Dit document is gedownload door () op November 24, 2026.
Mag alleen gekopieerd worden voor gebruik binnen de school van 'gebruiker'. Zie ook de abonnementsvoorwaarden van Leapo.

Computational thinking doe je ZO!

'Computational thinking doe je ZO!' is het tweede deel van vier Digi-doeners om digitale geletterdheid concreet te kunnen maken in de bovenbouw van het basisonderwijs. De vier onderdelen van digitale geletterdheid (mediawijsheid, computational thinking, ICT-basisvaardigheden en informatievaardigheden) komen op een creatieve, ervaringsgerichte en laagdrempelige manier aan bod. In dit tweede deel ontdekken de leerlingen wat computational thinking is, wat het betekent en ze gaan computational thinking offline én online ervaren. De les heeft zowel doe- als praatopdrachten en is daardoor afwisselend en interactief. Totale duur: 1 à 2 uur.

★ TIP

Maak van deze Digi-doener een complete themamiddag! Wanneer je alle opdrachten doet zoals beschreven, ben je twee uur bezig. Je kunt er ook voor kiezen om het online programmeren (slide 7) aan te bieden als verrijking op een ander moment.

VERBINDING MET BEROEPEN EN DE ARBEIDSMARKT

Computational thinking is een belangrijke vaardigheid die je in veel beroepen terugziet. Het kunnen denken als een computer, computational thinking, geeft veel steun in verschillende beroepen. Computational thinking is een manier van denken om problemen op te kunnen lossen. Wanneer je computational thinking eigen hebt gemaakt, ben je in staat goed na te denken over informatie, dit in kleine stukjes te verdelen, te versimpelen en in een schema te zetten. Ook heb je vaardigheden geleerd om mogelijke oplossingen te bedenken en uit te proberen volgens een algoritmisch schema. Voor veel beroepen zijn deze vaardigheden onmisbaar! Computational thinking komt dan ook op veel plekken voor. Denk maar eens aan de programmeurs van Google die complexe programma's schrijven om hun software foutloos te laten werken. Wetenschappers die simulaties uitvoeren moeten ook algoritmisch kunnen denken en architecten moeten hun ontwerpplan concreet kunnen maken in een 3D-programma als SketchUp.

⚖ ETHIEK EN TECHNOLOGIE

Wij en onze leerlingen groeien op in een wereld waarin computers, machines en apparaten onmisbaar zijn. Deze apparaten groeien steeds meer uit tot zelfdenkende apparaten dankzij smart technologie en kunnen op den duur ook mensenwerk gaan overnemen. De computers krijgen daardoor verantwoordelijkheden te dragen, waarvan we niet weten tot hoe ver ze deze kunnen dragen. In de maatschappij waarin onze kinderen opgroeien zal dit nog normaler zijn, kijk maar eens naar de snelle ontwikkelingen van smart apparaten. Onze leerlingen zijn immers al behoorlijk gewend aan zelfdenkende apparaten. Maar in hoeverre blijven onze leerlingen, de burgers van de toekomst, eigenaar over dit proces en de apparaten? Is er voor hen nog een weg terug als ze voelen dat ze geen eigenaar meer zijn over hun eigen technologie?

LESOPBOUW

- Introductie: De leerlingen leren het begrip computational thinking kennen.
- Verdieping: De leerlingen leren welke basisvaardigheden van belang zijn bij computational thinking.
- Doen: De leerlingen gaan offline én online ervaren hoe computational thinking werkt.
- Afronding: We blikken terug op wat de leerlingen hebben geleerd.

VOORBEREIDING & BENODIGDHEDEN

Van tevoren kun je een aantal dingen doen:

- Lees de handleiding en lesbrief.
- Digibord met internetverbinding: klik door de slides voor op het digibord.
- Verzamel vijftien devices (één per groepje).
- Maak groepjes van twee of drie leerlingen.
- Print de lesbrieven.
- Bekijk eventueel alvast de video's uit deze les.
- Bekijk eventueel de site www.code.org/dance en speel alvast de eerste levels.

Stichting FutureNL ontwikkelt onafhankelijk lesmateriaal. © 2021 Iedereen mag deze les downloaden, delen en uitprinten (alleen niet als je er geld mee wilt verdienen). Doen! Creative Commons licence: CC BY-NC-ND 4.0. Deze les is gemaakt door Frank Veenstra.

