

Computationeel denken concreet in de klas

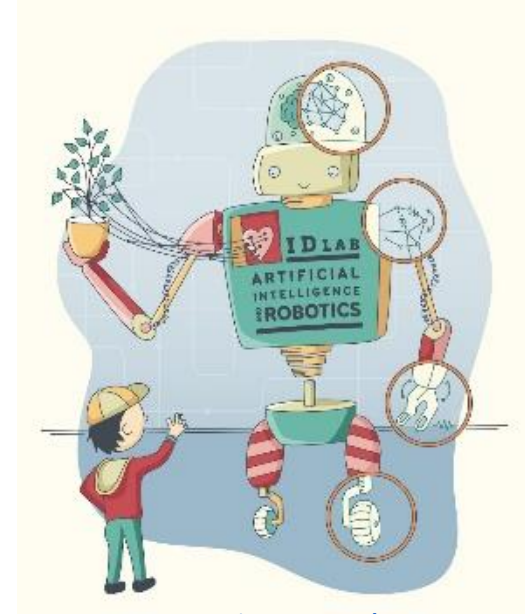
21 september 2022 - Ontmoetingsdag InnoVET



Natacha Gesquière



Francis wyffels



airo.ugent.be



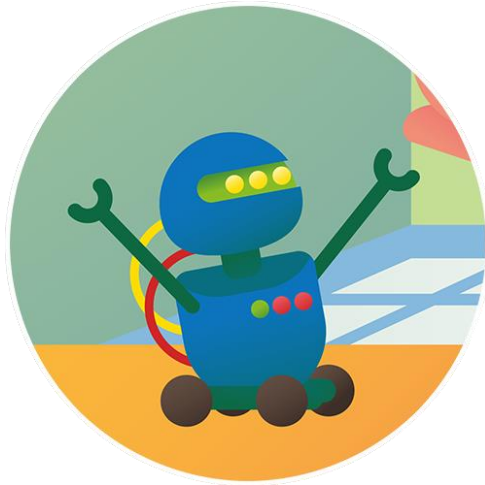
Bjarne Van de Velde



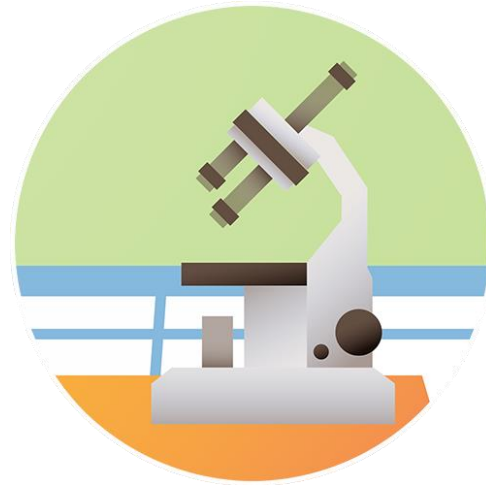
Tom Neutens



Lesmaterialen



Innovatief



Onderzoeksgedreven



Inclusief



Maatschappelijk relevant



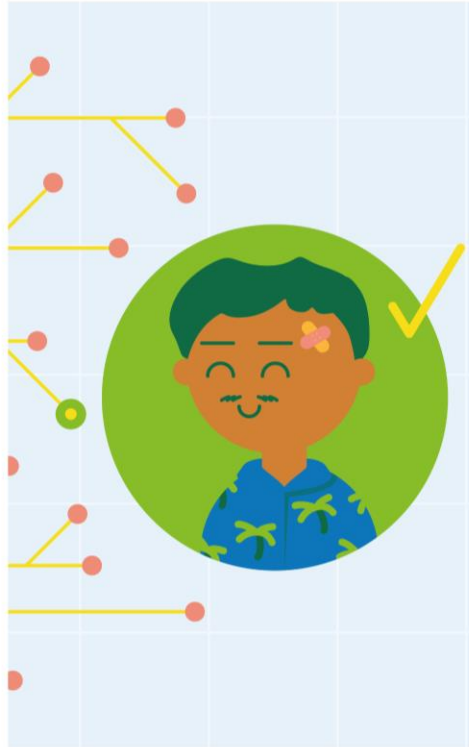
AI in de maatschappij

Lesmaterialen – leerlijn rond AI

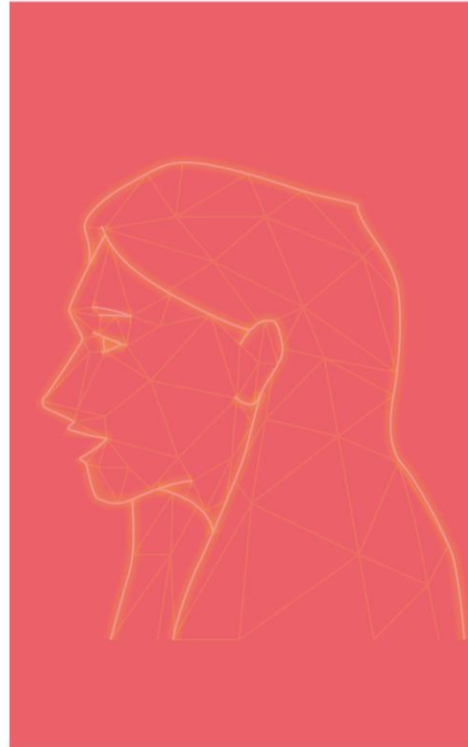
SOCIALE ROBOT



AI IN DE ZORG



CHATBOT



AI IN DE KUNST



KIKS



Lesmaterialen – leerpaden voor leerkrachten

KIKS

[Lees meer >](#)



Leerlingen uit de derde graad (SO) onderzoeken hoe planten zich via hun huidmondjes aanpassen aan de klimaatverandering. Daarvoor tellen ze deze huidmondjes met artificiële intelligentie en beeldherkenning.

AI in de Kunst

[Lees meer >](#)



Kunnen we kunst maken met artificiële intelligentie? Leerlingen uit de tweede en derde graad (SO) leven zich creatief uit met AI en reflecteren over het resultaat. Is dit kunst? Ook ontdekken ze hoe AI ons cultureel erfgoed kan beschermen.

Python in de Wiskundeles

[Lees meer >](#)



De programmeertaal Python biedt toffe mogelijkheden om de wiskundeles te verrijken. Van de stelling van Pythagoras tot de opmaak van eigen grafieken, digitale beeldverwerking en lineaire regressie, alles wordt duidelijker met Python.

AI in de Zorg

[Lees meer >](#)



Ziekenhuizen maken vandaag al gebruik van artificiële intelligentie. Leerlingen uit de tweede en derde graad (SO) ontdekken welke systemen er bestaan en hoe ze dokters helpen om beslissingen te nemen. Zo leren leerlingen de principes van de beslissingsboom, een veelgebruikte techniek in machine learning.

Sociale robot

[Lees meer >](#)



Terwijl leerlingen uit de eerste graad (SO) een sociale robot knutselen en programmeren, leren ze complexe problemen oplossen via computationeel denken. Spelenderwijs werken ze aan de nieuwe eindtermen rond digitale competenties.

WeGoSTEM

[Lees meer >](#)



We dagen kinderen van de derde graad (BO) uit om een kunstrobot die kan tekenen te programmeren. Spelenderwijs leren de kinderen heel wat STEM-vaardigheden, van techniek tot computationeel denken.

Programmeren met Python

[Lees meer >](#)



De basisprincipes van programmeren aanleren? Dat kan perfect via dit pakket. We gaan aan de slag met Python en leren alles over sequenties, herhalingsstructuur en keuzestructuur. Dat is precies wat er in de eindtermen staat. En meer!

Chatbot

[Lees meer >](#)



Waar taal en technologie samenkomen, ontstaat het domein van Natural Language Processing. Kan een computer teksten begrijpen, vertalen of zelfs schrijven? Kan een computer emoties herkennen? Leerlingen uit de tweede en derde graad (SO) leren er alles over in dit pakket.

AI in de Landbouw

[Lees meer >](#)



Rotte tomaten weghalen tijdens de oogst? Daar kan artificiële intelligentie bij helpen. Maar hoe? Leerlingen uit de tweede en derde graad (SO) gaan met AI aan de slag. Misschien kan een trillende lopende band het systeem nog beter maken?

Computationeel denken

[Lees meer >](#)



Hoe kunnen we complexe problemen oplossen met behulp van een computer? Dankzij computationeel denken! Dat kan je leren via allerlei activiteiten met óf zonder computer. Wij helpen je alvast op weg.

