing



https://pro.katholiekonderwijs.vlaanderen/computationeel-denken

www.leer-scratch.be



Concepten

Algoritme

Een algoritme is, in het kort, een stapsgewijze procedure om een probleem op te lossen. We leggen dit begrip uit met behulp van enkele voorbeelden uit het dagelijks leven waar de kinderen mee vertrouwd zijn. Verder gebruiken we het begrip zelf ook echt in de cursus. Zo benoemen we oefeningen als 'het maken van een algoritme' wanneer dit het geval is (zie fig. 2).

(Figuur 2: Eerste eigen algoritme)

Zo krijgt de leerling een oefening (zie fig. 3) waarbij hij/zij Ruby in een vierkant moet laten wandelen. D
leerling zal dan eerst moeten nadenken over hoe hij/zij Ruby in een vierkant kan laten wandelen. Wat is he
juiste algoritme? Zodra dit algoritme gevonden is, kan het naar Scratch omgezet worden.

(Figuur 3: Wandelen in een vierkant)

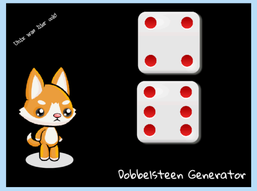
Abstractie

Met abstractie willen we overbodige details weglaten of bepaalde zaken op een andere manier voorstellen. Met Scratch gaan we hier niet zo diep op in. Doordat dit geen gemakkelijk begrip is, is het misschien ook wat meer weggelegd voor het secundair onderwijs.

Toch kunnen we met Scratch ook al abstractie toelichten aan de hand van variabelen. Variabelen vormen namelijk een manier om informatie op te slaan en hier een betekenis aan te geven. We gebruiken dit om bijvoorbeeld namen, getallen, een leeftijd, etc op te slaan.

Wanneer we in een spel de waarde van twee dobbelstenen willen weergeven (zie fig. 10), gebruiken we een variabele voor dobbelsteen één en een variabele voor dobbelsteen twee. Wordt zo een dobbelsteen geworpen, dan kennen we op een willekeurige manier getallen toe tussen één en zes. Zo een getal komt dan overeen met de worp van een dobbesIteen.

Niet alleen voor dobbelstenen, maar ook voor speelkaarten (zie fig. 11) kunnen variabelen gebruikt worden. Zo komen de waarden één tot en met tien overeen met de numerieke waarden van de speelkaarten, waarbij elf, twaalf en dertien respectievelijk de boer, dame en heer voorstellen.



(Figuur 10: Dobbelstenen)

(Figuur 11: Speelkaarten)

https://www.leer-scratch.be/info/computationeeldenken/

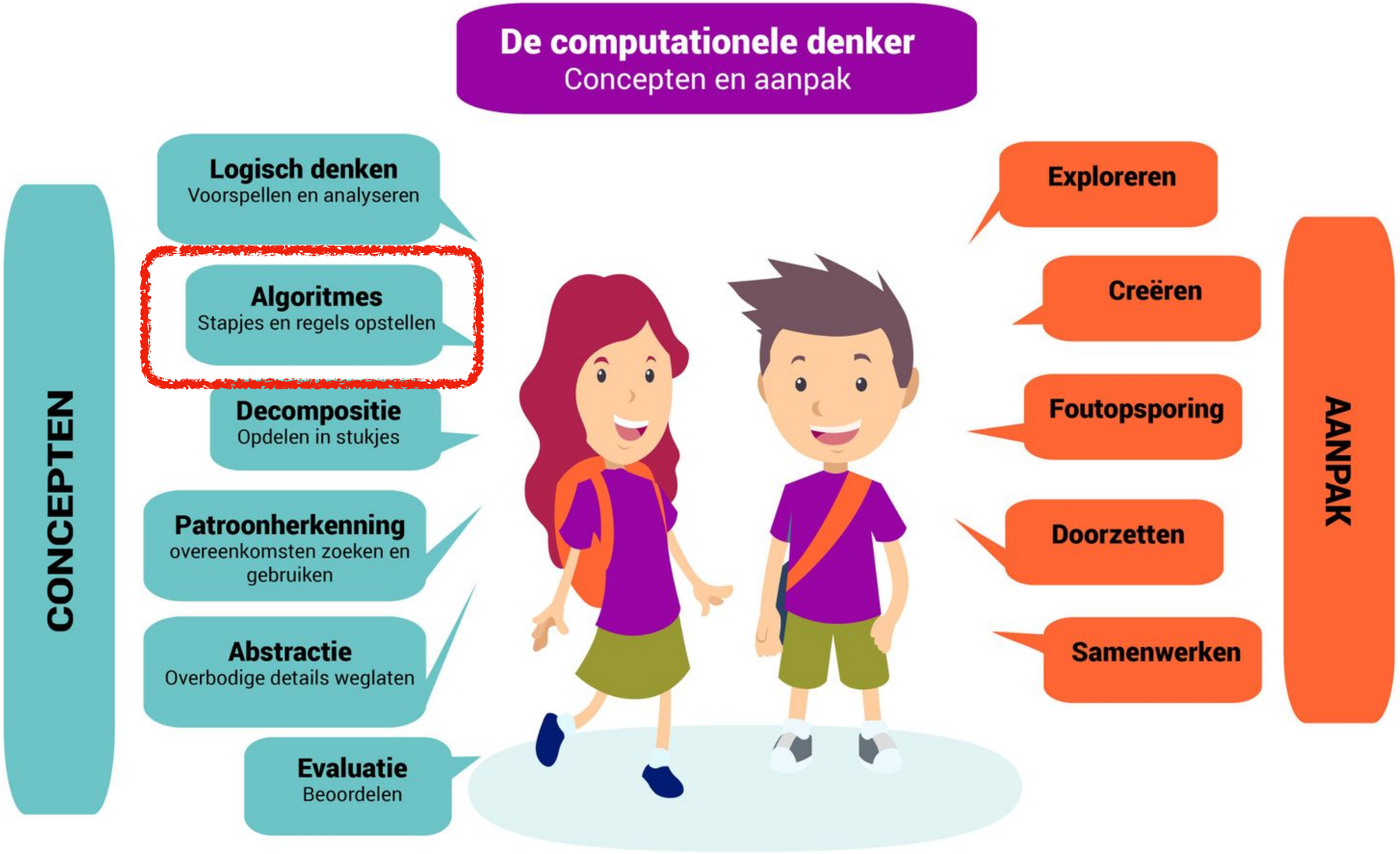
Decompositie

Decompositie houdt in dat we grotere problemen opsplitsen in kleinere delen, die we dan afzonderlijk kunnen behandelen.

Bijvoorbeeld als we een huis willen tekenen (zie fig. 5) doen we dit niet in één keer. We gaan verschillende delen proberen te herkennen, die we dan afzonderlijk kunnen tekenen. Dus voor het huis zouden we kunnen beginnen met de voorgevel, daarna de rechtergevel, dan het dak, etc.

