

如何让儿童更好地学编程？*



——佩珀特《儿童机器：重新思考计算机时代的学校》一书综述

纪九梅 汪 琼

(北京大学 教育学院, 北京 100871)

摘要: 佩珀特在《儿童机器：重新思考计算机时代的学校》一书中提到“学习能力”是面向未来最重要的能力，学习编程则是能够有效提升学习能力的途径。让儿童学编程的本质是锻炼儿童的学习能力，而为了让儿童更好地学编程，佩珀特认为应开发适合儿童的“自然”编程语言，并借助儿童喜爱的“对象”将编程延伸至电脑屏幕之外，还需秉持“助学法”、“建造主义”等教育哲学理念以从学习理论层面指导儿童编程教育。佩珀特指出，当关于学习的理念发生改变，形成基于计算机的学习文化时，“儿童学编程”将成为一件再自然不过的事，而这离不开学校、教师、教育研究者与家长的共同努力。

关键词: 儿童编程；编程语言；“助学法”；“建造主义”；“微世界”

【中图分类号】G40-057 【文献标识码】A 【论文编号】1009—8097(2021)02—0114—08 【DOI】10.3969/j.issn.1009-8097.2021.02.016

引言

西摩·佩珀特(Seymour Papert)是推动儿童编程的先驱，而《儿童机器：重新思考计算机时代的学校》(*The Children's Machine: Rethinking School in the Age of the Computer*)^[1](下文简称《儿童机器》)一书是他在观察和研究“计算机进入学校”之后撰写的第二本有关“儿童与计算机”的论著。在这本书中，佩珀特不仅批判了学校所推崇的“学习方法”，还质疑了学校中既定的“学习内容”；他认为面对无限的知识爆炸，应该在有限的时间内学习真正有价值的知识，即学会学习，不断提升学习的能力。佩珀特发现，计算机具备改变人类思维方式、提升学习能力的潜质，可以产生变革教育的作用，而“编程”是计算机提供的变革途径之一。佩珀特认为儿童对计算机有天然的热情和认同感，既是最适合学习新技术的群体，也是受益于新技术最多的群体，因此应该让“儿童对计算机编程”(Child Programs the Computer)。

在佩珀特看来，学编程其实就是学习用人和计算机都能听得懂的语言来沟通，编程语言虽然是人工造的，却应该尽可能自然。想要让儿童更好地学习编程，首先要在编程语言的开发方面下功夫，设计更适合儿童的编程语言。学编程不只是让儿童在屏幕前敲代码，还可以与能让儿童动手操作的“对象”结合起来，这不仅会唤起更多的情感因素，还能将多学科的知识融入其中。所谓儿童对计算机编程，就是儿童教计算机学习新的命令，儿童能够站在不同于以往的角度对学习进行思考，从而掌握“助学法”(Mathetic)的关键原则。在建构主义(Constructivism)的基础之上，佩珀特发展出“建造主义”(Constructionism)，从教育哲学的理论高度指导儿童如何学习编程。儿童学习编程呼唤新型学习文化的生成，这种文化应该是助学的、多元化的，应有助于计算机发挥其革命性的力量，让教育产生质的变化。

一 为儿童设计适合的编程语言

1 将编程语言通俗化，降低学习门槛

不仅大众对儿童究竟能不能够学会编程这个问题存在疑问，当时从事发展心理学研究的学

者也怀疑那些还未达到“形式思维阶段”的儿童是否具备这样的能力，佩珀特认为对这个问题的回答取决于如何看待“编程”的含义。

佩珀特认为“编程”不应该是某些科技从业者的特权，计算机语言不仅要让普通人能用，更要让儿童能用，这背后存在将计算机语言通俗化的需求。学编程并不是为了学习某种具体的计算机编程语言，而是为了学习编程背后所蕴含的问题解决方法，如“模块化编程”、“漏洞调试”等^[2]，这些学习经历能让儿童改变对学习的看法、对知识的看法和对自己的看法。更为重要的是，儿童以后无论做什么事情，其曾经在学习编程时所习得的技能和养成的思维都可能会自然、流利地得到使用，这是具有计算机素养¹的体现，会让儿童受益终生。佩珀特表示，如果真想用编程改变儿童的学习、让儿童以充满乐趣的方式进行探索，那么与其试图讲授现有的编程语言，不如尝试开发一种更有可能满足儿童需求与能力的编程语言。因此，他所在的麻省理工学院与教育技术研究公司 Bolt, Beranek & Newman 成立了开发 LOGO 语言的专门团队，此语言的出现与发展降低了学习编程的年龄门槛。

