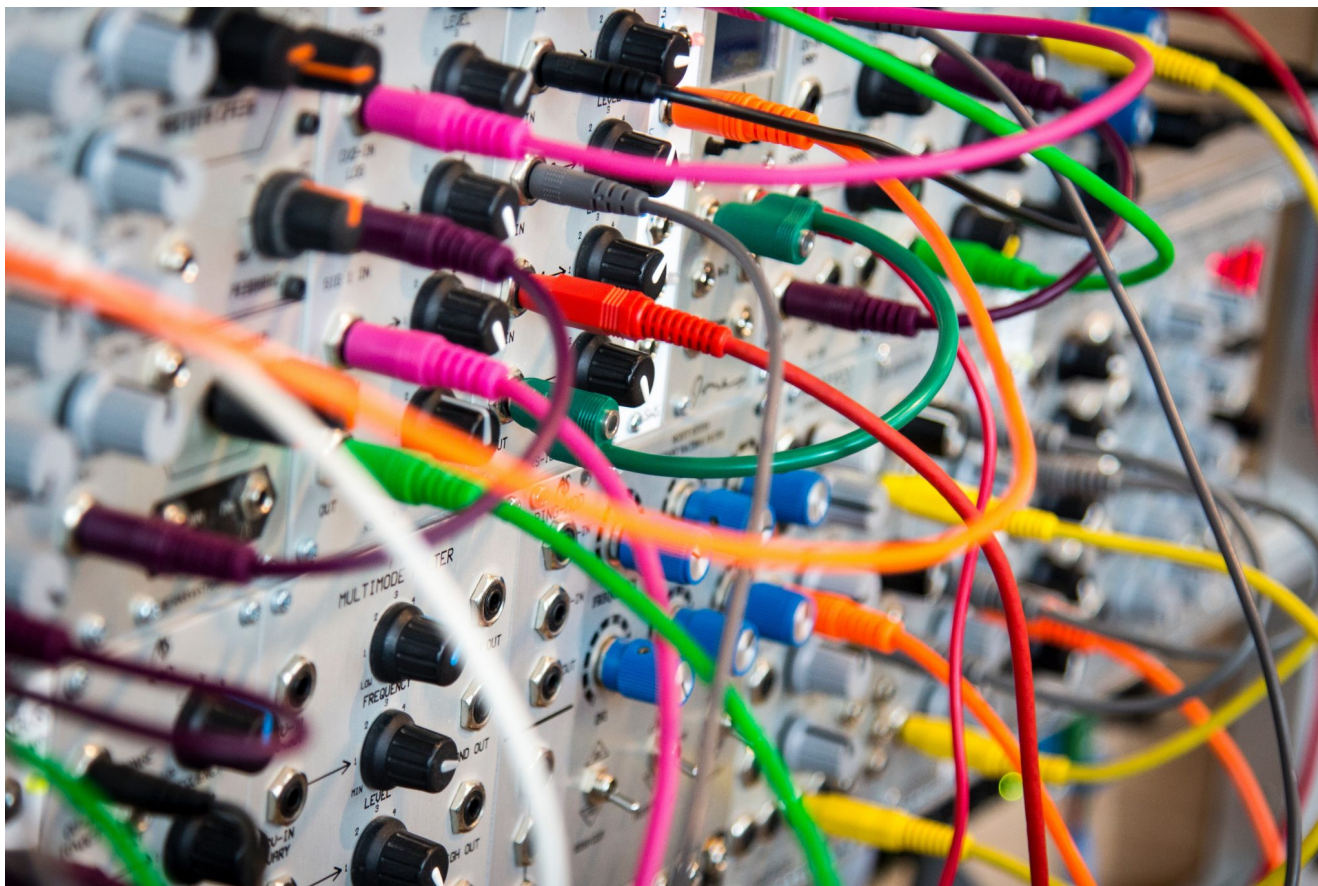


Computational thinking: de drie grootste misvattingen

written by Michiel Lucassen

7 februari 2019



Computational thinking is een begrip dat binnen steeds meer scholen bekend wordt. Het leren omgaan met complexe problemen is een belangrijke vaardigheid, en computational thinking helpt daar goed bij. Bij zo'n nieuw begrip zie je dan wel meteen dat er ook veel misconcepties zijn. In dit artikel kijken we naar drie grootste misvattingen rondom computational thinking.

Wat is het ook al weer?