Python in STEM

[Lees meer >](#)



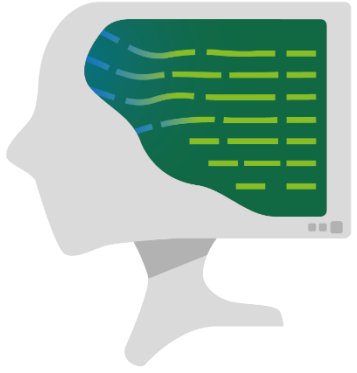
In dit pakket leer je complexe problemen eenvoudiger én sneller aanpakken met de programmeertaal Python. Programmeren kan immers een verbindende rol spelen tussen wetenschap, techniek, ontwerp en toegepaste wiskunde. Kortom, dankzij Python halen we het beste uit STEM.

Physical computing

[Lees meer >](#)



Een muziekinstrument, auto of weerstation bouwen? Dat kan met Dwenguino, een microcontrollerplatform met een eigen grafische programmeeromgeving. Kinderen uit de derde graad (BO) of de eerste graad (SO) kunnen er meteen mee aan de slag. In het echt of in onze simulator.



Computationeel denken helpt om de gedigitaliseerde wereld beter te begrijpen.



Wat is computationeel denken?

Computationeel denken is een methodologie om een complex probleem zo aan te pakken dat je een computer zou kunnen inzetten om het op te lossen.



Het begrip 'computer' wordt gebruikt in de ruime zin van het woord: het betreft desktops, laptops, tablets, smartphones en andere informatieverwerkende systemen.

Computationeel leren denken kan zowel a.d.h.v. unplugged als plugged voorbeelden en activiteiten gebeuren.

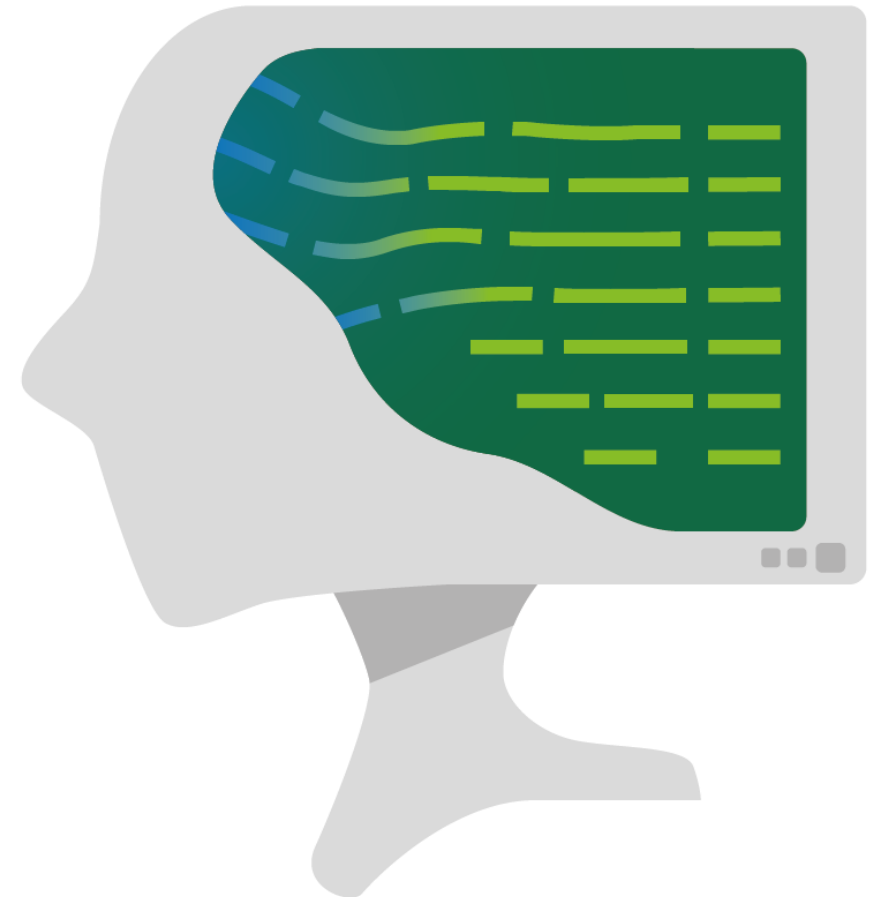
Voor sommige problemen kan je via deze methodologie gemakkelijker een oplossing voor het probleem vinden, of kan je gemakkelijker de computer inzetten om het probleem op te lossen.



Computationeel denken is niet nieuw!

Het komt bijvoorbeeld voor als

- er een lampje brandt op je dashboard;
- je een taart bakt volgens een recept;
- een pianist een muziekstuk inoefent;
- je een gps gebruikt.



De computationele denker

CONCEPTEN

Decompositie

Patroonherkenning

Abstractie

Algoritme

Logica

Evalueren



Exploreren

Creëren

Samenwerken

Debuggen

Doorzetten

Creativiteit

AANPAK

De computationele denker

CONCEPTEN

Decompositie

Patroonherkenning

Abstractie

Algoritme

Logica

Evalueren



Exploreren

Creëren

Samenwerken

Debuggen

Doorzetten

Creativiteit

AANPAK

Problemen oplossen - Ruimer dan basisconcepten

We willen eigen data visualiseren en analyseren.
Is er een trend die zich openbaart in deze data?

Verkennen van het probleem. Wat heb ik nodig?

Subtaken (**decompositie**):

1. Data verzamelen, bv. door meting (Wat meten? Hoe meten? Toestel?)
2. Data visualiseren (Hoe visualiseren? Bv. **puntenwolk en trendlijn.**)
3. Welke software? (Bv. Geogebra, Excel, Graphmatica, **Python.**)
4. Data aanbieden aan de computer. Wat is geschikt formaat? **csv**



Trendlijn?

Kijk naar de vorm van het spreidingsdiagram.
Ellipsvormig? Andere vorm?

De vorm van de puntenwolk bepaalt welke vorm er geschikt is voor de trendlijn. (**patroonherkenning**)



De trendlijn is een benadering (model) van de werkelijkheid, die toelaat voorspellingen te doen voor de evolutie in de toekomst. (**abstractie**)



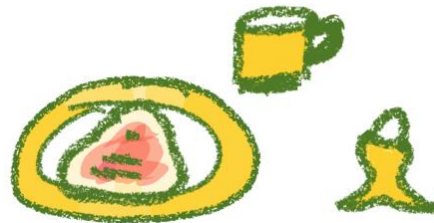
Een **algoritme** om in Python de trendlijnen te visualiseren (a.d.h.v. Python-modules)

- Definieer gedaante van vergelijking rechte.
- Definieer functie om coëfficiënten trendlijn te bepalen passend bij gegeven datapunten.
- Lees data in (**csv-bestand**).
- Bepaal a.d.h.v. data en bovenstaande definities de trendlijn.
- Visualiseer data en trendlijn (tekenvenster - puntenwolk - trendlijn).



Decompositie

Ontbijt maken



Decompositie

Ontbijt maken



Boterham met confituur
Zacht gekookt ei
Kopje thee met suiker

Een probleem doordacht opsplitsen in deelproblemen, zodat elk deel afzonderlijk kan worden aangepakt.

De opsplitsing moet goed gekozen zijn, zodat het oplossen van het probleem gemakkelijker bekomen wordt.

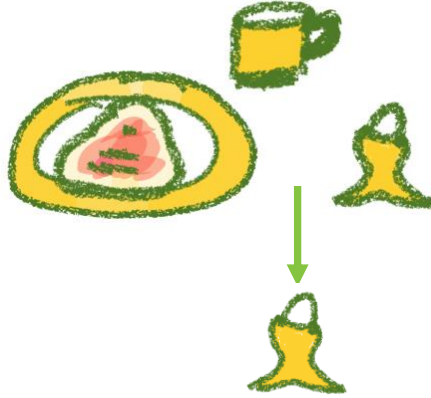
Decompositie

Een probleem doordacht opsplitsen in deelproblemen, zodat elk deel afzonderlijk kan worden aangepakt.



Boterham smeren

Ontbijt maken



Ei koken

De opsplitsing moet goed gekozen zijn, zodat het oplossen van het probleem gemakkelijker bekomen wordt.



