

Interview Han van der Maas



'Voor mij is programmeren een essentiële tool, zoals hamer en spijkers voor een timmerman'

Werk Hoogleraar en hoofd van de groep Psychologische Methodenleer van de Universiteit van Amsterdam | Leeftijd 48
Leerde programmeren Als 3^e jaars student psychologie
Eerste programmeertaal Pascal

Hoe ben je zelf geïnteresseerd geraakt in programmeren?

Pas op de universiteit, in mijn 3^e jaar van de studie psychologie volgde ik een cursus Pascal. Ik vond het meteen superleuk en had onmiddellijk door dat dit voor mijn studie enorm belangrijk was; ik programmeerde er psychologische experimenten mee. Voor mij is programmeren een essentiële tool, zoals hamer en spijkers voor een timmerman.

Waarom is programmeren belangrijk?

Het is bijna een open deur. Er is een enorme maatschappelijk-economische vraag naar mensen die informatica echt in de vingers hebben. Die creatief met ict kunnen omgaan, op de hoogte zijn van de nieuwste technieken en zelf kunnen programmeren. Veel bedrijven – maar ook de kunsten en de wetenschap – zitten te springen om mensen die niet louter computerconsument zijn, maar die computers onder controle hebben. Zulke mensen zijn nu heel moeilijk te vinden. Daarom vind ik dat alle kinderen moeten leren programmeren.

Waarom hoort programmeren thuis op school?

Programmeren is een expertise, net als pianospelen, schaken, rekenen en taal. Het is moeilijk, het duurt lang voor je er succes mee hebt en je leert het met vallen en opstaan. Niemand leert goed programmeren zonder af en toe eens met z'n muis te gooien van frustratie. Het kost veel tijd en oefening voor je het zo goed in de vingers hebt dat je er echt iets creatiefs mee kunt doen.

Natuurlijk kan iedereen met bijvoorbeeld Scratch al snel leuke en originele dingen maken. Dat werkt motiverend. Maar om na het leuke begin vol te houden met leren om programmeren echt in de vingers te krijgen, dat kost moeite en doorzettingsvermogen.



Leren programmeren moet je
niet aan het hobbycircuit overlaten



Programmeren zou vanaf de brugklas structureel op het rooster moeten staan

Nerds zoals ik nemen vaak aan dat iedereen – net als wij – programmeren meteen leuk vindt en er uit zichzelf veel tijd in steekt. Dat is natuurlijk een denkfout. Een paar procent van de kinderen vindt programmeren zo superleuk, dat ze het zelf via internet of via clubjes leren. Maar de meesten kiezen liever wat anders om te doen in hun vrije tijd. Dat geldt al helemaal voor meisjes. Het is mij nog niet gelukt om mijn dochters serieus aan het programmeren te krijgen; terwijl mijn zoon er zo goed in is dat hij er een vakantiebaantje aan overgehouden heeft. Dus zelfs in mijn eigen gezin dat behoorlijk informatica-minded is, blijkt het erg lastig om tegenwicht te bieden aan de tweedeling jongens – meisjes in de maatschappij.

Leren programmeren moet je daarom niet aan het hobbycircuit overlaten, dat is een taak van het onderwijs. Zeker op de middelbare school moet het een serieus vak worden. Ook rekenen en wiskunde zouden de meeste kinderen niet uit zichzelf leren als het geen verplicht schoolvak was. Voor programmeren geldt hetzelfde.

Vanaf welke leeftijd moet programmeren op het rooster?

Wetenschappers zijn het er niet over eens op welke leeftijd kinderen over alle cognitieve vaardigheden beschikken om te leren programmeren. In de praktijk blijkt dat de meeste kinderen van 7 jaar het prima aankunnen om kennis te maken met bijvoorbeeld de app Kodable. Ik vind dan ook dat al op de basisschool aandacht besteed moet worden aan programmeren. Dat hoeft niet wekelijks op het rooster, dat kan heel goed in projecten. Elk kind dat groep 8 verlaat moet naar mijn mening minstens twee keer een project hebben gedaan met programmeren (bijvoorbeeld een weekje Scratchen).

Op de middelbare school hoort informatica wat mij betreft in de top-4 van belangrijke vakken – naast Nederlands, Engels en wiskunde. Vanaf de brugklas zou het structureel op het rooster moeten staan, drie uur per week. En er moet een centraal indexamen voor komen.



Door het opbouwen van een expertise leer je dat inspanning loont

Net als bij wiskunde zou er een A- en een B-variant van het vak kunnen komen. Natuurlijk met aandacht voor het creatieve, apps maken bijvoorbeeld, maar iedereen moet ook de basis van een of twee programmeertalen leren.

Als er niet genoeg ruimte is in het rooster, moet misschien een ander vak plaatsmaken. Waarom is het eigenlijk zo'n automatisme dat vwo-leerlingen die er een beetje bovenuit steken, Grieks en Latijn gaan doen? Waarom niet informatica?

Helpt programmeren kinderen met hun andere schoolwerk?

Vaak wordt geopperd dat kinderen door te leren programmeren allerlei cognitieve vaardigheden aanleren, zoals gestructureerd denken, concentratie en doorzettingsvermogen. De vaardigheden die kinderen ontwikkelen bij het leren programmeren, zouden dan doorwerken op andere leerstofgebieden.

Uit wetenschappelijk onderzoek blijkt dat dat niet vanzelf gaat. Je moet kinderen de overeenkomsten laten zien tussen wat ze geleerd hebben bij programmeren en wat nu aan de orde is bij bijvoorbeeld natuurkunde. Het is niet zo dat leren programmeren (of leren schaken) automatisch fantastische effecten heeft, die impliciet en onbewust doorwerken naar andere gebieden.

Dat geldt trouwens voor alle expertises: dat je een top-violist bent, zegt bar weinig over hoe je kunt fluitspelen of schaken. Wel leer je door het opbouwen van één expertise dat inspanning kan lonen. Hierdoor is wellicht de motivatie beter op te brengen om veel te oefenen om een volgende expertise te verwerven.





Kun je niet beter meteen een serieuze programmeertaal leren op latere leeftijd?

Ik vind dat alle studenten moeten kunnen programmeren. Zelf geef ik les aan psychologiestudenten in de moderne programmeertaal R. Voor veel van hen is dit de eerste kennismaking met programmeren, het is daardoor een pittig vak. Ze moeten tegelijk met de syntax van R ook de basisprogrammeerconcepten leren begrijpen, zoals de verschillende soorten loops (for, while, repeat). Dat is best lastig. Als een stukje code niet blijkt te werken, kan het bijvoorbeeld zijn dat ze het concept 'loop' niet goed begrijpen,

dat ze de syntax van R niet goed toegepast hebben, of dat ze gewoon een stomme typefout hebben gemaakt. Dat loopt dan allemaal door elkaar en ze weten zelf vaak niet waar de fout zit.

Ik denk dat het enorm helpt bij het aanleren van een serieuze programmeertaal als je al eerder een simpelere taal hebt leren gebruiken. Dan begrijp je de concepten al en kun je je concentreren op de syntax van de nieuwe taal. Want elke taal heeft zo zijn eigen regeltjes en gekkigheden. Kinderen die nu op school leren programmeren, hebben later op het vervolgonderwijs een grote voorsprong.