Ook als we een eigen verhaal/mop/film willen visualiseren (zie fig. 6), moet er nagedacht worden in welke scènes we dit opsplitsen. Vervolgens kunnen we iedere scène apart uitwerken.

(Figuur 6: Een eigen verhaal maken bestaande uit verschillende scènes)



https://pro.katholiekonderwijs.vlaanderen/computationeel-denken

Wat is een algoritme?

Een *algoritme* is een reeks van instructies die één voor één uitgevoerd kunnen worden door een computer of door een mens.

Denk bijvoorbeeld aan een kookboek. Weet jij hoe je een cake kan bakken? Een kookboek legt dit stop voor stap uit. Wanneer je al die stappen goed uitvoert dan zal je op het einde van een lekkere cake kunnen smullen.

Maar niet alleen een kookboek maakt gebruik van algoritmen. Wanneer je een nieuw dans leert, moet je stap voor stap alle bewegingen leren. Wanneer je al die verschillende pasjes en bewegingen in de juiste volgorde uitvoert, dan krijg jij een mooie dansvoorstelling. Het is nu net precies die opeenvolging van stappen wat wij een algoritme noemen.

Voor een computer werkt dit net zo. Door te programmeren geven we de computer allemaal instructies die één voor één moeten uitgevoerd worden. Het is dan ook erg belangrijk om te leren goede instructies te geven. Alleen zo kunnen we zeker zijn dat de computer precies doet wat we willen.

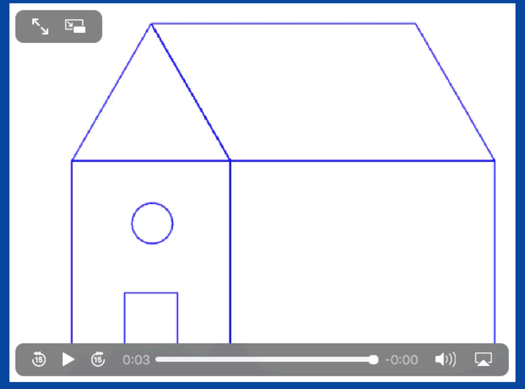
In Scratch nemen instructies de vormen van blokken aan. In de volgende delen, leer je meer over deze blokken en ondertussen maak je ook je eerste algoritme.

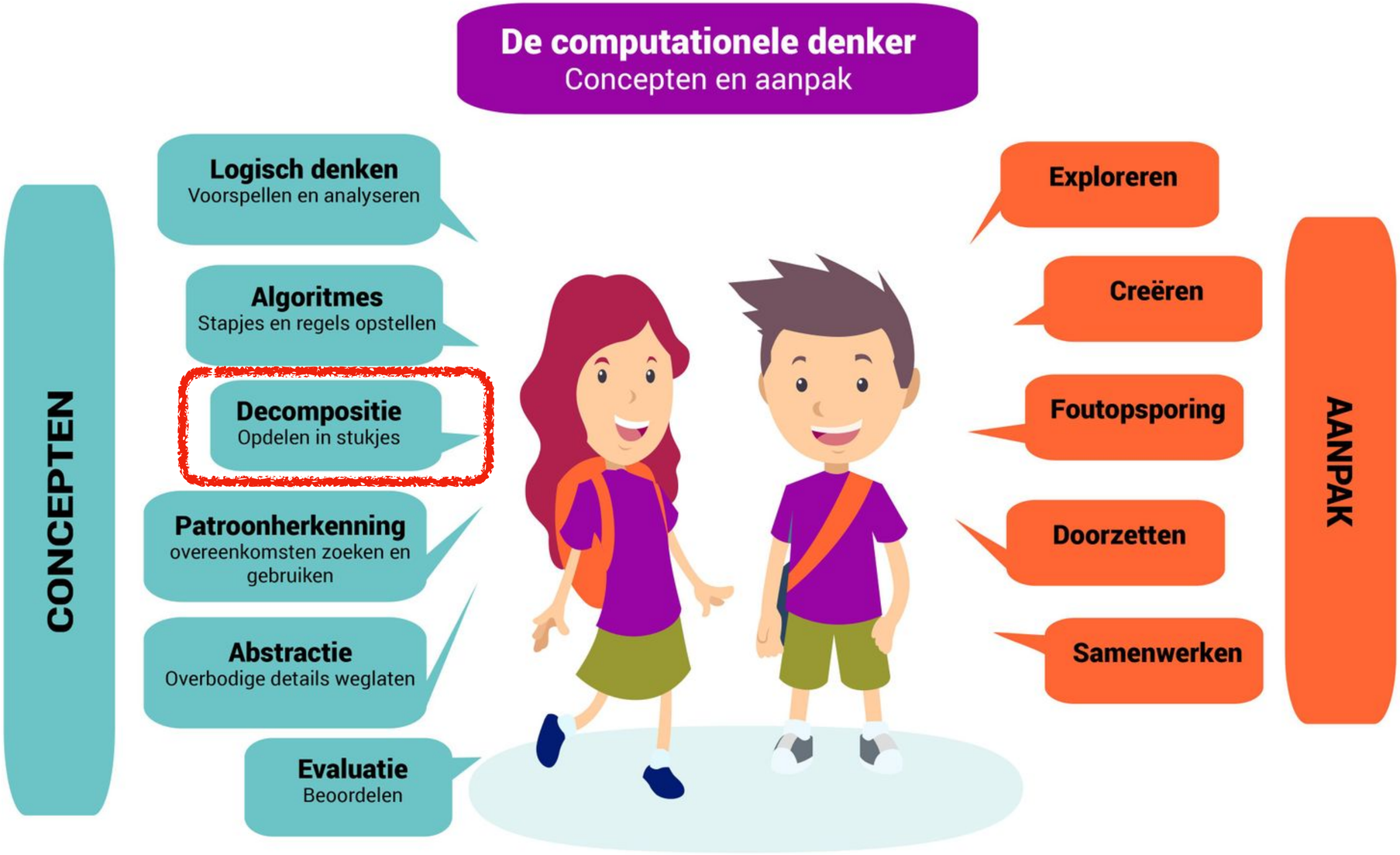
In oefening 3 maak je al een eerste eigen algoritme!

3. Wandelen in een vierkant:
    1. Laat Ruby eerst eens naar rechts bewegen en daarna naar beneden.
    2. Kan jij het vierkant nu verder afmaken?
    3. Wat moet je aanpassen als je Ruby het vierkant in een andere richting wil laten lopen? Bijvoorbeeld, eerst naar boven en dan naar links?