Thee zetten

Decompositie

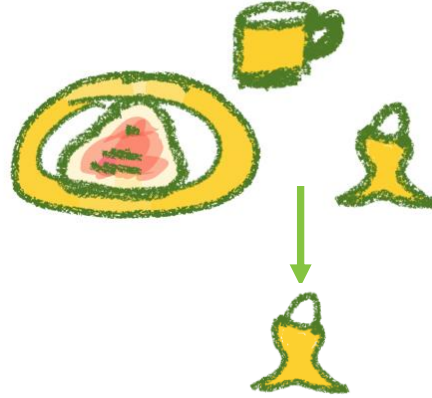
Een probleem doordacht opsplitsen in deelproblemen, zodat elk deel afzonderlijk kan worden aangepakt.



Boterham smeren



Ontbijt maken



Ei koken



De opsplitsing moet goed gekozen zijn, zodat het oplossen van het probleem gemakkelijker becomes wordt.



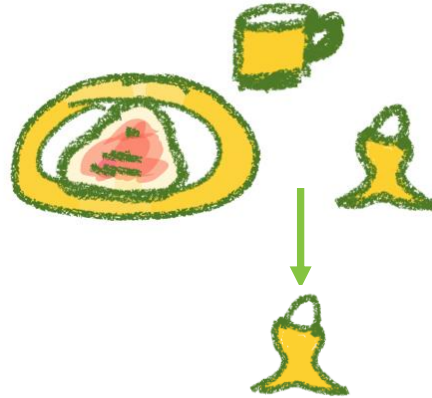
Thee zetten



Decompositie

Een probleem doordacht opsplitsen in deelproblemen, zodat elk deel afzonderlijk kan worden aangepakt.

Ontbijt maken



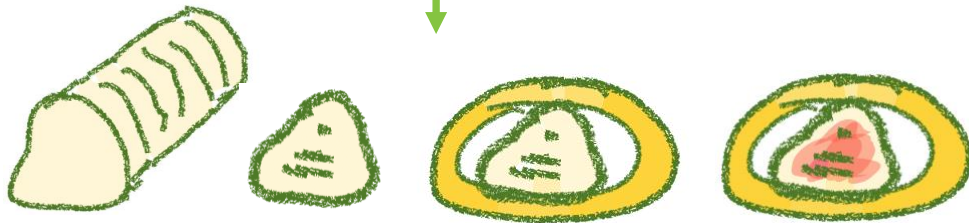
De opsplitsing moet goed gekozen zijn, zodat het oplossen van het probleem gemakkelijker bekomen wordt.



Boterham smeren

Ei koken

Thee zetten



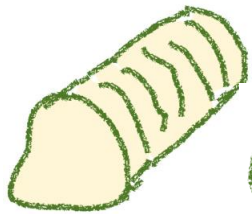
Brood nemen Boterham nemen Boterham op bord leggen Confituur smeren

Decompositie

Een probleem doordacht opsplitsen in deelproblemen, zodat elk deel afzonderlijk kan worden aangepakt.

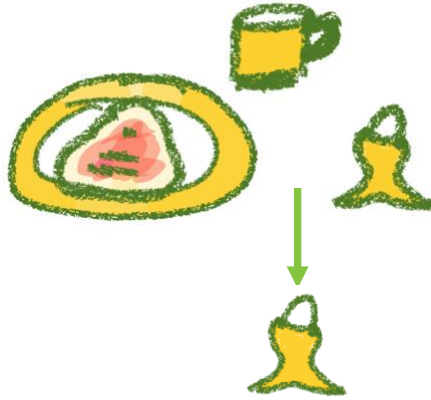


Boterham smeren



Brood nemen Boterham nemen Boterham op bord leggen Confituur smeren

Ontbijt maken



Ei koken



Water koken Ei erin dopje

De opsplitsing moet goed gekozen zijn, zodat ze het oplossen van het probleem gemakkelijker maakt.

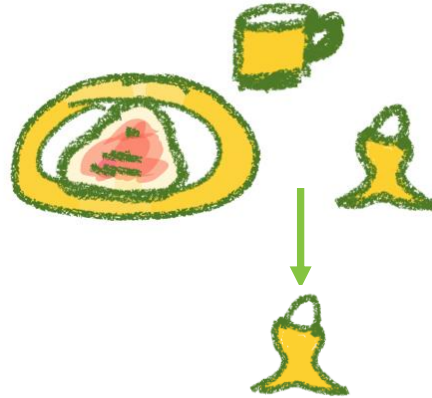


Thee zetten

Decompositie

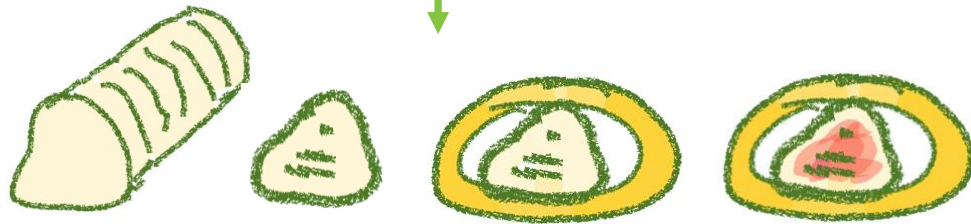
Een probleem doordacht opsplitsen in deelproblemen, zodat elk deel afzonderlijk kan worden aangepakt.

Ontbijt maken



De opsplitsing moet goed gekozen zijn, zodat het oplossen van het probleem gemakkelijker bekomen wordt.

Boterham smeren



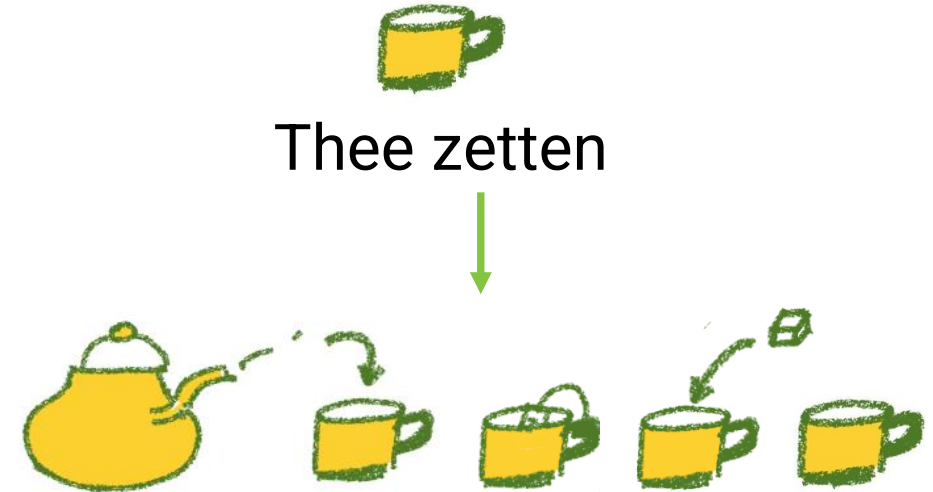
Brood nemen Boterham nemen Boterham op bord leggen Confituur smeren

Ei koken



Water koken Ei erin Ei in dopje

Thee zetten



Water koken Water in kopje Theezakje erin Klontje suiker erin

Patroonherkenning

Achterhalen dat bepaalde aspecten van een probleem gelijkenissen vertonen, waardoor het vereenvoudigd kan worden; maar ook herkennen dat een probleem gelijkenissen vertoont met een eerder opgelost probleem.

Patroonherkenning

Achterhalen dat bepaalde aspecten van een probleem gelijkenissen vertonen, waardoor het vereenvoudigd kan worden; maar ook herkennen dat een probleem gelijkenissen vertoont met een eerder opgelost probleem.



Patroonherkenning

Achterhalen dat bepaalde aspecten van een probleem gelijkenissen vertonen, waardoor het vereenvoudigd kan worden; maar ook herkennen dat een probleem gelijkenissen vertoont met een eerder opgelost probleem.



Patroonherkenning

Achterhalen dat bepaalde aspecten van een probleem gelijkenissen vertonen, waardoor het vereenvoudigd kan worden; maar ook herkennen dat een probleem gelijkenissen vertoont met een eerder opgelost probleem.



Patroonherkenning

Achterhalen dat bepaalde aspecten van een probleem gelijkenissen vertonen, waardoor het vereenvoudigd kan worden; maar ook herkennen dat een probleem gelijkenissen vertoont met een eerder opgelost probleem.