2 LOGO 语言与“海龟”微世界是可以参照的范本

LOGO 语言的灵感来源于佩珀特的一次随手涂鸦，他发现像画画这样直观的事情更能唤起低龄儿童的兴趣，他们可以使用 LOGO 画他们想要的任何东西。在 LOGO 中有“海龟”这样一个可以辅助儿童思考的对象 (Object-to-think-with)：最初的海龟是实体的，编程时用计算机控制海龟移动，还可以画下海龟运动的轨迹；后来，“实体海龟”变成了屏幕上的“光速海龟”，能够瞬间绘制出儿童感兴趣的路径或形状。

在微世界中，儿童需要让自己随着“海龟”移动，学会通过身体发现更多的数学知识，并“使用”数学知识来指挥“海龟”画画。微世界的控制指令要与儿童的口语学习风格相匹配，让他们以一种非正式的方式学习和使用数学，从而为学习风格的过渡提供缓冲。使用数学知识来画画比单纯地学数学知识更易让儿童喜爱 (体现“热爱”)，他们要自己动脑筋用海龟画出任何想要的内容 (体现“掌控感”)，且认为自己做的事是真正有意义、有价值的 (体现“智力认同”)，这足以改变儿童与数学的关系，进而还能改变儿童与学习的关系——这种活动与学校数学课上教师创设某个情境来“诱导”学生做计算有本质的不同：儿童在做自己想做的事情。

佩珀特认为，LOGO 与微世界的构建是他对改善儿童认知结构做出的一种有效尝试，但不是唯一方式，他期待未来能有更多的人在 LOGO 的影响下探索出更多适合儿童的方法。总之，让儿童学编程的前提是：要为儿童设计、创造适合其能力和需求的编程语言，最好能够根据成功学习的三要素 (即热爱、掌控感、智力认同) 为儿童创造出一个能够自由探索的微世界。

二 拓宽儿童编程的学习范围

1 儿童编程要充分调动身体与情感参与

为了不让计算机仅仅充当强化练习学科知识的“帮凶”，佩珀特想到了让儿童学习编程：他最初的设想和实践是让初中生编写初级的人工智能程序，通过建立人工智能的心智模型，促使学生反思人是如何学习和思考的，进而更好地理解学习；后来，他发现通过编程不光可以模仿人类智能——这对于低龄儿童来说有些抽象，也可以模仿动物智能 (如能够找到妈妈的乐高

¹ 佩珀特将“计算机素养”定义为“从计算机文化中获得认知”，认为素养更多地代表着一种看待世界的方式。

小猫咪)。动物智能虽然没有人类智能高阶,但允许儿童有更多的身体参与,且更容易唤起儿童的情感参与,故同样可以让儿童受益且更适合儿童。

佩珀特用“控制论”(Cybernetics)²来指代这一被拓宽的编程学习领域,这也直接促成了乐高编程机器人的诞生。儿童可以用乐高积木和传感器建造一个具有智慧的“生命体”,并通过编程赋予它们一定的生物特性。这种活动有足够的开放性,可以调动儿童更多的想象、创造和情感参与。当儿童用LEGO-LOGO套件自由构造任何他们想做的东西时,他们会将自己代入其中,在微世界操控程序探索更复杂的学习,此时知识转化成了“能被使用的知识”而具有了价值。佩珀特认为,不要总是以一种教育者的姿态试图让儿童去做点什么,而要允许儿童通过控制论编程的方式去表达个人的想法和学习风格。这也意味着“控制论”对于多元学习风格有更高的宽容度,因为它不主张权威、单一的认识论,而是鼓励多角度思考、支持多元思维模式。

2 “控制论”中有益于提升学习能力的思维模式

“控制论”蕴含一些有价值的思维模式,能推动儿童的思维发生改变,从而提升学习能力:

①拼装修补(Bricolage)³,意味着灵活利用手头可用的东西来建造或创造。儿童在编程和动手搭建的过程中可能会遇到各种问题,能够运用手头的工具、已知的方法去解决新情境中出现的问题,其实就是在拼装修补。佩珀特将“拼装修补”定义为一种协商式的工作模式,在这种动手操作的、具象思维的指导下,利用手头的工具去解决问题就不会执着于寻求某种通用的方法,如此儿童就能明白通用的方法并不一定是唯一的、最好的,而只是可选工具中的一项;当手头的工具经过多次实践的打磨,之后儿童在解决问题时就会愈发得心应手。

②模糊正确(Vaguely Right),指的是利用当前的知识和资源,朝着模糊正确的方向前进,在这个过程中更易处理各种不确定性。与不够灵活的精确编程所造成的“精确错误”(Precisely Wrong)相比,这种逐步尝试、在反馈中摸索的方法反而更能达到目标,因此“模糊正确”也被称为“引航式”(Pilotage)编程方式。照此方式学习的儿童,不会局限于过去被反复强化的思想,不会认为“权威就是正确”、“答案和解法是唯一的”,而是会发现部分的、定性的知识也是好的知识,这更利于发展其科学思维。

③反馈修正,是指能够根据反馈进行调整。对于那些无法完全精准预测的事件来说,“控制论”有很强的适应能力。儿童对带有传感器的对象进行编程,传感器反馈带来的生命特性不仅能激发儿童的兴趣,还能调动其身体的参与,并有助于儿童理解反馈对于系统性的工作方式来说有何作用。对于整个系统的运作来说,实际上并不需要每个部分都完全正确,根据得到的信息进行反馈修正也能从整体上让系统更好地工作,让系统面对意外情况时能保持良好的适应性。

三 从学习理论层面对儿童学编程进行指导

1 “助学法”与“建造主义”指导以儿童为中心的编程学习

(1) 助学法

佩珀特认为现有的学习理论都是基于对传统的学校教育情境进行研究之后提出的,对于编

² 控制论是一门研究生命体、机器、组织的内部或彼此之间控制和通信的科学。

³ “Bricolage”一词来源于人类学家 Claude Levi-Strauss 于 1966 年出版的著作《野蛮人的头脑》(The Savage Mind),这是一个法语单词,是指“原始”社会使用手头工具摆弄的“具体科学”,此处翻译为“拼装修补”。

程学习这种未来的、新的学习方式来说,传统的学习理论可能并不适用。他发现在教育领域中有“教学法”(Pedagogy)一词,却没有与“助学法”对应的单词,这其实体现了英语文化对教学和学习这两种理念重视程度的不同,于是他造了“Mathetic”这个词。佩珀特认为“让儿童对计算机编程”是儿童真正主导的学习活动,而想要由关注教师的教转为关注学生的学,就需要对如何学习这一问题进行更多的探讨,专业术语则是讨论的前提。佩珀特认为“授人以鱼不如授人以渔”,在这个隐喻中,知识代表鱼,助学法是钓鱼的知识,计算机是优质的钓鱼工具,“微世界”则是蕴含很多鱼的水域,所以提供给儿童“微世界”是改善学习的更好方法。在LOGO编程中,可以充分实践“助学法”的原则,如在编程时将新问题与已经解决的旧问题联系起来、将复杂的问题分解成简单的子问题、将自己代入海龟唤起身体共鸣等。海龟几何成为了练习“助学法”策略的载体,有助于引导儿童边做边思考,并将其迁移到其它学习中去。

(2) 建造主义

与关注教学、认为更好的学习取决于教学改进的“教授主义”(Instructionism)不同,佩珀特提出的“建造主义”(Constructionism)是关注学习的,认为应该以最少的教学实现更多、更好的学习。佩珀特的建造主义是在皮亚杰的建构主义(Constructivism)基础之上发展的,建构主义更侧重于认知和精神层面,是大脑内的活动,认为知识的掌握意味着大脑将接受的信息建构成了个人版本;而建造主义包含更多身体参与、感觉和情感层面的因素,认为大脑内的建构活动需要现实世界中能够参与进去的活动来支持,因为建造是需要材料的。在建造主义的指导下,儿童在编程时要动手搭建,通过制造自己感兴趣的事物来编程;另外,儿童也要注重编程之后进行作品的分享与交流,这种“社交性”能够调动更多的情感参与,这是建造主义支持知识建构的特点之一。