Oefening 4: Eigen huis

Het eindresultaat



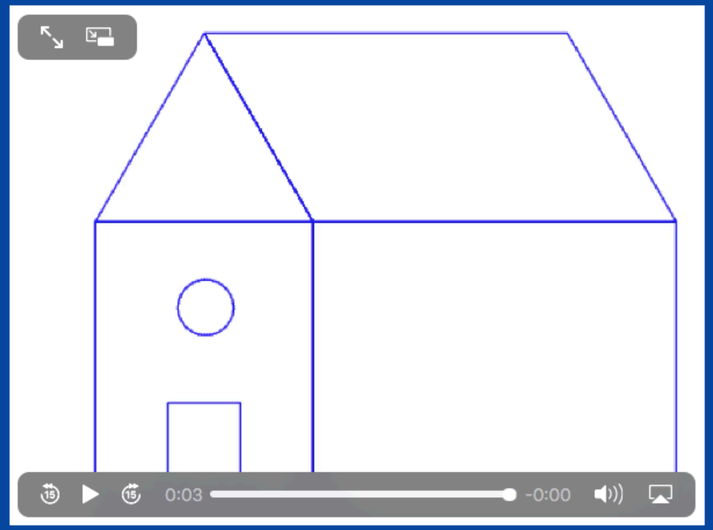


https://pro.katholiekonderwijs.vlaanderen/computationeel-denken

Decompositie

Oefening 4: Eigen huis

Het eindresultaat

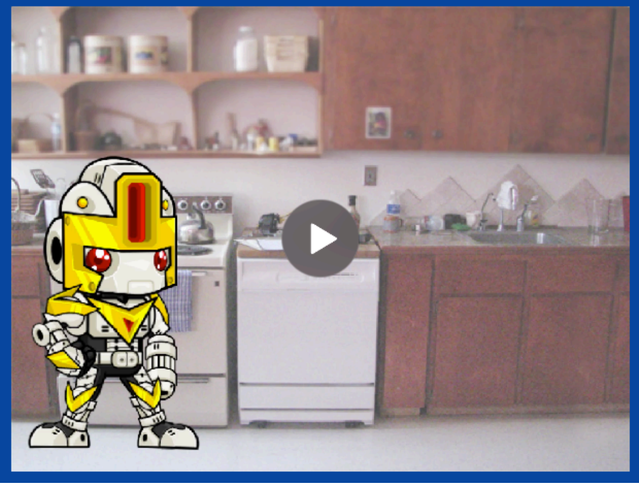


Oefening 4: Eigen verhaal

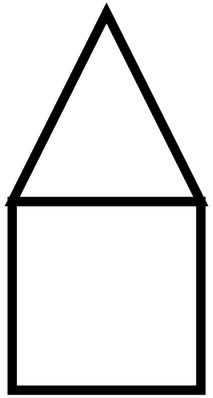
Tijd voor wat creativiteit. In deze oefening maak je je eigen verhaal! Wij hebben alvast een project aangemaakt met Ruby en Unix. Je kan dit project gebruiken of er zelf één maken.

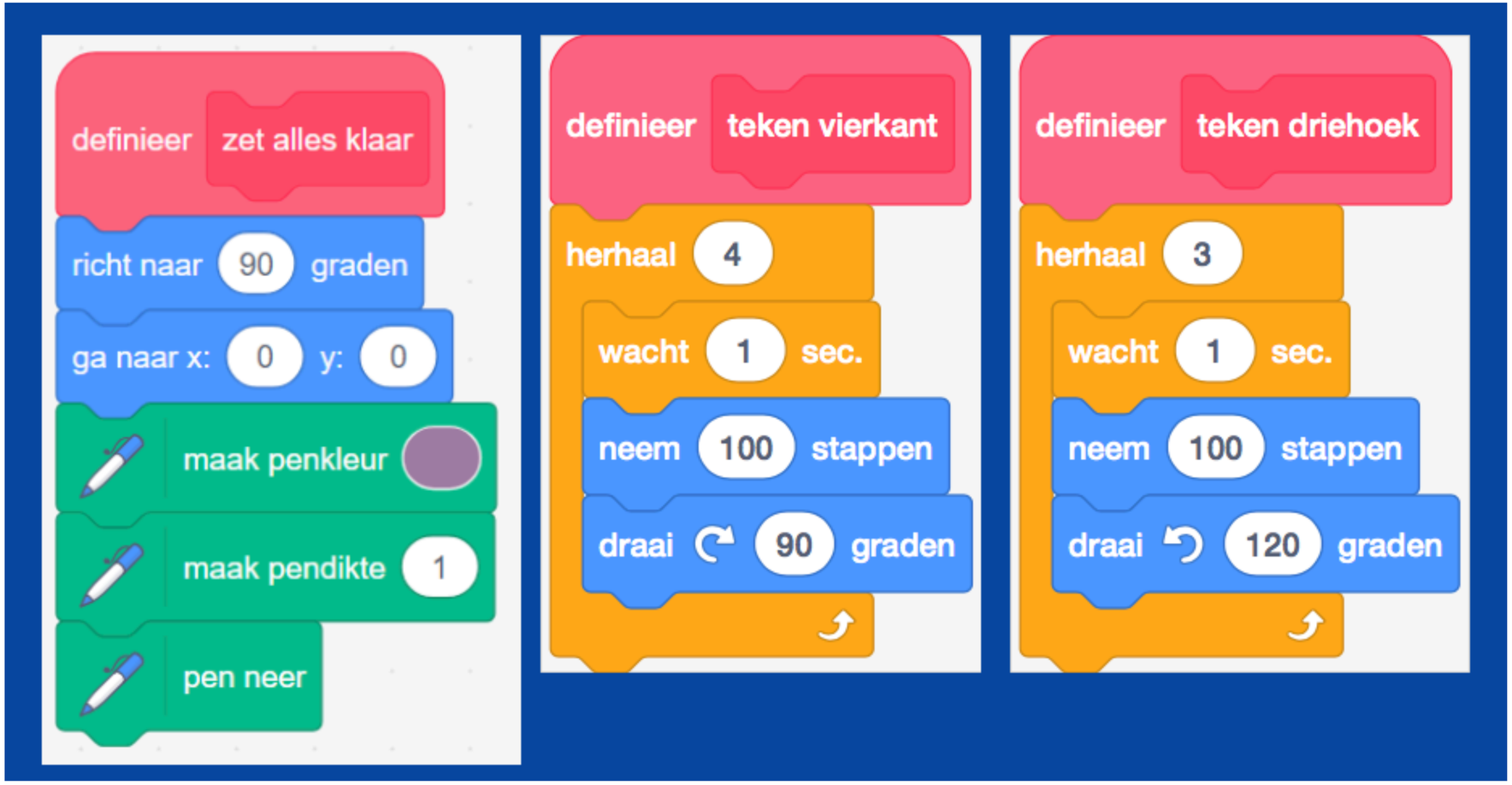
In onderstaande filmpje zie je een uitwerking van deze opdracht. Maar jij bepaalt jouw verhaal uiteraard zelf!