Patroonherkenning

Achterhalen dat bepaalde aspecten van een probleem gelijkenissen vertonen, waardoor het vereenvoudigd kan worden; maar ook herkennen dat een probleem gelijkenissen vertoont met een eerder opgelost probleem.

Ontbijt maken

Boterham met choco
Kopje koffie met wolkje melk

Algoritme

Expliciteren van een reeks eenduidige instructies die stapsgewijs moeten worden uitgevoerd. Inzien dat deze reeks van instructies en de volgorde ervan essentieel zijn om het gewenste resultaat op te leveren.

Algoritme

Expliciteren van een reeks eenduidige instructies die stapsgewijs moeten worden uitgevoerd. Inzien dat deze reeks van instructies en de volgorde ervan essentieel zijn om het gewenste resultaat op te leveren.

Boterham smeren

Algoritme

Expliciteren van een reeks eenduidige instructies die stapsgewijs moeten worden uitgevoerd. Inzien dat deze reeks van instructies en de volgorde ervan essentieel zijn om het gewenste resultaat op te leveren.

Boterham smeren



Brood
nemen

Boterham
nemen

Boterham
op bord leggen

Confituur
smeren

Abstractie

Het negeren van informatie of het verbergen van bepaalde details. Bij het aanpakken van complexe problemen is er nood aan abstractie omdat het redeneren vlotter verloopt zonder de ballast van irrelevante details.



Abstractie

Het negeren van informatie of het verbergen van bepaalde details. Bij het aanpakken van complexe problemen is er nood aan abstractie omdat het redeneren vlotter verloopt zonder de ballast van irrelevante details.



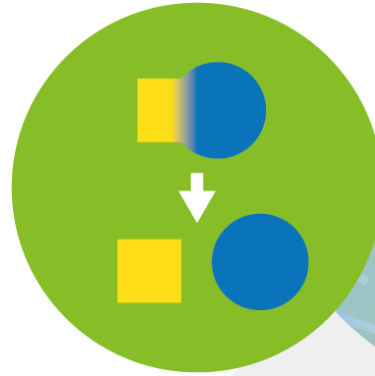
Basisconcepten van computationeel denken

Een probleem doordacht opsplitsen in goedgekozen deelproblemen, zodat elk deel afzonderlijk kan worden aangepakt, en een oplossing van het probleem gemakkelijker bekomen wordt.

Negeren van informatie die niet nodig is om een probleem op te lossen.

Bepaalde details verbergen om in grote lijnen over een probleem te kunnen nadenken. Bij het aanpakken van complexe problemen is er nood aan abstractie omdat het redeneren vlotter verloopt zonder de ballast van irrelevante details.

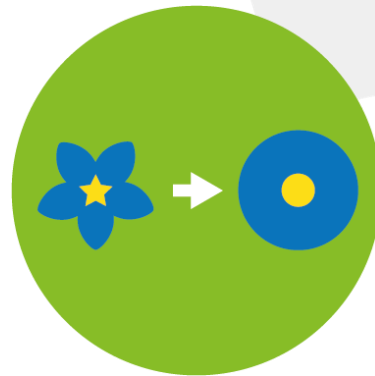
Decompositie



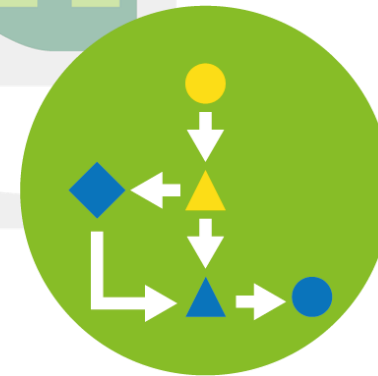
Patroonherkenning



Achterhalen dat bepaalde aspecten van een probleem gelijkenissen vertonen, waardoor het vereenvoudigd kan worden. Herkennen dat een probleem gelijkenissen vertoont met een eerder opgelost probleem.



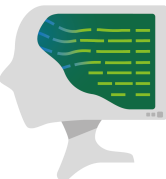
Abstractie



Algoritmisch denken

Expliciteren van een reeks eenduidige instructies die stapsgewijs moeten worden uitgevoerd. Inzien dat deze reeks van instructies en de volgorde ervan essentieel zijn om het gewenste resultaat op te leveren.

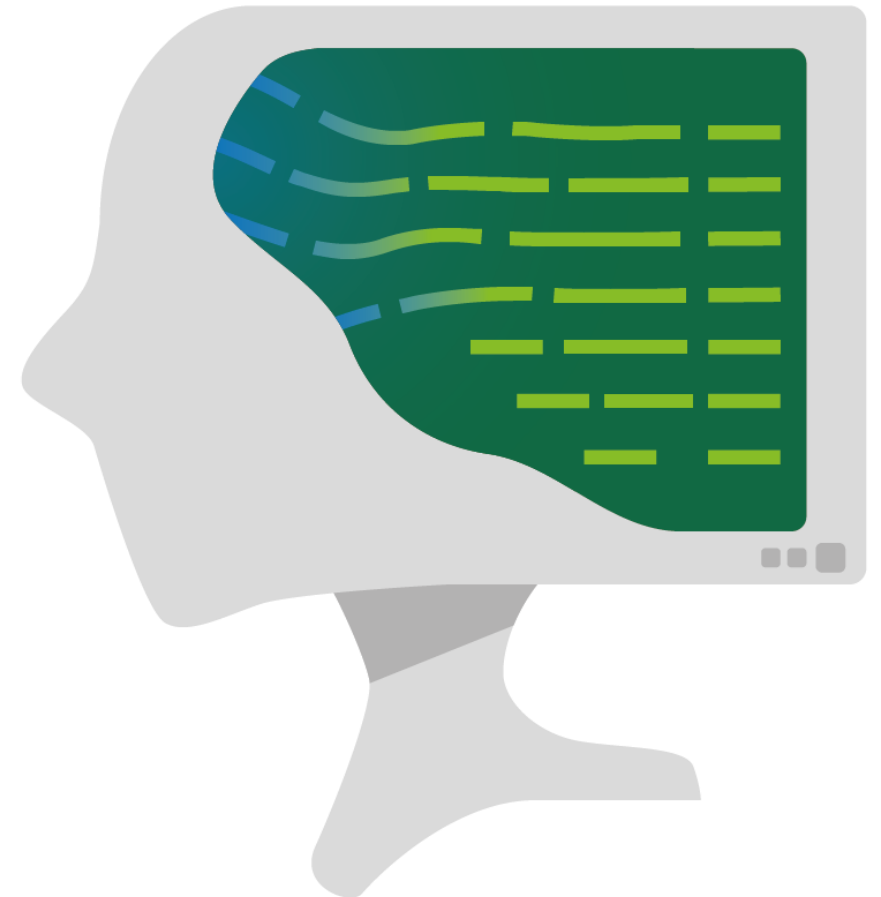
Voor een bepaald probleem zijn deze vier basisconcepten vaak verweven met elkaar.

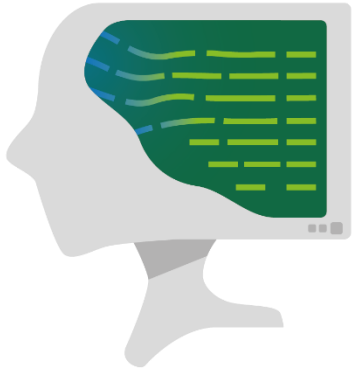


Computationeel denken is niet nieuw!

Het komt bijvoorbeeld voor als

- er een lampje brandt op je dashboard;
- je een taart bakt volgens een recept;
- een pianist een muziekstuk inoefent;
- je een gps gebruikt.