Computational thinking is een pakket aan denk-vaardigheden en strategieën die helpen bij het oplossen van complexe problemen. Vaak wordt dit gekoppeld aan het omgaan met digitale technologie: het leren kijken naar een probleem zodat een computer het kan oplossen. Hierbij horen vier belangrijke begrippen: decompositie, patronen herkennen, abstractie en algoritmes. Het klinkt misschien ingewikkeld, maar dat valt best mee. [In dit artikel](#) keken we eerder naar deze

begrippen en de toepassing ervan in het onderwijs.

Misvattingen

Het koppelen van computational thinking aan computers is erg logisch: de termen komen dan ook vanuit de digitale wereld. Maar het zorgt ook meteen voor grote misvattingen! Zo zou computational thinking alleen maar gaan om programmeren en kan het alleen maar digitaal. Dat betekent dan dus ook dat het ingewikkeld is en helemaal niet nodig is voor iedereen. Of toch niet?

Misvatting 1: Computational thinking is hetzelfde als programmeren.

Misschien wel de hardnekkigste misconceptie rondom computational thinking: het is hetzelfde als [programmeren](#). Ook andersom wordt de redenering vaak gemaakt: programmeren in de les is hetzelfde als werken aan de vaardigheden van computational thinking. Veel lesmateriaal rondom computational thinking is gemaakt voor programma's als [Scratch](#), dus misschien is het logisch dat deze link gelegd wordt. De waarheid is anders: programmeren is echt iets anders dan computational thinking. Je kunt gebruik maken van de verschillende denkvaardigheden van CT bij programmeren, maar dat hoeft helemaal niet. Ook hier geldt het voorbeeld andersom: door te programmeren kun je werken aan CT, maar dat is zeker niet noodzakelijk.

Misvatting 2: Je hebt altijd een computer nodig (of een microbit!)

Een andere denkfout die rondom computational thinking gemaakt wordt is dat het altijd digitaal moet. Zonder [microbit](#) kun je dat niet leren, en iedere leerling moet natuurlijk een laptop hebben! Ook dit klopt niet. Computational thinking gaat over het systematisch in stukken hakken van een complex probleem, en hierdoor kijken op welke manier het opgelost kan worden. En dat kan heel goed zonder apparaat! Sterker nog: een goede basis is juist om eerst te beginnen met 'unplugged' activiteiten. Een beetje onhandig is dat dit soort activiteiten wel vaak weer het woord programmeren bevatten, maar het gaat telkens om het leren werken in stappen. [In dit artikel](#) verzamelden we een aantal mooie voorbeelden.

Misvatting 3: Computational thinking is ingewikkeld en niet voor iedereen nodig.

Ook dit is iets wat je vaker hoort als het om computational thinking gaat. En

eerlijk: eigenlijk is dat ook wel terecht. Het begrip 'computational thinking' is al vervelend omdat er geen goede Nederlandse vertaling voor is. Daarnaast zijn de andere begrippen (decompositie, patronen herkennen, abstractie en algoritmes) ook zeker niet eenvoudig. En dat is jammer, want inhoudelijk is het namelijk een goede set aan vaardigheden waar elke leerling wat aan heeft. Het leren om een groot probleem in kleine stukjes te verdelen en zo stap voor stap het probleem op te lossen is een vaardigheid waar iedereen iets aan heeft. Door ook nog te leren om hier zelf een (systematische) oplossing voor te bedenken zorg je ook nog eens dat het probleem blijvend opgelost is. Leren plannen? Nieuwe ideeën bedenken? Werken met deadlines? Onderzoek doen voor school? Allemaal dingen waarbij je iets hebt aan computational thinking.

Aan de slag dus!

Zo bekeken is het dus zeker iets wat de moeite waard is om te behandelen in de klas. Sterker nog: vanuit [Curriculum.nu](https://www.curriculum.nu) werkt het team [Digitale Geletterdheid](#) aan het toekomstige curriculum waarin computational thinking ook een duidelijke plaats krijgt. De komende jaren zal er dus steeds meer aandacht voor komen, en hopelijk zullen deze misvattingen dan ook niet meer relevant zijn. Elke leerling heeft er namelijk iets aan om op een logische manier problemen op te lossen, of dat nu digitaal of analoog is.

Dit artikel is afkomstig van [Vernieuwend onderwijs.nl](https://www.vernieuwend onderwijs.nl).

Bekijk de meeste recente versie van ons artikel op onze website.