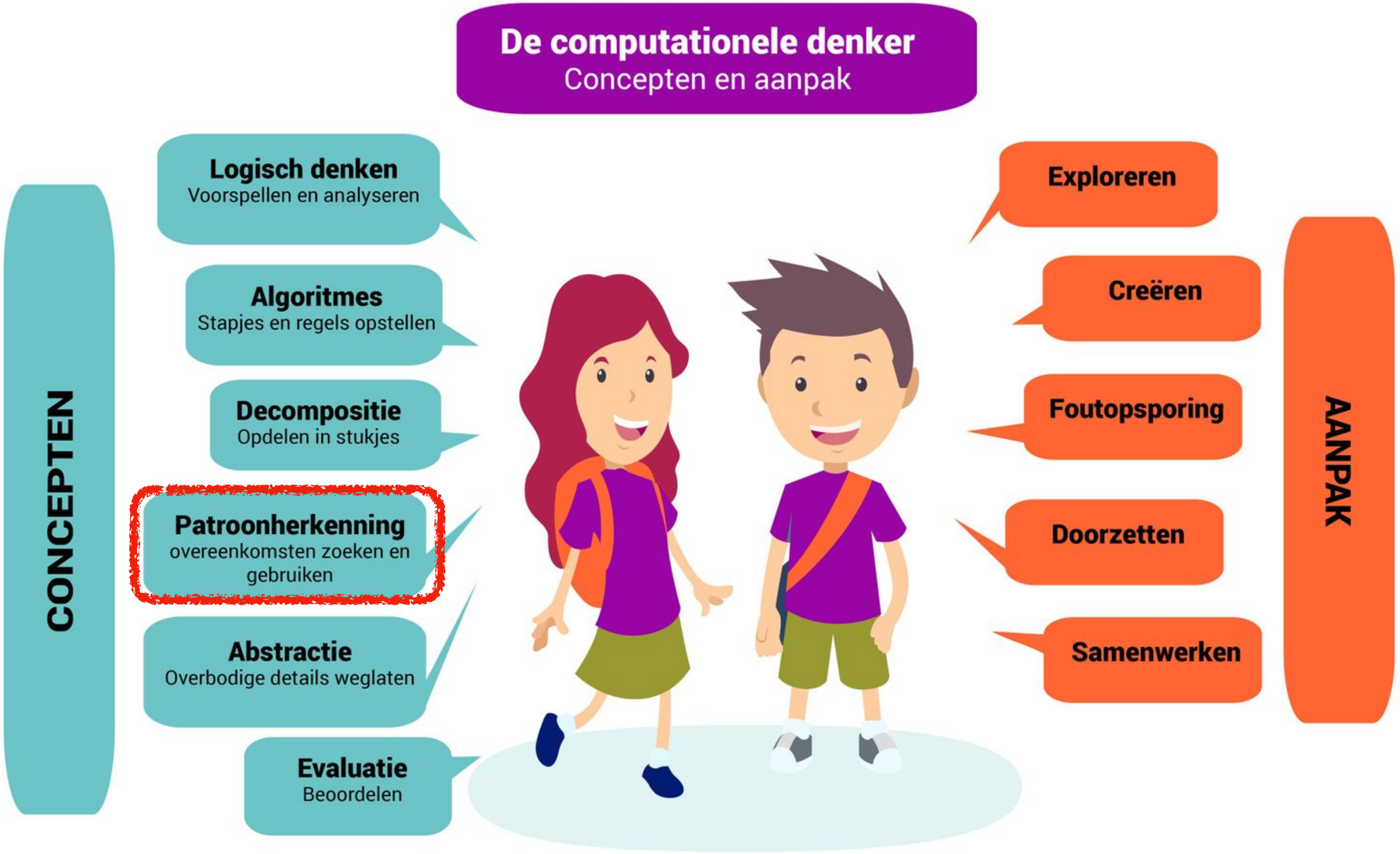
Het eindresultaat



Grotere programma's maak je in kleine stapjes. Dit wordt makkelijker als we eigen blokken maken.