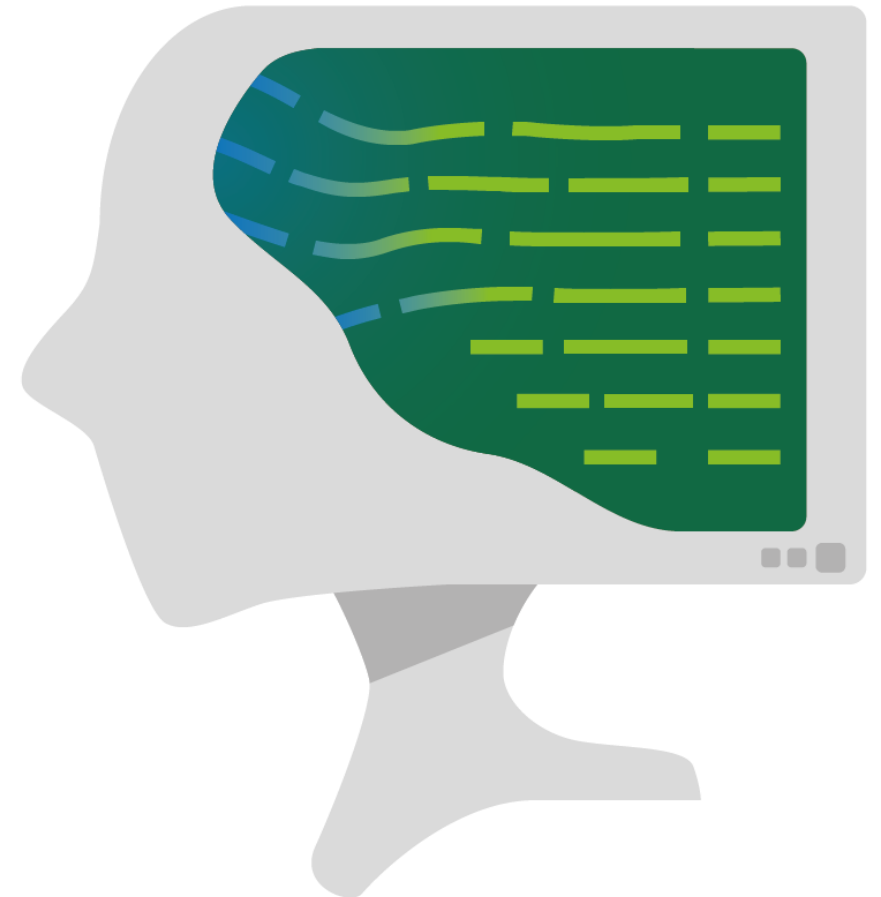
Basisvaardigheid zoals lezen, schrijven en rekenen

Vormt de basis van computerwetenschappen



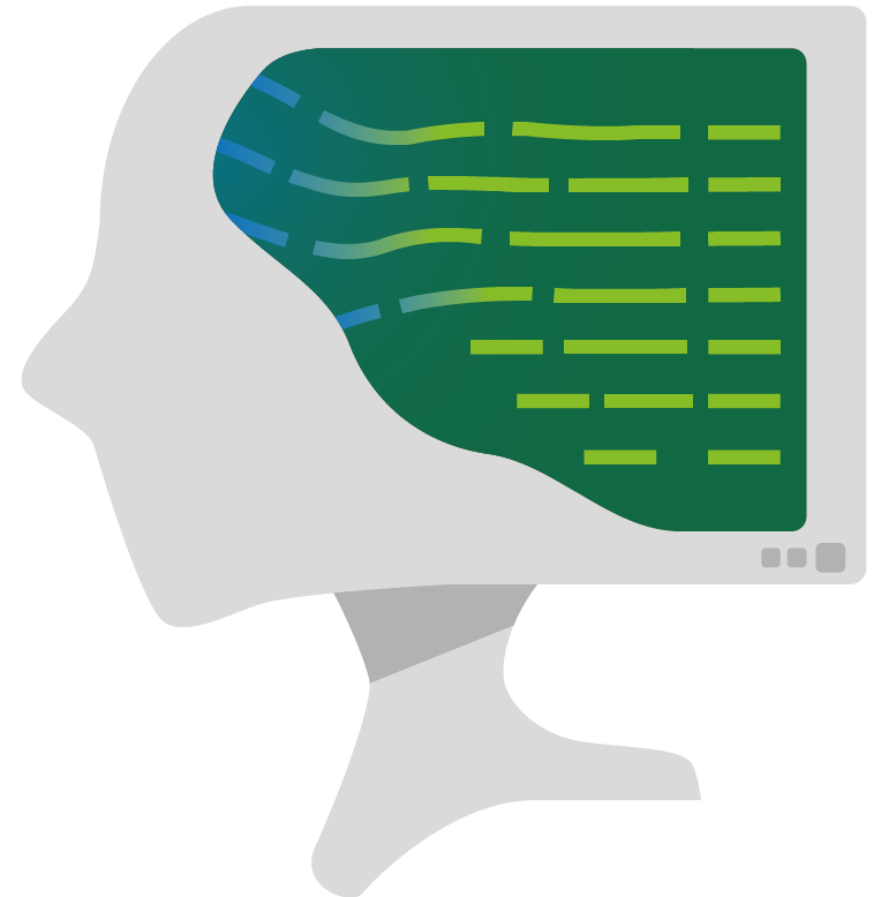
Concepten van computationeel denken aanbrenge

zonder de computer;
met de computer.



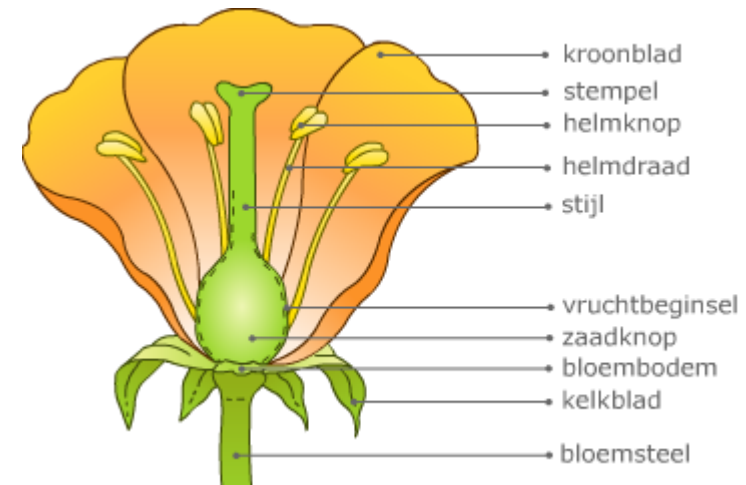
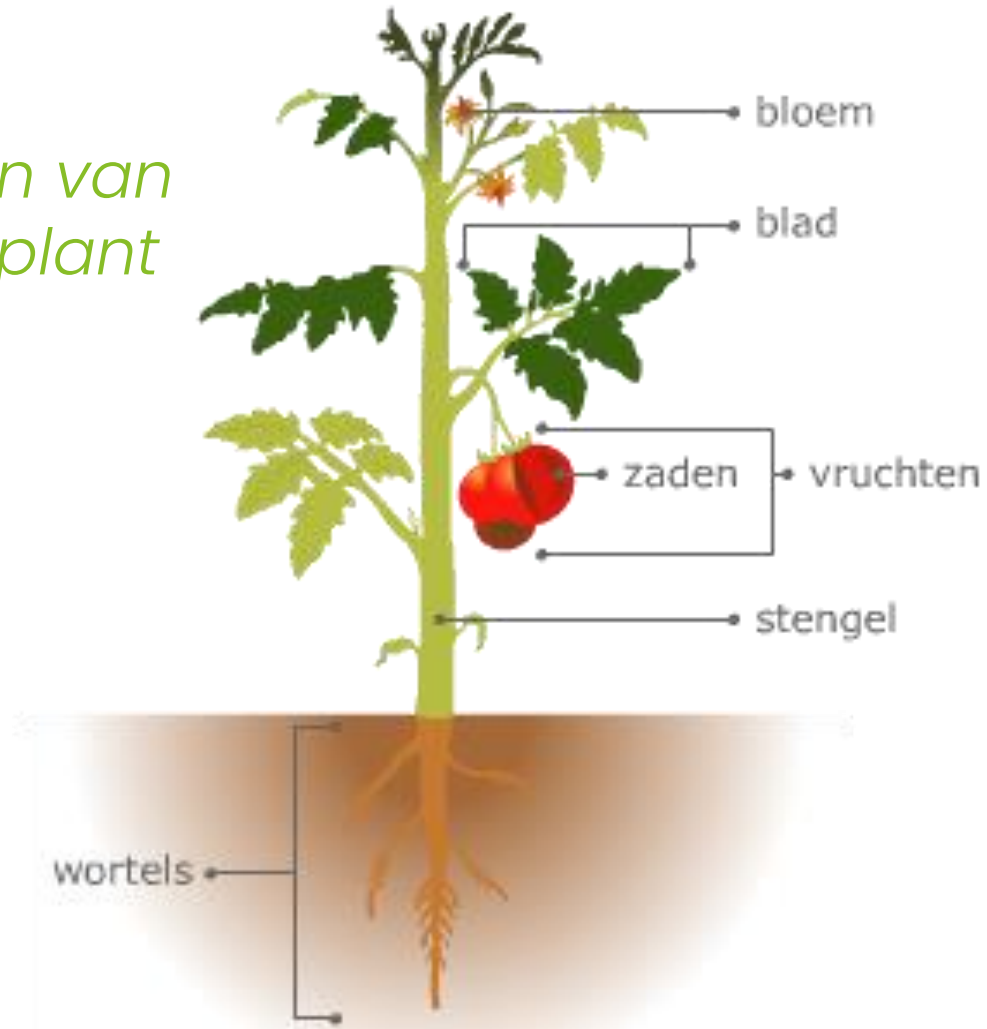
Concepten van computationeel denken aanbrenge