Dit document is gedownload door () op November 24, 2026.
Mag alleen gekopieerd worden voor gebruik binnen de school van 'gebruiker'. Zie ook de abonnementsvoorwaarden van Leapo.

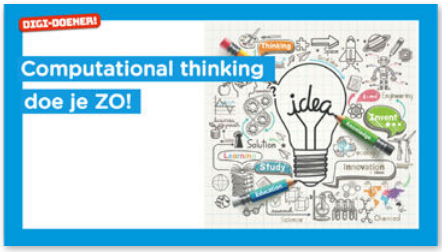
DOEL VAN DE LES

Domein curriculum 2021	Leerdoelen digitale vaardigheden	Kerndoel vak	21st century skills
1 Data & informatie DG1.2 Digitale data.	**1 Computational thinking** De leerling kan zelf een algoritme bedenken, beschrijven en uitvoeren.	**1 Oriëntatie op jezelf en de wereld** De leerling leert oplossingen voor technische problemen te ontwerpen, deze uit te voeren en te evalueren.	**1 Kritisch denken**
2 De werking en het (creatieve) gebruik van digitale technologie DG3.2 Aansturen van en creatie met digitale technologie.	**2 ICT-basisvaardigheden** De leerling maakt gebruik van de functionaliteiten van verschillende apparaten, platforms en besturingssystemen.	**2 Rekenen** De leerling leert aanpakken bij het oplossen van reken-/wiskundeproblemen te onderbouwen en leert oplossingen te beoordelen.	**2 Probleem oplossen**

INTRODUCTIE

Openingsslide

Vertel de leerlingen dat we vandaag gaan leren wat computational thinking betekent, welke vaardigheden daarvoor nodig zijn en dat ze dit offline én online gaan ervaren. #Aandeslag!



Lesbrief opdracht 1:

Vraag aan de leerlingen welk cijfer (0 t/m 10) zij zichzelf nu geven wat betreft hun kennis over computational thinking. Wat weten zij er al van? Laat de leerlingen het cijfer opschrijven op de lesbrief bij opdracht 1. Op jouw signaal houden ze hun persoonlijke score omhoog. Bekijk en bespreek met elkaar de cijfers aan de hand van onderstaande vragen. Eventueel kunnen de leerlingen met een hoog cijfer iets van hun kennis toelichten aan de groep.

- Wie geeft zichzelf een laag cijfer (0 - 4) en waarom is dit?
- Wie geeft zichzelf een gemiddeld cijfer (5 - 7) en waarom is dit?
- Wie geeft zichzelf een hoog cijfer (8 - 10) en waarom is dit?
- Wat is het klassengemiddelde? Hoe berekenen we dit?
- Wat is er nodig om het cijfer na deze les omhoog te krijgen?

BEROEPENSLIDE/ARBEIDSMARKT

Slide 1, Praten met de klas

Vertel de leerlingen dat zij opgroeien in een wereld waarin computers, machines en apparaten onmisbaar zijn. In ziekenhuizen, bij grote techbedrijven zoals Google en Facebook, maar ook in de supermarkt om de hoek worden apparaten ingezet om onze maatschappij te dienen. Om deze apparatuur te kunnen begrijpen en bedienen, moeten we ook een beetje kunnen denken als een computer. Je kunt je dus voorstellen dat je in veel beroepen computational thinking kunt toepassen. Denk maar eens aan de programmeurs van Google die complexe programma's schrijven om hun software foutloos te laten werken. Wetenschappers die simulaties uitvoeren moeten ook algoritmisch kunnen denken en



Stichting FutureNL ontwikkelt onafhankelijk lesmateriaal. © 2021 Iedereen mag deze les downloaden, delen en uitprinten (alleen niet als je er geld mee wilt verdienen). Doen! Creative Commons licence: CC BY-NC-ND 4.0. Deze les is gemaakt door Frank Veenstra.

Dit document is gedownload door () op November 24, 2026.
Mag alleen gekopieerd worden voor gebruik binnen de school van 'gebruiker'. Zie ook de abonnementsvoorwaarden van Leapo.

architecten moeten hun ontwerpplan concreet kunnen maken in een 3D-programma zoals SketchUp. Als de kassa in je plaatselijke supermarkt niet werkt, zal de caissière eveneens in kleine stapjes het probleem moeten kunnen oplossen. Ook zij is dan bezig met computational thinking.