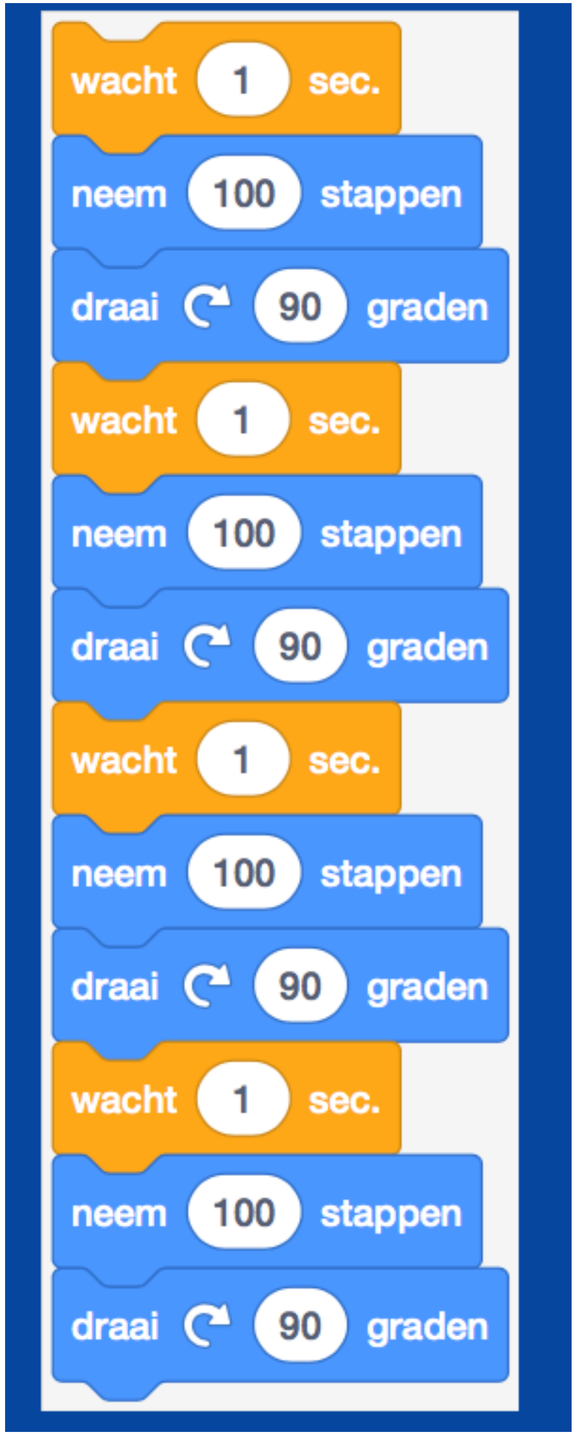






https://pro.katholiekonderwijs.vlaanderen/computationeel-denken

Patronen



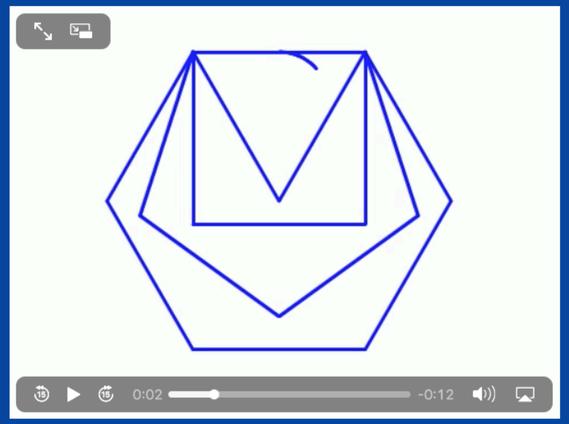
Teken een vierkant

Patronen

Oefening 3: Een aantal figuren tekenen

In deze oefening gaan we tekenen!

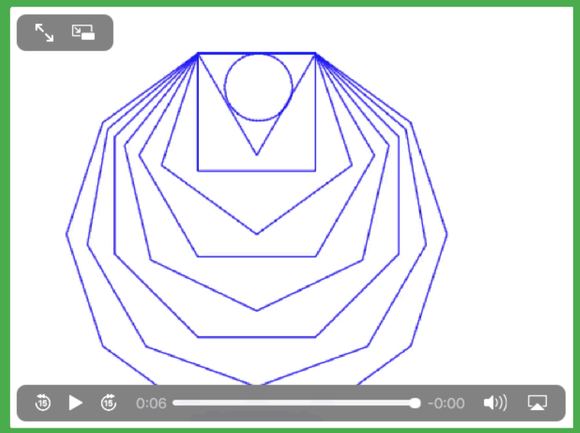
Het eindresultaat



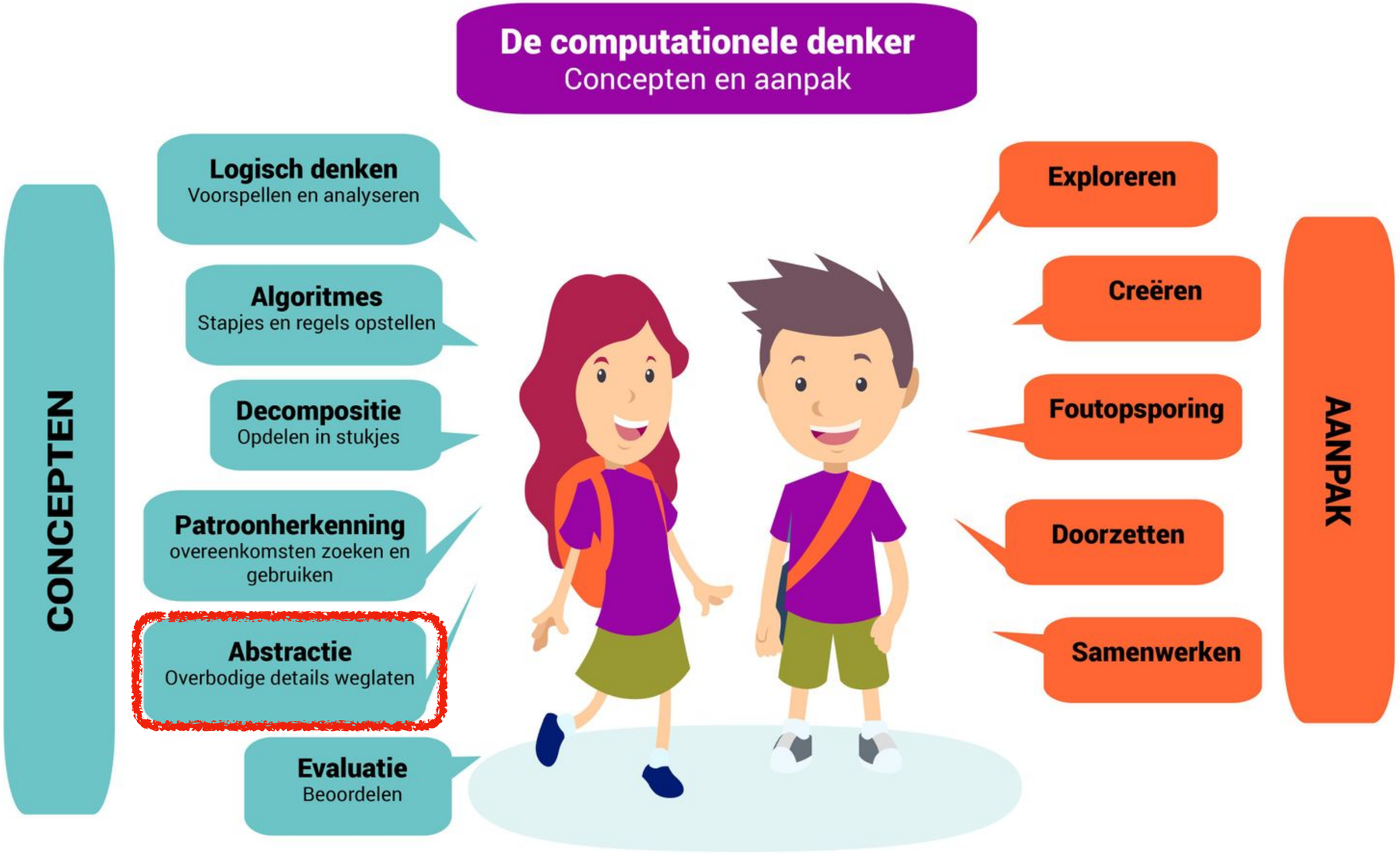
Oefening 6: Veelhoeken tekenen

In het vorig hoofdstuk heb je al eens een project gemaakt waarin je een aantal veelhoeken tekende. Gebruik je opgedane kennis om dit project wat overzichtelijker te maken.

Het eindresultaat



Wauw! Elke veelhoek kan eigenlijk op dezelfde manier getekend worden. Daarom heb ik maar één blokje nodig om alle veelhoeken te kunnen tekenen.



https://pro.katholiekonderwijs.vlaanderen/computationeel-denken

Abstractie

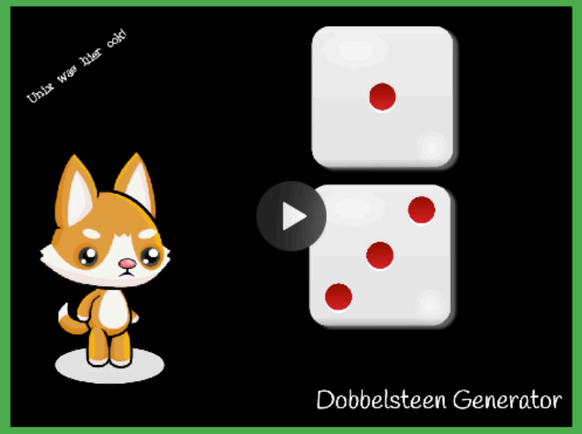
Wat is een variabele?