zonder de computer; *unplugged*
met de computer. *plugged*

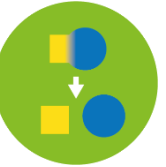


Decompositie - patroonherkenning

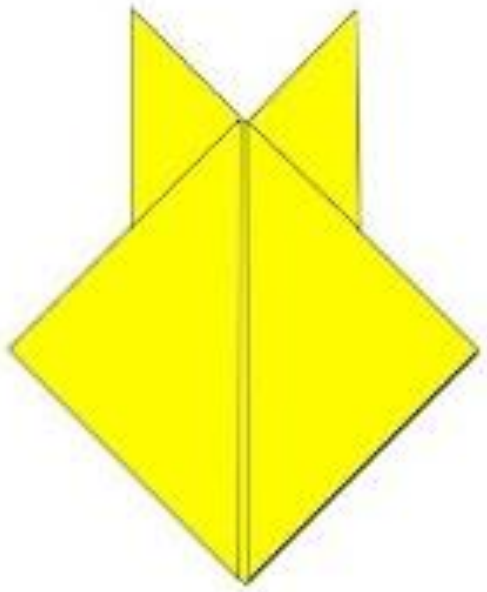
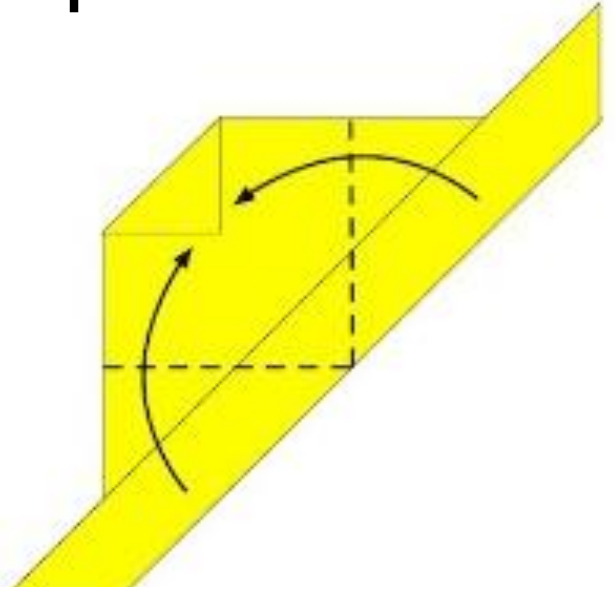
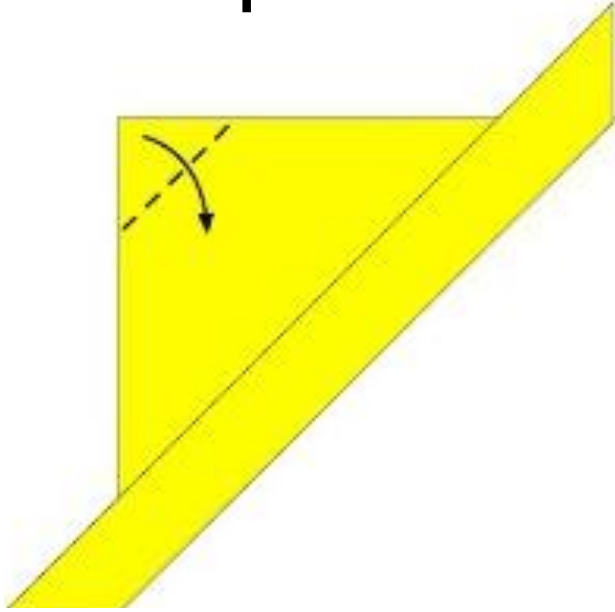
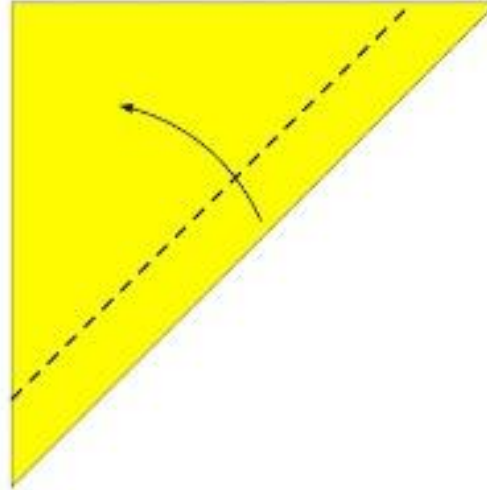
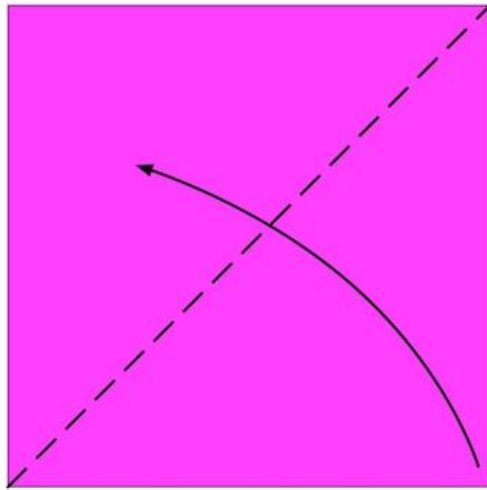
*Delen van
een plant*



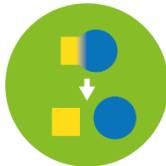
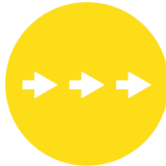
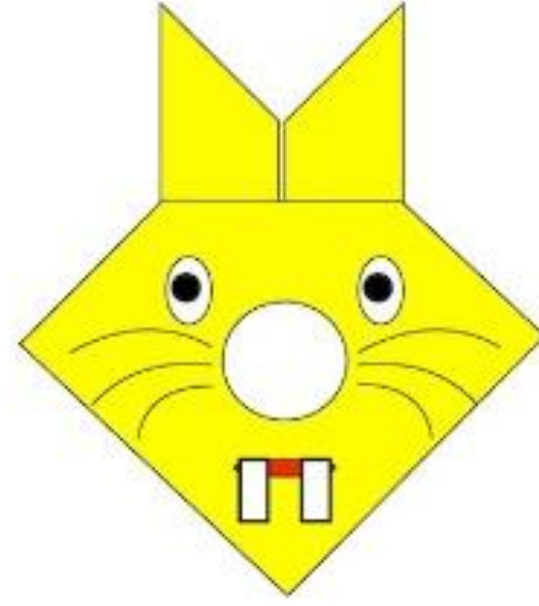
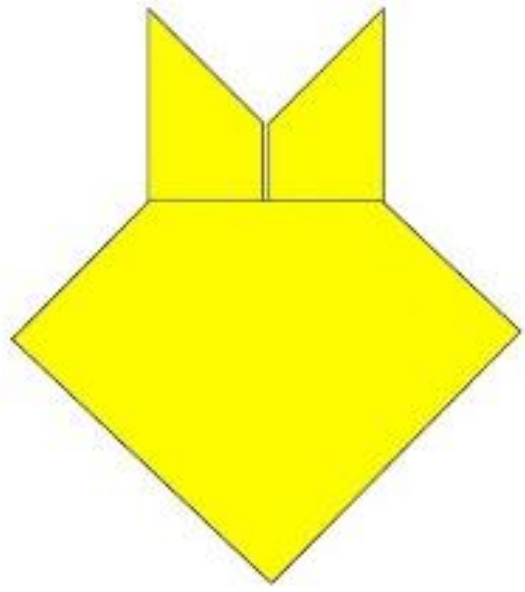
Afbeeldingen Corine Pol;
VO-content