Slide 2, Praten met de klas



Vertel de leerlingen dat computational thinking een belangrijke vaardigheid is voor iedereen die in de 21e eeuw leeft, werkt en leert. Eigenlijk zou iedereen dan ook computational thinking moeten beheersen. Computational thinking gaat over leren programmeren en het kunnen denken als een computer. Echter, als wij konden denken als computers, dan hadden we de computer niet nodig gehad. Daarom is computational thinking meer dan denken als een computer en leren programmeren. Met computational thinking leer je vaardigheden waarmee je op een creatieve manier problemen kunt oplossen met behulp van technologie. Je leert analytisch te denken en handelingen op te delen in kleinere stapjes. Samen vormen deze stapjes een programma waarmee je het probleem kunt ontleden en tot de juiste oplossing kunt komen. Sla je een stapje over en klopt je programma niet, dan moet je gaan debuggen. Dat betekent dat je aan de slag gaat met het opsporen en verhelpen van fouten in je (computer) programma.

Slide 3, Praten met de klas



Vertel de leerlingen dat je voor computational thinking een aantal belangrijke vaardigheden nodig hebt. Computational thinking doe je namelijk niet zo 1, 2, 3. Vraag aan de leerlingen welke vaardigheden zij nodig denken te hebben en laat ze deze op de lesbrief, bij opdracht 2, in de wolken opschrijven. Wanneer ze goed hebben opgelet, hebben ze al een aantal voorbeelden gehoord in de vorige slide. Maar dit zijn ze nog niet allemaal!
Mogelijke antwoorden:

- Je moet goed nadenken over informatie.
- Je moet informatie in logische stukjes kunnen verdelen.
- Je moet informatie kunnen vereenvoudigen.
- Je moet informatie in een schema (of tekening) kunnen zetten.
- Je moet oplossingen kunnen bedenken en uitproberen.
- Je moet kunnen denken volgens een algoritme.
- Je moet de oplossing kunnen toepassen in vergelijkbare situaties.

★TIP!

Wil je als leerkracht jezelf meer verdiepen in computational thinking? Kijk dan eens op het vakportaal van SLO (https://www.slo.nl/vakportalen/vakportaal-digitale-geletterdheid). Daar vind je tips voor leerdoelen, leermiddelen en literatuur die je kunt raadplegen. Ook de andere onderdelen van digitale geletterdheid staan daar uitvoerig beschreven.

Stichting FutureNL ontwikkelt onafhankelijk lesmateriaal. © 2021 Iedereen mag deze les downloaden, delen en uitprinten (alleen niet als je er geld mee wilt verdienen). Doen! Creative Commons licence: CC BY-NC-ND 4.0. Deze les is gemaakt door Frank Veenstra.

Dit document is gedownload door () op November 24, 2026.
Mag alleen gekopieerd worden voor gebruik binnen de school van 'gebruiker'. Zie ook de abonnementsvoorwaarden van Leapo.

VERDIEPING

Slide 4, Praten en denken



⚖ Vertel de leerlingen dat zij opgroeien in een wereld waar computers, machines en apparaten onmisbaar zijn. Vooral de laatste jaren zijn er op technologisch gebied grote sprongen gemaakt. Veel apparatuur wordt steeds meer zelfdenkend en -sturend. Kijk maar eens naar dit filmpje uit 1998: https://www.youtube.com/watch?v=TNwhIHqM60g. Hierin wordt al snel duidelijk dat de positie van de mobiele telefoon enorm veranderd is, tegenwoordig kunnen we (bijna) niet meer zonder een mobiele telefoon. We bedienen onze lampen ermee, checken onze social media en houden contact via WhatsApp en beeldbellen. De mobiele telefoon is nu dus onmisbaar, maar wat gebeurt er als dit ook voor andere apparaten gaat gelden? Wat als apparaten ons werk gaan overnemen en steeds meer voor ons gaan denken? De apparaten krijgen daardoor een verantwoordelijkheid te dragen, waarvan we niet weten tot hoe ver we dit willen laten gaan. Het zou kunnen zijn dat we steeds meer de grip verliezen op deze apparaten en dat wij ons geen eigenaar meer voelen over onze eigendommen en ons eigen leven. Bespreek met de leerlingen dit ethische dilemma aan de hand van (enkele van) onderstaande vragen.