In een computerprogramma kun je een variabele gebruiken om een bepaalde waarde op te slaan. Dit kan tekst (bijvoorbeeld je voornaam) of een getal (bijvoorbeeld je leeftijd) zijn. Deze informatie wordt dan ergens in het computergeheugen opgeslagen zodat we die nadien terug kunnen raadplegen. Hiervoor moeten we aan deze informatie wel een unieke naam koppelen.

Oefening 2: Dobbelsteen generator

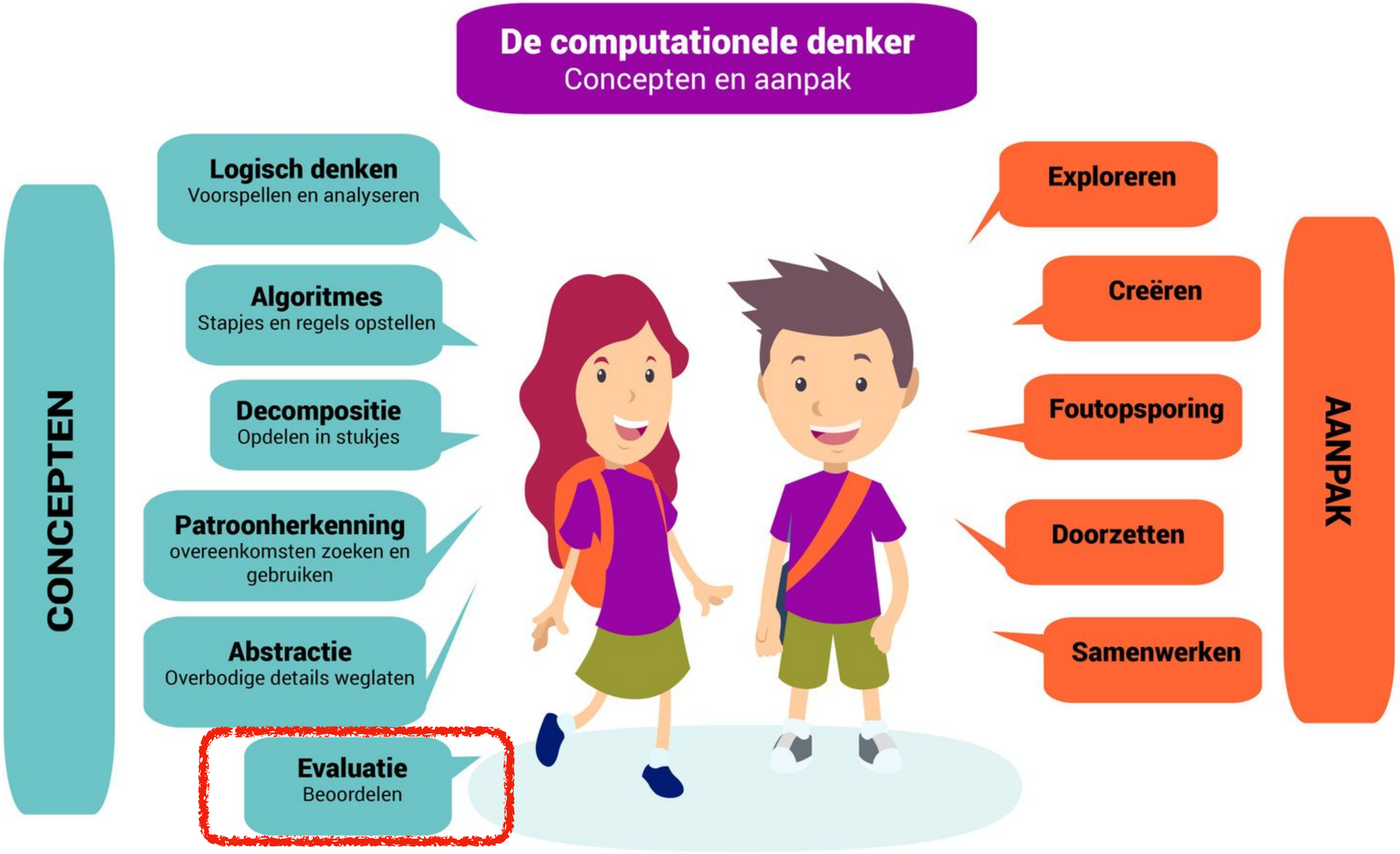
In deze oefening gooien we met twee dobbelstenen en berekenen de som van de ogen.