Duidelijke instructies - decompositie - sequentie

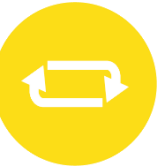
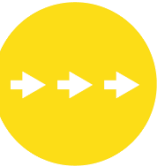
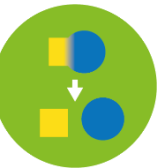
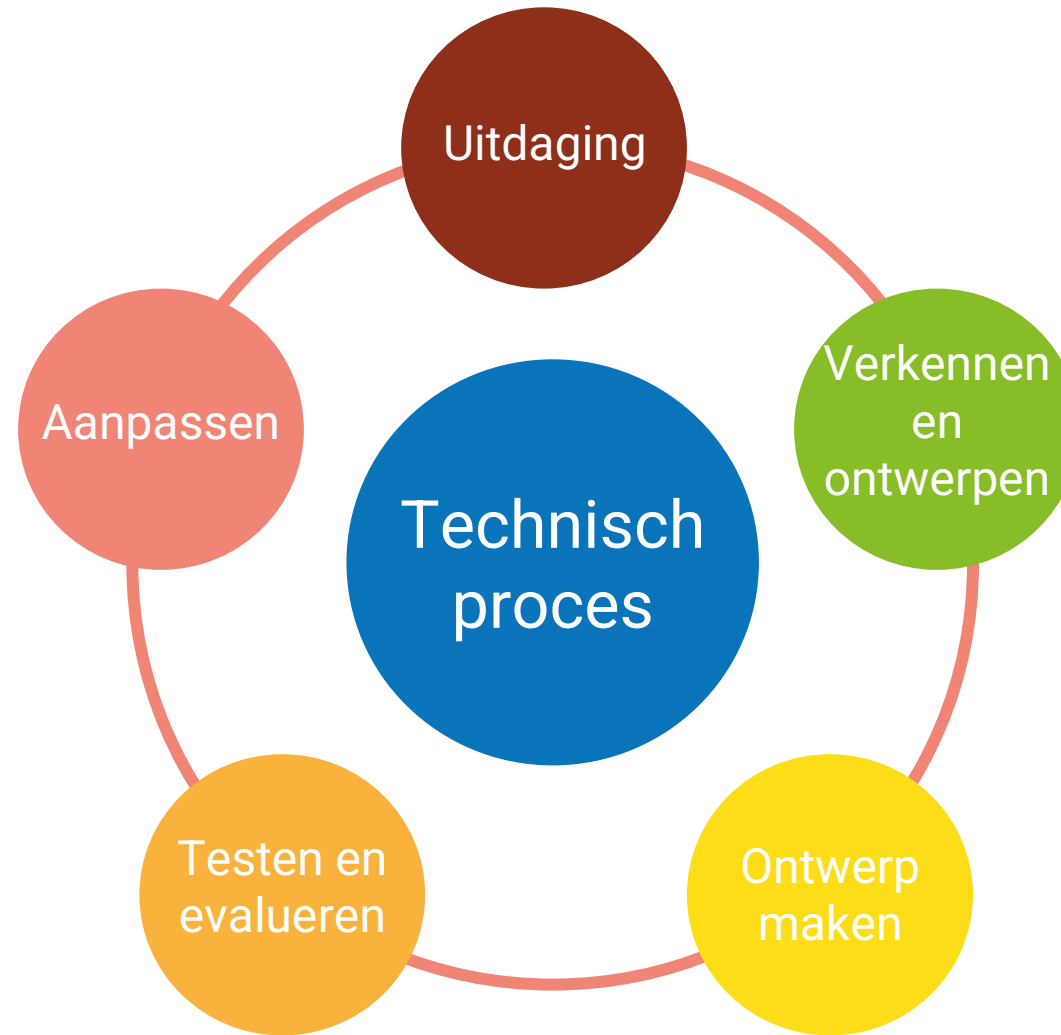



TURN
OVER

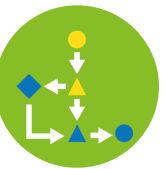


SAFN

Algoritme - decompositie – sequentie - herhaling



Sorteeralgoritme



<https://www.youtube.com/watch?v=eI0nniqgQnc>

Opdracht rijdende robot

Surf naar de **simulator** <https://blockly.dwengo.org/>
Test de gegeven code uit.

De robot zou voortdurend in een **vierkant** moeten rijden.

Verbeter de code door de getallen in de gegeven blokken aan te passen!

Maak enkel gebruik van de gegeven blokken, voeg geen blokken toe.

- Logica
- Lussen
- Formules
- Tekst
- Variabelen
- Dwenguino
- Sociale robot
- Commentaar

zet klaar

herhaal

dc-motor

nummer 1
snelheid -180

dc-motor

nummer 1
snelheid 100

wacht 200 ms

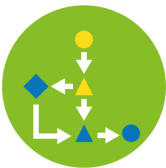
dc-motor

nummer 1
snelheid 5

dc-motor

nummer 2
snelheid -200

wacht 500 ms



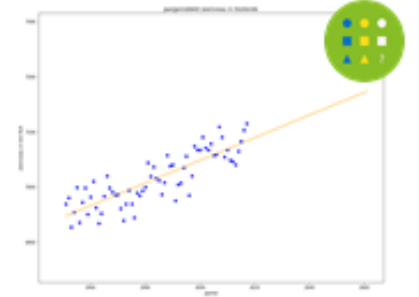
We willen analyseren hoe het zeeniveau in Oostende in de toekomst zal evolueren op basis van data uit het verleden.

Verkennen van het probleem. Wat heb ik nodig? Subtaken (**decompositie**):

1. Data verzamelen
2. Data visualiseren
3. Trendlijnen bepalen



Trendlijn?
Kijk naar de vorm van het spreidingsdiagram.
Ellipsvormig? Dat wijst op lineaire regressie.
(**patroonherkenning**)



De trendlijn is een benadering (model) van het schommelende zeeniveau die de stijgende trend weerspiegelt, en toelaat voorspellingen te doen voor de evolutie van het zeeniveau in de toekomst.
(**abstractie**)



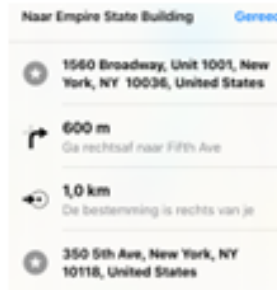
Een **algoritme** om in Python de trendlijnen te visualiseren (a.d.h.v. Python-modules)

- o Definieer gedaante van vergelijking rechte
- o Definieer functie om coëfficiënten trendlijn te bepalen passend bij gegeven datapunten
- o Lees data in
- o Bepaal a.d.h.v. data en bovenstaande definities de trendlijn
- o Visualiseer data en trendlijn



New York. Wat is de afstand in vogelvlucht van Times Square naar het Empire State Building, uitgedrukt in mijl? De routeplanner verstrekt de afstanden om van Times Square naar het Empire State Building te rijden.

Voorkennis van de leerlingen: twee types oefeningen op de stelling van Pythagoras, nl. berekenen van de schuine zijde als de twee rechthoekszijden gekend zijn, en berekenen van een rechthoekszijde als de schuine zijde en de andere rechthoekszijde gekend zijn.



Decompositie in subtaken:

1. Vat het probleem samen in een figuur.
2. Herken een rechthoekige driehoek in de figuur.
3. Duid op de figuur aan wat gekend is.
4. Zoek de lengtes van de twee rechthoekszijden op in de gegeven route.
5. Pas de stelling van Pythagoras toe.



Het stratenplan vertoont een **patroon**: de straten vormen een rechthoekig rooster. De afstand in vogelvlucht kan dus bepaald worden als de schuine zijde van een rechthoekige driehoek. Herkennen dat het **probleem verwant** is met een eerder opgelost probleem: oefening is er een van het type waarbij de schuine zijde berekend wordt als de twee rechthoekszijden gekend zijn.



De straten en kruispunten worden **geabstraheerd** tot zijden en hoeken van een rechthoekige driehoek. Enkele instructies van het programma worden gegroepeerd in een **functie**, waarbij een abstractie wordt gemaakt van de waarden en de gedetailleerde instructies uit het probleem.



Algoritme: Bereken de vierkantswortel van de som van de kwadraten van de parameters van de functie. (De parameters zijn de lengtes van de rechthoekszijden.)



```
def schuinezijde(a,b):  
    """Schuine zijde berekenen uit rechthoekszijden."""  
    c_kwadraat = a**2 + b**2  
    c = math.sqrt(c_kwadraat)  
    return c  
vogelvlucht = schuinezijde(rhz1, rhz2)  
print(vogelvlucht)
```

De twee rechthoekszijden rhz1 en rhz2 worden geïnitieerd in het programma, of opgevraagd aan de gebruiker.