- Wat vinden zij van dit dilemma?
- Zijn zij zich bewust van de plek die technologie in hun leven inneemt?
- Hoe ver mogen apparaten (mee)denken over hun leven en wat mogen ze overnemen?
- Waar ligt voor de leerlingen de grens?
- Hoe gaan zij ervoor zorgen dat zij eigenaar blijven over hun eigen apparatuur en bijbehorende software?
- Is het zo erg om geen eigenaar meer te zijn over je eigen apparatuur en software?

DOEN

Slide 5, Luisteren



Bekijk met de leerlingen de video van SkillsDojo over programmeren (https://www.youtube.com/watch?v=fi5M4hKpcCg). Vertel de leerlingen dat programmeren en computational thinking erg veel op elkaar lijken. Veel mensen gebruiken beide termen dan ook vaak door elkaar, bij programmeren oefen je echter belangrijke vaardigheden die bij computational thinking horen. Programmeren wordt dan ook gezien als een belangrijke basisvaardigheid voor het uitvoeren van computational thinking. Zoals je in het filmpje kon zien, schrijf je als programmeur een instructie (script) met kleine stapjes om de computer (of ander apparaat) een handeling uit te laten voeren. De computer voert dit script vervolgens uit. Wanneer hier een fout in zit, voert de computer dit ook fout uit. Het is dus erg belangrijk om een goed script te kunnen schrijven. Er komen dan ook diverse belangrijke vaardigheden bij computational thinking kijken om de fout te debuggen en een goed script te schrijven!

Stichting FutureNL ontwikkelt onafhankelijk lesmateriaal. © 2021 Iedereen mag deze les downloaden, delen en uitprinten (alleen niet als je er geld mee wilt verdienen). Doen! Creative Commons licence: CC BY-NC-ND 4.0. Deze les is gemaakt door Frank Veenstra.

Dit document is gedownload door () op November 24, 2026.
Mag alleen gekopieerd worden voor gebruik binnen de school van 'gebruiker'. Zie ook de abonnementsvoorwaarden van Leapo.

Slide 6, Doen



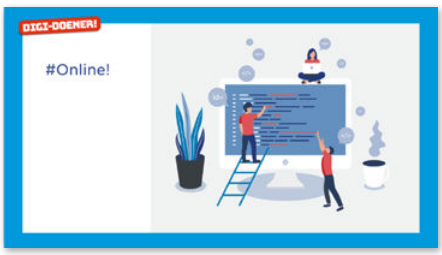
Vertel de leerlingen dat zij zichzelf offline gaan programmeren. In de vorige slides hebben ze geleerd waarom kunnen programmeren zo belangrijk is voor computational thinking. Ze gaan nu een eigen probleem verdelen in kleine stapjes en in een script verwerken. Ook gaan ze de scripts uitproberen op elkaar. Lukt het de leerlingen om het probleem op te lossen en te werken aan hun (offline) vaardigheden wat betreft computational thinking?

Om de leerlingen een voorbeeld te geven, gaan ze eerst kijken naar Dave die met zijn zoon aan de slag gaat met offline programmeren. In het filmpje zie je of het ze lukt om een juiste instructie te schrijven en een lastig probleem in kleine stukjes te verdelen. Kijk samen met de leerlingen naar het filmpje van Dave en zijn zoon: https://www.youtube.com/watch?v=JmCsuyuVvjQ&t=102s.

Vertel de leerlingen dat ze in het filmpje van Dave en zijn zoon zagen hoe moeilijk het is om een goed script te schrijven. Maar dat telt vast niet voor de leerlingen in deze klas! Gaan ze de uitdaging aan als programmeur en gaan ze voor eeuwige roem? Laten we starten!

- Vertel de leerlingen het probleem (een eigen probleem verzinnen mag ook!):
  Een tijdje geleden hebben jullie gehoord dat in de klas alle ramen open moeten blijven staan. Daardoor is er voldoende ventilatie in het klaslokaal en blijven we allemaal fit! Nu zijn er echter een aantal ouders die geklaagd hebben, want hun zoon of dochter heeft het koud! Na goed overleg is er besloten dat er in de klas voortaan een jas gedragen moet worden. Deze jas moet aangetrokken worden door een klasgenootje, zodat ze ook leren samenwerken. Dat is iets wat voor de directeur heel erg belangrijk is! Lukt het jullie om een script te schrijven waardoor alle leerlingen bij elkaar de jas kunnen aantrekken? Als het lukt, zal de directeur jullie zien als beste klas ooit! En wie wil dat nu niet? We gaan voor de eeuwige roem!
- Verdeel de klas in tweetallen (of drietallen bij een grote groep).
- Laat de leerlingen, op de lesbrief bij opdracht 3, het script schrijven waarmee ze het probleem laten oplossen door een medeleerling. Let op: ze mogen maximaal vijftien stappen gebruiken!
- Neem de lesbrief in en deel deze willekeurig in de klas weer uit.
- Laat de leerlingen nu de scripts uitvoeren.
- Is het ze gelukt om het probleem op te lossen via het script en heeft iedereen zijn jas aan?
- Bespreek de opdracht na met de leerlingen:
  * Welke scripts waren een succes en hoe kwam dit?
  * Welke scripts zijn niet gelukt?
  * Debug samen met de leerlingen één van de niet gelukte scripts.