Het eindresultaat



Oefening 4: Trek een kaart

Het eindresultaat

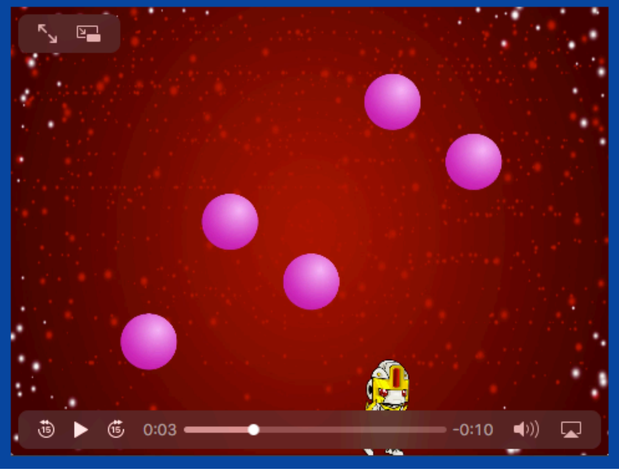


https://pro.katholiekonderwijs.vlaanderen/computationeel-denken

Oefening 3: Het regent gehaktballen

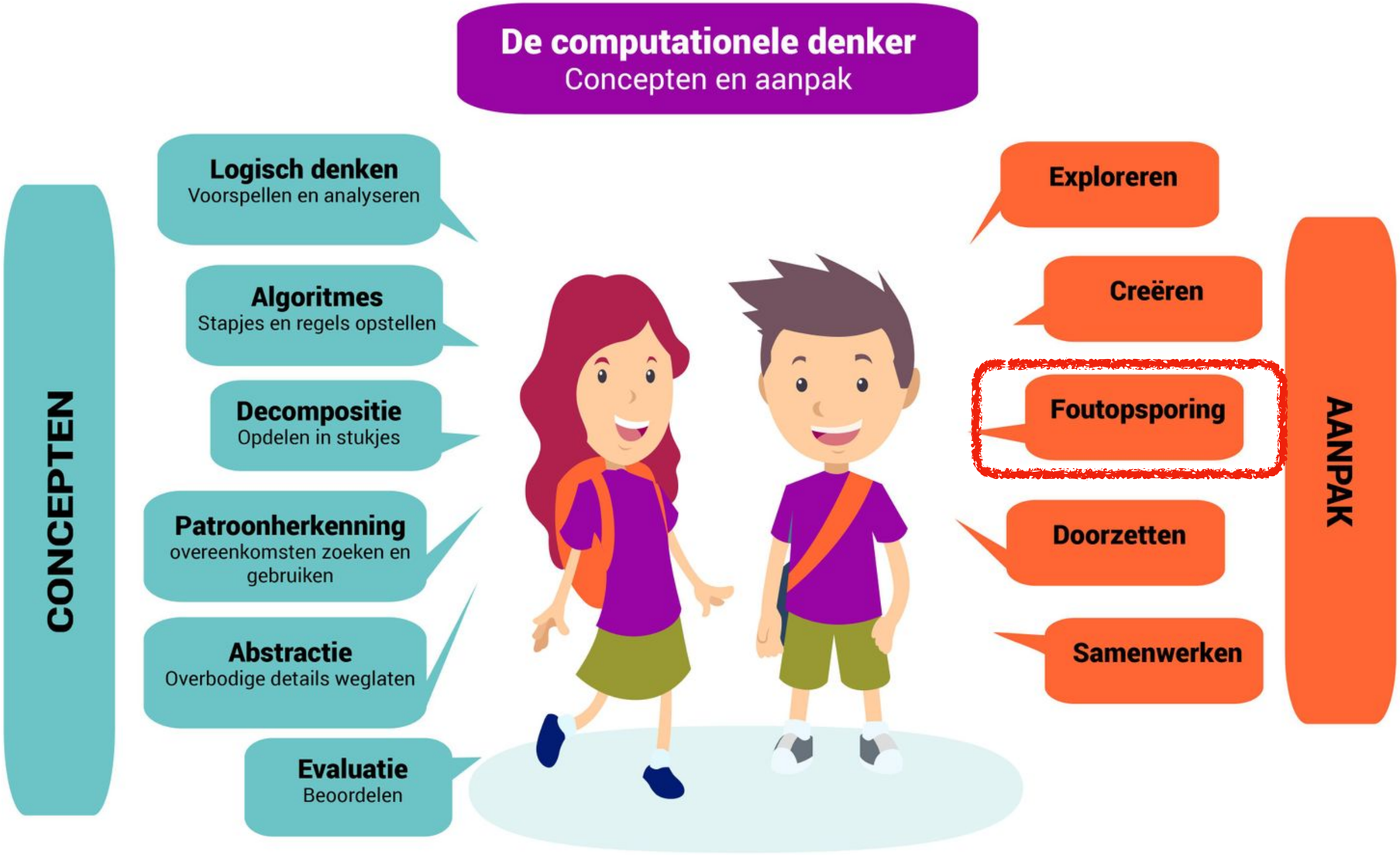
Heb je ooit de film "Het regent gehaktballen" gezien? We gaan ook zoiets maken. Ben je er klaar voor?

Het eindresultaat



Evaluatie

- Bewegen de ballen allemaal mooi naar beneden? Lijkt het alsof ze echt vallen?
- Is de snelheid van de ballen allemaal een beetje verschillend?
- Beweegt Unix naar rechts wanneer je het pijltje naar rechts induwt?
- Beweegt Unix naar links wanneer je het pijltje naar links induwt?
- Ziet jouw oplossing er ongeveer hetzelfde uit als in het filmpje?



https://pro.katholiekonderwijs.vlaanderen/computationeel-denken

Wist je dat debuggen een ander woord is voor het zoeken en verbeteren van fouten in een computerprogramma?

Oefening 5: Een slechte mop

In onderstaand filmpje vertellen Unix en Ruby elkaar wat moppen. Maar in het project dat je kan downloaden is er wat fout gelopen met de signalen. Kan jij vinden waar het fout loopt en het programma aanpassen?

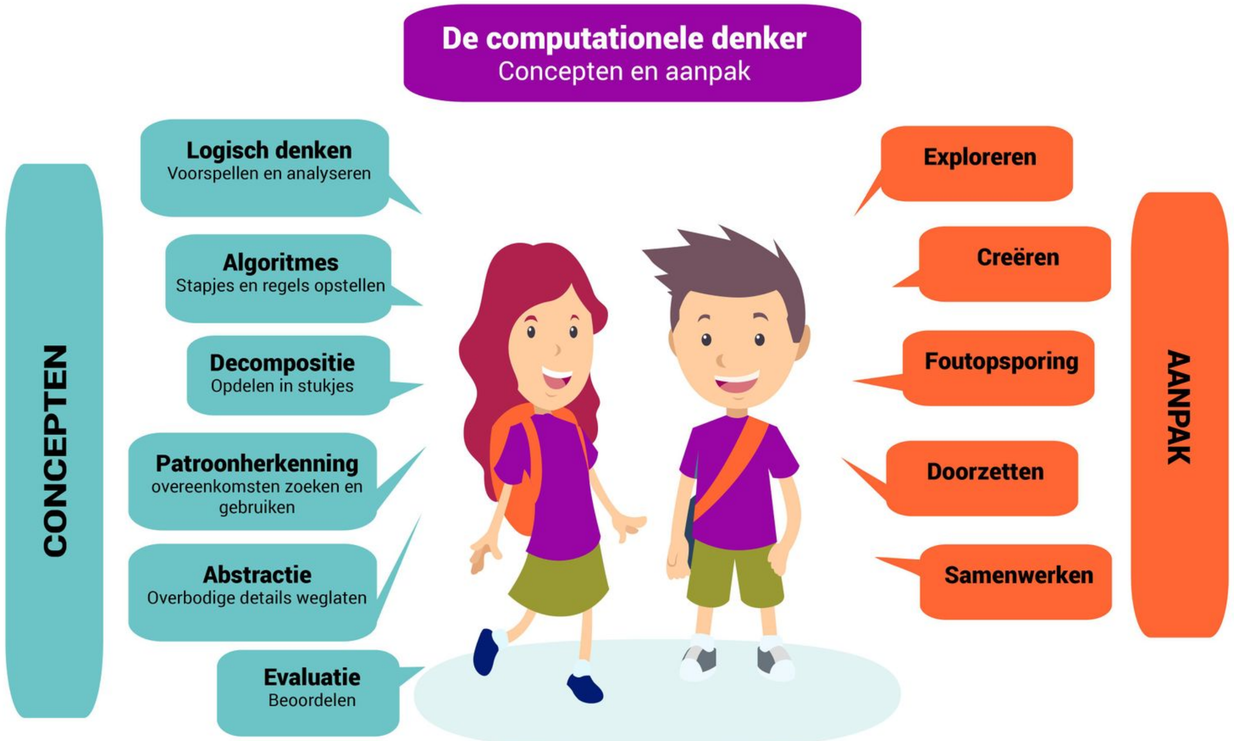
Juiste resultaat

Programmeerconcepten

- Sequentie
- Input & Output
- Variabelen
- Herhaling
- Beslissingen
- Eigen blokken

Samenvatting

Computationeel denken is problemen op zo een manier benaderen dat computers kunnen gebruikt worden om ze op te lossen.