Stel gelabelde data uit de zorgsector voor op een manier die geschikt is om er beslissingen mee te nemen betreffende een diagnose of een behandeling.

Subtaken (**decompositie**):

1. In de voorbeelden op zoek gaan naar een patroon (een gerichte gewortelde graaf, een binaire beslissingsboom).
2. Uit welke elementen is een beslissingsboom opgebouwd?
3. Wat is de werking van een beslissingsboom?
4. Hoe de mate van spreiding weergeven door een getal.
5. Welke berekeningen leiden tot de juiste splitsing?
6. Hoe geef je ja-nee vragen vorm a.d.h.v. een logische (wiskundige) uitdrukkingen?
7. Ontwerpen van een algoritme om een binaire beslissingsboom te construeren.
8. De implementatie van het algoritme in de computer.
9. De data voorverwerken tot het gewenste formaat.

De bestaande voorbeelden tonen dat zulke data vaak worden voorgesteld door een gerichte gewortelde graaf, meestal binair. M.a.w. een binaire beslissingsboom is geschikt om de data te representeren. (**patroonherkenning**)

Een binaire beslissingsboom vertrekt uit een wortel en een ja-nee vraag die een scheiding van de data oplevert in twee verzamelingen die zo weinig mogelijk spreiding over de categorieën vertonen. Dat gaat op een analoge manier verder met volgende ja-nee vragen. De boom wordt zo opgebouwd met takken en knopen. Tot slot geven de bladeren van de beslissingsboom de mogelijke beslissingen. (**patroonherkenning**)

De beslissingsboom is een abstracte voorstelling van de oplossing in de vorm van een graaf. Het is een model voor de oplossing. (**abstractie**)
Deze voorstelling is bovendien zeer transparant.

Algoritme om binaire beslissingsboom op te stellen.

1. Bekijk welke wortel de beste splitsing oplevert. Maak takken voor de mogelijke uitkomsten van de ja-nee vraag.
2. Bekijk voor elke tak welke knoop het meest geschikt is voor de volgende splitsing. Maak takken voor de mogelijke uitkomsten van de ja-nee vragen.
3. Herhaal de vorige stap telkens opnieuw voor de nieuw aangemaakte takken.
4. Stop met de herhaling als de bekomen verzamelingen homogeen zijn, dus enkel elementen uit een en dezelfde klasse bevatten.

Dit is een iteratief proces: om de wortel en de knopen te bepalen wordt telkens dezelfde techniek toegepast.



We willen een AI-systeem ontwikkelen dat reviews op socialemediaposts classificeert volgens sentiment (positief, neutraal, negatief).

Verkennen van het probleem. Wat heb je nodig?

Subtaken (**decompositie**):

1. Keuze maken tussen **regelgebaseerd** of datagebaseerd AI-systeem
2. Sentimentwoordenboek in Nederlands inladen
3. De reviews voorverwerken (tokeniseren, hoofdletters en leestekens verwijderen)
4. De reviews representeren a.d.h.v. woordenboekvorm van de woorden erin, samen met hun woordsoort
5. Matching tussen de woorden en het sentimentwoordenboek om sentimentscore elk woord te bepalen
6. Bepaal de sentimentscore en sentiment de reviews.



Het gebruik van een lexicon (woordenboek specifiek voor de taak) en tokeniseren is een techniek die veel wordt toegepast in taaltechnologie, bv. cyberpest detectie, auteursherkenning, automatisch vertalen, tekst genereren ... (**patroonherkenning**)



Door te tokeniseren worden de zinnen herleid tot de woorden die ze bevatten.

De hoofdletters en leestekens worden genegeerd. De woorden worden herleid tot hun woordenboekvorm en aangevuld met hun woordsoort. (**abstractie**)

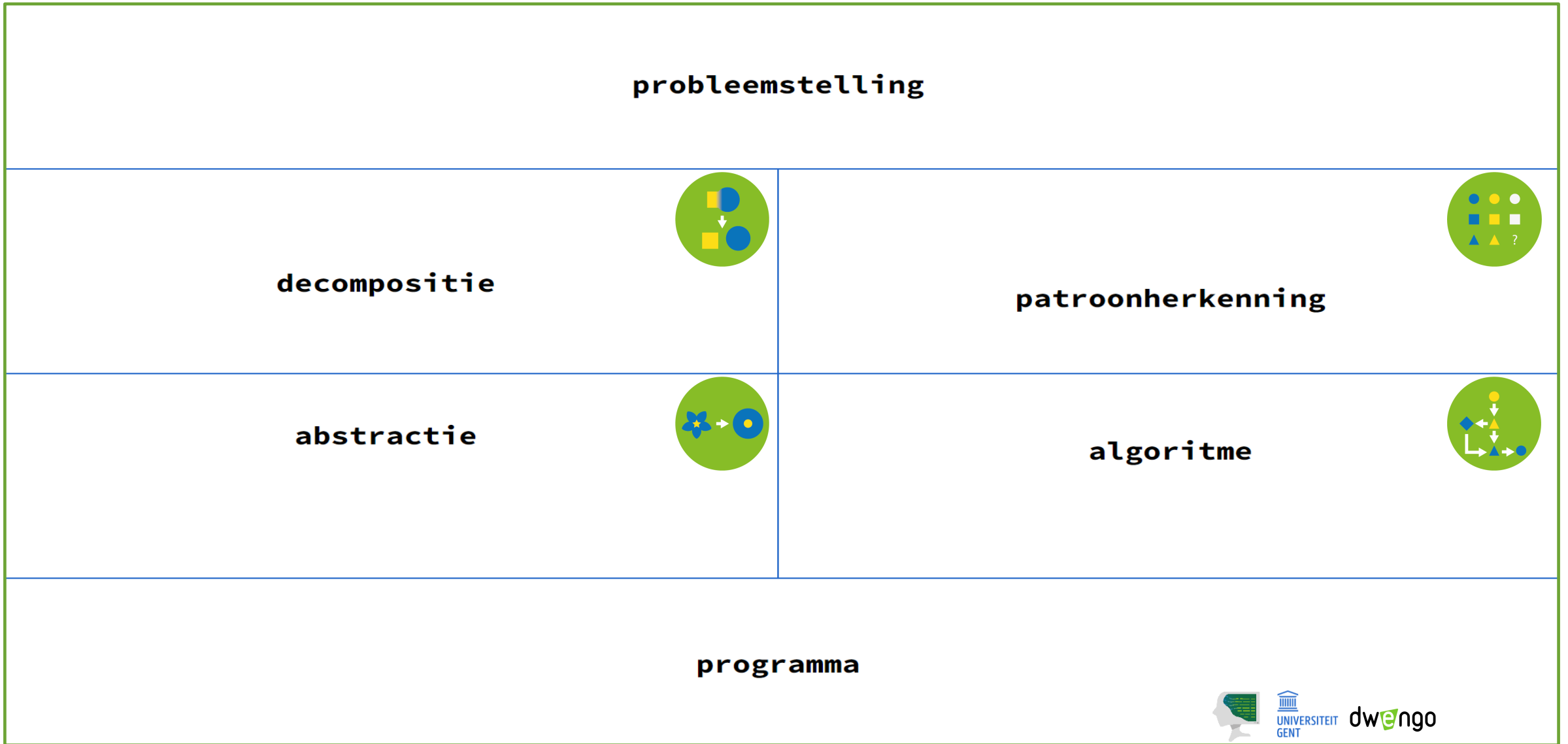
De voorverwerking laat toe om een datastructuur die geschikt is om efficiënt te zoeken, te gebruiken.



Een **algoritme** om het sentiment van een review te bepalen

- Maak een lijst van de woorden die in de zin voorkomen, met hun woordsoort. Negeer leestekens.
- Zoek elk woord op in het sentimentwoordenboek.
- Sla de score van elk woord op in een lijst.
- Tel de scores op.
- Bepaal het sentiment a.d.h.v. van een wiskundige uitdrukking (groter dan 0, gelijk aan 0, kleiner dan nul).





Waar vind je het materiaal?

dwengo.org

aiopschool.be

istem.be

blockly.dwengo.org

aiopschool.be/agenda