Slide 7, Doen



Let op: maak voor deze slide een keuze! Kies je ervoor om deze opdracht aan te bieden ter verrijking op een ander moment? Of heb je gekozen voor een themamiddag en laat je de leerlingen de

Stichting FutureNL ontwikkelt onafhankelijk lesmateriaal. © 2021 Iedereen mag deze les downloaden, delen en uitprinten (alleen niet als je er geld mee wilt verdienen). Doen! Creative Commons licence: CC BY-NC-ND 4.0. Deze les is gemaakt door Frank Veenstra.

Dit document is gedownload door () op November 24, 2026.
Mag alleen gekopieerd worden voor gebruik binnen de school van 'gebruiker'. Zie ook de abonnementsvoorwaarden van Leapo.

opdracht nu doen? Dit is afhankelijk van de tijd die je hebt gepland voor deze les in je lesrooster.

Vertel de leerlingen dat ze bij de vorige opdracht offline gewerkt hebben aan hun vaardigheden wat betreft computational thinking. Nu is het tijd om dit online te gaan doen! Voor deze opdracht gaan ze hun eigen dancefeest programmeren binnen een uur. Deel de devices uit en laat de leerlingen naar https://code.org/dance surfen. Vanaf hier kunnen ze zelfstandig aan de slag nadat ze op de startknop hebben geklikt. Laat ze eerst de introductievideo bekijken, hun leeftijd opgeven in het volgende scherm en daarna starten met hun eerste instructie. Lukt het ze om hun dancefeest te organiseren na een uur programmeren? Wanneer ze het uur programmeren afgerond hebben, kunnen ze eventueel hun certificaat printen. Maak hier met elkaar afspraken over.

Let op: is het niet gelukt om het uur programmeren af te ronden? Laat de leerlingen dan noteren waar ze gebleven zijn, op een later tijdstip kunnen ze dan alsnog hun certificaat verdienen!

★TIP!

Gun jij je leerlingen na deze les nog een keer een uur programmeren, maar dan met een echte professional uit het bedrijfsleven? Via FutureNL kun je een gratis programmeerles aanvragen. De professional gaat met jou en je leerlingen aan de slag om ze nog enthousiaster te maken over programmeren. Nieuwsgierig geworden? Kijk dan eens op: www.futurenl.org/codeuur/.

AFRONDING

Slide 8, Praten met de klas



Vertel de leerlingen dat we de les over computational thinking gaan afronden. We hebben met elkaar gekeken naar wat computational thinking is, wat de rol van programmeren is bij computational thinking en we hebben offline en online ervaren hoe het is om een probleem op te lossen in kleine stukjes.

Vraag aan de leerlingen: als ze terugkijken naar het cijfer dat zij zichzelf hebben gegeven aan het begin, is dat nu na afloop van deze les veranderd? Is hun kennis verbeterd? Laat de leerlingen hun tweede cijfer opschrijven op de lesbrief en delen met de klas. Laat een aantal leerlingen die zichzelf een ander cijfer hebben gegeven, vertellen waarom ze hun cijfer hebben veranderd. Mogelijke vragen kunnen zijn:

- Waarom is het cijfer veranderd of hetzelfde gebleven?
- Wat heb je geleerd?
- Welke informatie was nieuw voor je?
- Wat wist je al?
- Is er nog informatie die je gemist hebt?

Stichting FutureNL ontwikkelt onafhankelijk lesmateriaal. © 2021 Iedereen mag deze les downloaden, delen en uitprinten (alleen niet als je er geld mee wilt verdienen). Doen! Creative Commons licence: CC BY-NC-ND 4.0. Deze les is gemaakt door Frank Veenstra.