Bestaat uit concepten en aanpakken

Wordt in praktijk gebracht via programmeren

Overzicht

1. Computationeel Denken (CD) uitgelegd
2. **Plaats van CD in het leerplichtonderwijs**
3. Relevantie van CD

lichamelijk, geestelijk en emotionele gezondheid

Nederlands

andere talen

digitale competentie en mediawijsheid

sociaal-relationele competenties

wiskunde, exacte wetenschappen en technologie

burgerschap

historisch bewustzijn

ruimtelijk bewustzijn

duurzaamheid

financiële competenties

juridische competenties

innovatiedenken, creativiteit, probleemoplossend en kritisch denken

zelfbewustzijn en -expressie, zelfsturing en wendbaarheid

ontwikkeling van initiatief, ambitie, ondernemingszin en loopbaancompetenties

cultureel bewustzijn en culturele expressie

Nieuwe eindtermen 1ste graad

- **4. Digitale competentie en mediawijsheid**
  - **4.1& 4.2:** Digitale media en *toepassingen gebruiken* om te creëren, te participeren en te interageren. — Applications
  - **4.3 & 4.4:** *Computationeel denken en handelen* — Foundations
  - **4.5-4.7:** Verantwoord, kritisch en ethisch omgaan met digitale en niet-digitale media en informatie. — Implications

Eindtermen 1ste graad

- **4. Digitale competentie en mediawijsheid**
  - **4.1& 4.2:** Digitale media en *toepassingen gebruiken* om te creëren, te participeren en te interageren.

    Applications
  - **4.3 & 4.4:** *Computationeel denken en handelen*

    Foundations
    - Bouwstenen van digitale systemen
    - Computationeel denken ***en*** programmeren

    Informaticawetenschappen
  - **4.5-4.7:** Verantwoord, kritisch en ethisch omgaan met digitale en niet-digitale media en informatie.

    Implications

Informaticawetenschappen

- Wetenschap die de digitale wereld verklaart
- Fundamenten: concepten met een lange houdbaarheidsdatum
- Wat je nodig hebt om als een informaticawetenschapper te denken en handelen
- Omvat computationeel denken

Computationeel denken en handelen.

4.3 De leerlingen onderscheiden bouwstenen van digitale systemen. (transversaal)

Met inbegrip van kennis
* Conceptuele kennis
- Bouwstenen van een digitaal systeem
> Input verwerking output
> Binair
> Hardware
> Digitale media zoals dataformaat
> Digitale toepassingen zoals tekstverwerking, multi-media verwerking, game
> Besturingssysteem
- Informatieverwerkende systemen en communicatie tussen deze systemen

1

Met inbegrip van dimensies eindterm
Cognitieve dimensie: beheersingsniveau begrijpen

4.4 De leerlingen passen een eenvoudig zelf ontworpen algoritme toe om een probleem digitaal en niet-digitaal op te lossen. (transversaal)

Met inbegrip van kennis
* Conceptuele kennis
- Concepten van computationeel denken: decompositie, patroonherkenning, abstractie, algoritmen
- Organisatie, modellering, simulatie en digitale representatie van informatie
- Debuggen (testen en bijsturen)
- Principes van programmeertalen: sequentie, herhalingsstructuur, keuzestructuur

2

3

Informaticawetenschappen

Overzicht

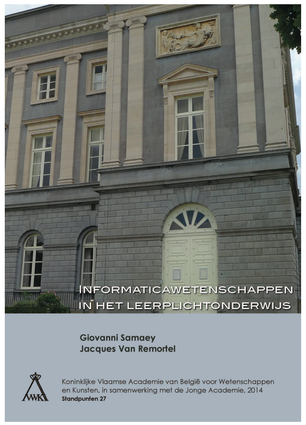
1. Computationeel Denken (CD) uitgelegd
2. Plaats van CD in het leerplichtonderwijs
3. **Relevantie van CD**

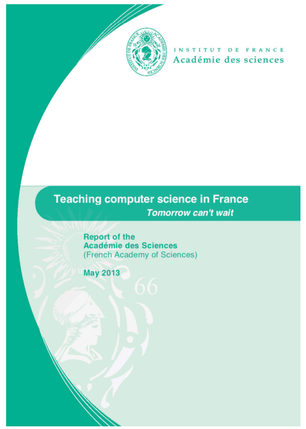
FORUM VOOR INFORMATICA WETENSCHAPPEN

{SI}² Sciences Informatiques pour le Secondaire Inférieur

2LINK²

De rol van *informaticawetenschappen* in het leerplichtonderwijs

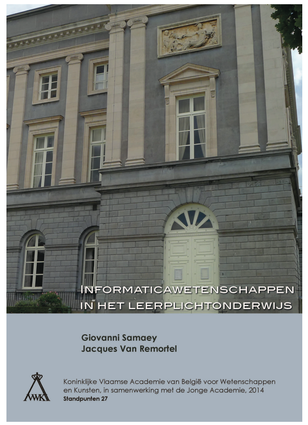




K12 COMPUTER SCIENCE FRAMEWORK

Informaticawetenschappen

Visie



- Basiskennis voor alle leerlingen
  - Waarom?
    - Informaticawetenschappen verklaren de digitale wereld
    - Informaticawetenschappen ondersteunen informaticavaardigheid
- Meer gespecialiseerde kennis voor deel van de leerlingen

Conclusie

- *Computationeel denken*
  - bestaat uit een aantal concepten en aanpakken
  - kan in praktijk gebracht worden via programmeren
- *Informaticawetenschappen*
  - omvat Computationeel Denken
  - verklaren de digitale wereld

- Nieuwe eindtermen bekijken in de context van informaticawetenschappen

Vragen?

- http://leer-scratch.be
- https://www.leer-scratch.be/info/computationeeldenken/
- https://www.program-uurtje.org
- https://www.progra-meer.org
- https://www.uhasselt.be/tutorials-python
- http://CDmicrobit.gitlab.io/
- http://www.csunplugged.nl
- https://computationeeldenken.org

Links

- http://leer-scratch.be
- https://www.leer-scratch.be/info/computationeeldenken/
- https://www.program-uurtje.org
- https://www.progra-meer.org
- https://www.uhasselt.be/tutorials-python
- http://CDmicrobit.gitlab.io/
- http://www.csunplugged.nl