



Computational Thinking in de onderbouw gaat over het begrijpen van **patronen, het opdelen van taken in eenvoudige stappen en het geven en ontvangen van instructies**. Je leert hoe een serie geordende instructies werkt, zoals een recept in een kookboek. Je leert om herhalingen en overeenkomsten in patronen te zien en hoe je deze kunt sorteren. Je leert begrijpen dat taken vaak uit verschillende stappen bestaan en dat samenwerken een taak makkelijker kan maken. Daarnaast leer je om instructies te geven en te ontvangen en dat deze op verschillende manieren overgebracht kunnen worden, zoals mondeling maar ook in de vorm van een tekening. Computational Thinking gaat niet alleen over computers, het gaat om het leren denken als een **probleemoplosser**.

Leerdoelen

Ik begrijp dat een computer gebruik maakt van een serie geordende instructies of regels om te weten wat hij moet doen.

- Ik weet dat zo'n instructie gaat volgens een bepaald patroon.
- Ik begrijp dat een patroon een herhaling is van bijvoorbeeld instructies, regels, cijfers, vormen of tekens.

Ik kan patronen herkennen in vorm, kleur en model.

- Ik kan verschillen en overeenkomsten in patronen herkennen en benoemen.
- Ik kan voorwerpen sorteren op vorm, kleur en model.
- Ik kan de (basis)vormen uit een voorwerp herkennen en benoemen. Bijvoorbeeld: een bloem bestaat uit één grote ronde cirkel en zes kleinere ronde cirkels

Ik kan de overeenkomsten benoemen tussen taken.

- Ik begrijp dat een taak opgeknipt kan worden in meerdere stappen en dat hier een logische volgorde in zit.
- Ik begrijp dat de ene taak leidt tot een volgende taak. (oorzaak en gevolg)
- Ik begrijp dat onderdelen van een systeem goed moeten samenwerken om een proces soepel te laten verlopen.

Ik kan een reeks instructies aan een klasgenoot geven voor het uitvoeren van een bepaalde taak.

- Ik weet dat een instructie een stappenplan is waarin beschreven staat hoe je iets moet uitvoeren.
- Ik begrijp dat er verschillende manieren zijn waarop je een instructie kunt overbrengen (mondeling, schriftelijk, getekend, via een video).
- Ik kan verschillende opdrachten achter elkaar geven.
- Ik kan zelf meerdere opdrachten achter elkaar uitvoeren zodat er een patroon ontstaat

Ik kan een taak opdelen in stappen en begrijp dat bepaalde stappen tegelijkertijd uitgevoerd kunnen worden.

- Ik begrijp dat ik tegelijk met een klasgenoot aan eenzelfde taak kan werken.
- Ik begrijp dat verschillende stappen soms tegelijkertijd uitgevoerd kunnen worden (of juist niet).
- Ik begrijp dat de opdracht sneller uitgevoerd kan worden als we deze eerst verdelen over meerdere personen.



In de middenbouw draait Computational Thinking om het eerste begrip van de logica achter computers en technologie. Je leert welke problemen je kan oplossen met behulp van een computer en dat deze ook kunnen helpen bij eenvoudige dagelijkse taken, zoals het regelen van de verlichting thuis. Je leert patronen en volgorden herkennen en je maakt kennis met het begrip '**algoritme**' (= een reeks stappen die steeds herhaald kunnen worden). Dit helpt je te begrijpen dat computers werken volgens het '**als-dan-principe**.' Deze kennis en vaardigheden helpen je bij het ontwikkelen van logisch denken en probleemoplossend denken. Zo leer je bijvoorbeeld nadenken over oorzaak en gevolg, of hoe je een probleem kan oplossen door de situatie eerst na te spelen.

Leerdoelen

Ik kan omschrijven welke problemen opgelost kunnen worden met een computer.

- Ik weet dat digitale middelen kunnen worden gebruikt om het leven in huis en in de stad makkelijker te maken, denk aan verlichting, thermostaat, tijdsaanduiding bij de bushalte.
- Ik begrijp dat sommige handelingen handmatig en sommige handelingen automatisch worden uitgevoerd en ik begrijp dat sommige handmatige handelingen 'geautomatiseerd' kunnen worden.
- Ik begrijp dat een computer geprogrammeerd moet worden, zodat apparaten of machines automatisch worden aangestuurd.

Ik kan patronen en volgorden in tekst, cijfers, beeld en geluid herkennen en benoemen.

- Ik kan patronen herkennen en benoemen en ook afwijkingen hierin herkennen en benoemen
- Ik kan een sortering aanbrengen in patronen
- Ik kan een variatie bedenken op een patroon.

