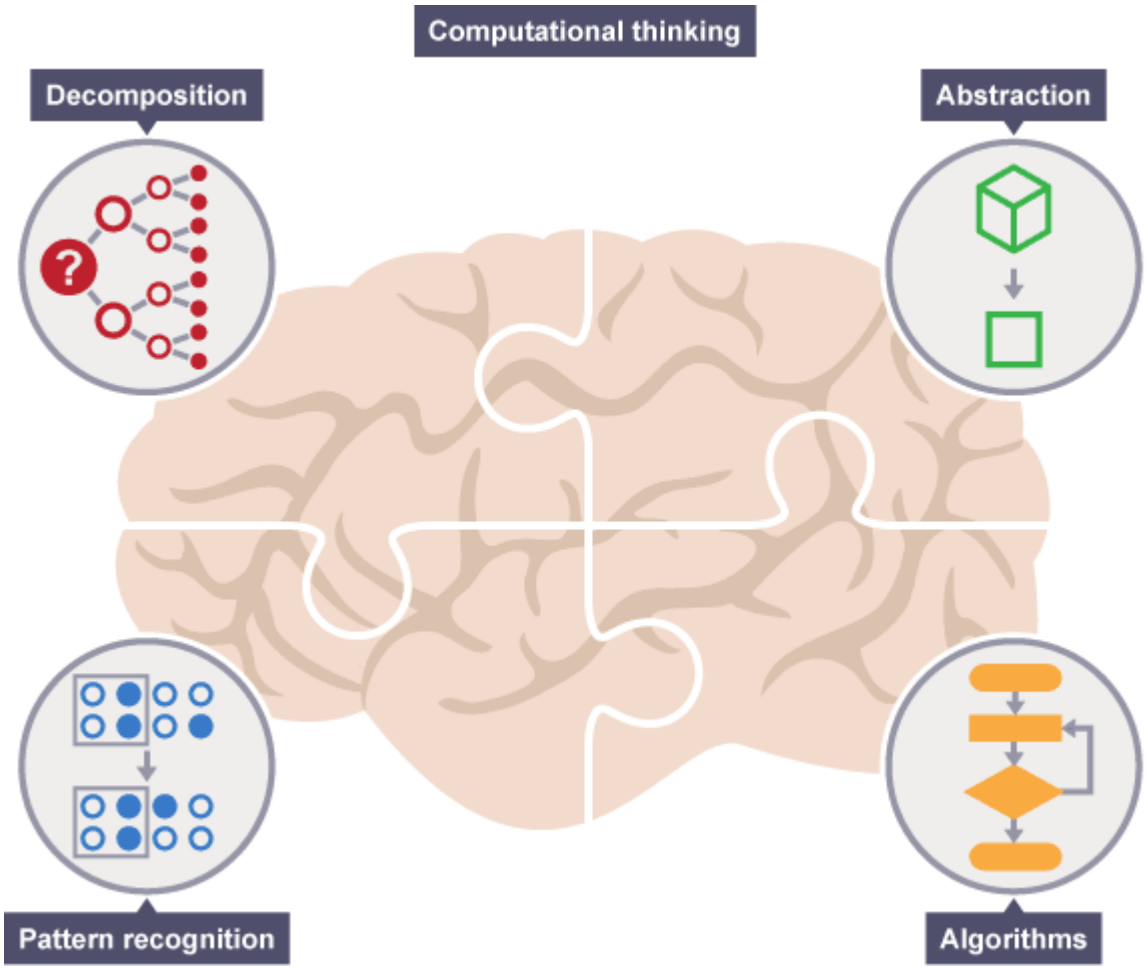
Computational Thinking. Ineens was het er. Tenminste, zo lijkt het er op. Kreeg het begrip eerst ruimte in het eindverslag van Onderwijs2032, een weekje later stond het ook in het nieuwe model van Kennisnet voor de 21e eeuwse vaardigheden. Maar wat is Computational Thinking eigenlijk? Moeten alle leerlingen in enen en nullen gaan spreken? Of gaat het meer over logisch denken?

Computational Thinking: een definitie

Volgens wikipedia is Computational Thinking een proces om problemen op te lossen. De term is bedacht door Seymour Papert in 1980 en sluit goed aan op zijn gedachten over maken in het onderwijs. Binnen Computational Thinking worden er 11 verschillende 'concepten' onderscheiden. Deze concepten zijn deels gebaseerd op hoe computers werken, maar gaan vooral over het leren denken op een logische manier. Computational Thinking is dus ook niet, zoals vaak gedacht wordt, hetzelfde als programmeren.

De concepten binnen Computational Thinking

De 11 concepten van Computational Thinking kunnen verdeeld worden in 4 groepen, zoals de BBC in de onderstaande afbeelding goed laat zien:



Maar wat houden die verschillende groepen in? Hieronder een korte uitleg:

Decompositie (Decomposition):
De eerste groep is 'decompositie'. Door grote problemen op te delen in kleine stukjes wordt het steeds eenvoudiger om het op te lossen. Dit is een nuttige vaardigheid voor leerlingen, want door stap voor stap kleine problemen op te lossen kun je uiteindelijk een groot probleem opgelost hebben.

Patronen herkennen (Pattern recognition):
Wanneer een probleem ontleed is en opgedeeld is in verschillende stukjes, wordt het tijd om te kijken naar patronen. Door te zoeken naar verschillende onderdelen die op elkaar lijken kan een groot probleem eenvoudiger opgelost worden, omdat niet telkens hetzelfde gedaan moet worden.

Abstractie (Abstraction):
Deze groep gaat over het filteren van de informatie. Wat is echt belangrijk? Wat raakt de kern van het probleem? Door te filteren krijgen leerlingen een beter idee van het probleem dat ze op proberen te lossen en kunnen ze effectiever aan de slag om het op te lossen.

Algoritmes (Algorithms):
De laatste groep gaat over de plan van aanpak: het maken van een algoritme. Door de te volgen stappen uit te werken kan een probleem opgelost worden. Dit kan natuurlijk alleen als het probleem helder is en opgedeeld is in verschillende onderdelen.

Kanttekeningen bij Computational Thinking

Zo gezien zijn het dus vaardigheden die leerlingen helpen om een beter overzicht te krijgen van complexe problemen. Toch iets anders dan denken als een computer of leren programmeren. Maar lijkt Computational Thinking dan eigenlijk niet heel erg op het 'ouderwetse' redeneren? En wat als Computational Thinking een strategie is om problemen op te lossen, waarom staan beiden dan in het nieuwe 21st century skills model van Kennisnet en SLO? Kortom, er zijn verschillende kanttekeningen die gemaakt kunnen worden bij Computational Thinking en het huidige gebruik ervan. Daarnaast is het op z'n minst interessant om te zien dat grote partijen als Google er mee bezig zijn. Gaat het dan om goed onderwijs, of om goede werknemers al vroeg op te leiden?

Computational Thinking in de klas

Hoewel de kanttekeningen zeker gemaakt kunnen worden is Computational Thinking als manier om problemen op te lossen interessant. Als je namelijk voorbij deze kanttekeningen kijkt zie je dat het inhoudelijk een erg sterke set vaardigheden is, waar elke leerling iets aan heeft. Dus wil je er mee aan de slag? Ben je op zoek naar werkvormen voor Computational Thinking in de les? In het engels zijn er veel verschillende bronnen te vinden die hierbij direct bruikbaar zijn. Een goede start is het al eerder genoemde materiaal van Google en van de BBC, of neem een kijkje in ons artikel over 'unplugged' activiteiten!