Ik kan omschrijven hoe digitale technologie een rol kan spelen bij het uitvoeren van taken die steeds herhaald worden

- Ik weet dat handelingen die steeds herhaald worden in het dagelijks leven, regelmatig door computers worden uitgevoerd.
- Ik begrijp dat handelingen die steeds herhaald worden kunnen worden weergegeven in een model of schema, bijvoorbeeld een knikkerbaan.

Ik heb kennis gemaakt met het begrip 'algoritme' en begrijp dat deze werkt volgens het 'als-dan-principe'.

- Ik weet wat oorzaak en gevolg betekent en dat computers deze gebruiken in de zin van 'als dit, dan dat'.
- Ik weet dat een algoritme een reeks opeenvolgende stappen is, die zich steeds herhalen.
- Ik kan een situatie herkennen en benoemen waarin gebruik wordt gemaakt van een algoritme. Bijvoorbeeld: zorgen voor een digitaal huisdier. 'Als ik het eten geef, dan is het blij'. Of een stoplicht: 'Als er weinig verkeer is dan springt het stoplicht eerder op groen'

Ik begrijp hoe je problemen kunt oplossen door de situatie eerst na te spelen, om zo te voorspellen wat er gaat gebeuren.

- Ik kan een spelletje spelen op de computer om te begrijpen hoe iets echt werkt of om een echt probleem op te lossen. (Bijvoorbeeld met het spel 'SimCity', dit is een simulatie van hoe echte steden functioneren, of 'Toca Lab: Elements' waar kinderen virtueel proefjes kunnen doen)
- Als ik een oplossing voor een probleem heb gevonden, kan ik kijken of deze ook in andere situaties werkt.



In de bovenbouw draait Computational Thinking om het verder ontwikkelen van het denkvermogen in relatie tot computers en technologie en hoe je dit kunt toepassen om problemen op te lossen. Je leert niet alleen over de werking van computers, maar ook hoe ze deze kunt inzetten om **creatieve oplossingen** te bedenken. Bijvoorbeeld door met digitale hulpmiddelen eenvoudige computerprogramma's te maken die je kunnen helpen om problemen te simuleren en op te lossen. Je leert problemen op te splitsen in logische onderdelen en deze om te zetten in herhaalbare stappen (**algoritmes**). Ook leer je het **nut** van computers te begrijpen bij automatisering van repeterende (herhalende) taken in allerlei vormen, zoals bijvoorbeeld in een fabriek.

Leerdoelen

Ik kan een probleem kleiner maken door deze te verdelen in verschillende onderdelen

- Ik kan een probleem opdelen in logische onderdelen. (Bijvoorbeeld: als ik een grote puzzel maak zoek ik eerst de hoekjes en stukjes met dezelfde kleuren uit).
- Ik kan de onderdelen in een logische volgorde plaatsen.
- Ik kan de onderdelen vertalen naar een reeks logische stappen vertalen die zichzelf steeds herhalen (algoritme).

Ik kan een stappenplan maken om zo een probleem op te lossen.

- Ik weet dat een algoritme een reeks opeenvolgende stappen is, die zich steeds herhalen.
- Ik kan een probleem oplossen met behulp van een algoritme.
- Ik weet dat robots werken met een algoritme dat door de mens is geprogrammeerd.

Ik kan omschrijven hoe computers en de programma's die ik gebruik, kunnen samenwerken aan een oplossing.

- Ik begrijp dat een computerprogramma kan helpen bij een terugkerende taak zodat deze wordt geautomatiseerd.
- Ik ken het begrip 'kunstmatige intelligentie' en kan één of twee voorbeelden noemen welke invloed het heeft op de samenleving en de toekomst.
- Ik weet dat een computer alleen kan werken als deze wordt aangestuurd door een mens.

Ik kan een oplossing van een probleem testen en waar dat nodig is aanpassen.

- Ik kan met passende digitale hulpmiddelen een eenvoudig computerprogramma schrijven dat mijn probleem simuleert (nabootst). (bijvoorbeeld met 'Blockly' een programmeertaal voor kinderen die werkt met blokken en variabelen, zoals het programma Scratch)
- Ik kan controleren of het computerprogramma dat ik gemaakt heb klopt en begrijp wat ik moet doen om het eventueel aan te passen.

Ik kan omschrijven hoe een oplossing ook bij een ander probleem ingezet kan worden.

- Ik kan de stappen van een probleem weergeven in een model, tekening of schema. Bijvoorbeeld: hoe kom ik van de school naar huis? Teken of programmeer een routekaart.
- Ik kan een model, tekening of een schema gebruiken voor het oplossen van verschillende, soortgelijke problemen. Bijvoorbeeld: ik kan dezelfde routekaart gebruiken om ook de weg van huis naar de winkel te tekenen of programmeren