2 儿童编程不应落入传统学校教育的窠臼

让儿童学编程不是在学校中开设编程课程这么简单,学校规定课程计划这种做法背后的知识观本身就存在问题:将编程知识划分成碎片,希望按照计划将这些知识存储进儿童的脑子里,而这些知识是死记硬背得来的,属于很难与其它知识建立联系的分离性知识。建造主义认为知识与知识之间并不存在严格的界限,也不存在明确的上下级关系,正因为如此,学生才能有一定的自主权,真正有价值的知识是那些可以灵活运用知识。

佩珀特秉持的建造主义重申了具象思维(Concrete)的重要性⁴,虽然学校更强调抽象思维的发展,但这并不意味着抽象思维比具象思维更优异。虽然皮亚杰认为学龄前儿童处于具象思维阶段,但这种具象思维并非只有思维未成熟的人才用,相反一些高手、专家、具有创造力的人也会用这种思维方式来指导工作、学习。对于儿童来说,更自然的具象学习方式应得到进一步的加强和延伸,如在LOGO编程环境中海龟的具体形象就能积极调动儿童的身体参与,让他们以更自然、更适合的方式学习。

3 从编程学习故事中总结出的学习建议

在《儿童机器》一书中,佩珀特记录了几个与儿童学编程有关的学习故事,这些学习活动都在“助学法”与“建造主义”的指导下开展。佩珀特不仅从实践层面证明了儿童能够通过学编程来促进学习,还从理论角度总结了三条有利于学习的建议:

⁴ 佩珀特认为皮亚杰最重要的贡献不是划分思维的四个发展阶段,而是对具象思维进行了详细的描述。

①花时间。很多学习上的不成功，都是由于没有真正花时间造成的。想要让儿童从计算机中真正受益，就需要有足够长的时间、足够多的机会让儿童自由接触计算机，用编程语言与计算机对话。就像“数学大陆”这个隐喻所说，生活在“说数学话”的大陆上可以自然习得数学，如果将编程固定在课程表的某个时间段内，就仅仅是让儿童在“数学大陆”上短暂地度个假，其学习效果一定会大打折扣。给学生充分的时间学习说起来简单，但在学校里却很难实现，因为每节课都有固定的时长，在被分成片段的课程表中，学生很难做到真正的时间充裕。

②用讨论促进学习。儿童在微世界中通过编程来画画，可以就自己的作品与他人充分进行讨论和分享。在用海龟编程作画的过程中，有的儿童喜欢用线条勾勒出轮廓，有的儿童钟情于弥散抽象的风格——从浅层来看，这只是作画方式的不同；但从深层来看，这却是习惯性的思维模式与学习风格的不同。儿童可以通过讨论发现彼此的异同，不仅可以加深对其它学习风格的理解和尊重，还有助于增强对自身学习风格的自信与认同。很多人因为害怕暴露自己不好的一面，而不愿意在别人面前讨论自己不懂、不会的事情，但自由讨论的学习经历实际上更容易让人从学习障碍中走出来。

③寻找联系。学习不光要找到自己感兴趣的点，还要寻找各知识点之间的逻辑和关系。学习环境的连通性有助于实现知识之间的相互关联、传播和迁移，而要将“学习”和“学习环境”联系起来，就要在环境的改变上下功夫，并不只是去改变环境中的个体。微世界就是这样一个充满丰富联系的场域，如用海龟画圆可以让儿童将高深的微积分知识与自己身体“转一点、前进一点”这样朴素的感知建立联系。佩珀特认为，当不同领域之间建立联系的时候，学习的过程倒不完全是在构建，而更像是在种植，知识在潜移默化之中也得到了发展。

佩珀特认为，局部的改善和个人的努力无法真正对教育产生变革性的作用，学校中的学习方式从来都不是唯一的，甚至都不能算是成功的；而学校的地位之所以难以撼动，是因为教育传统与教育理念过于牢固。为了让儿童能够更好地学习，需要发展一种全新的学习观、认识观和教育观——也就是说，要从教育哲学的理念层面来指导儿童学编程：关注“助学法”是为了让学习者真正处于主动地位，对改善学习方法保持关注；“建造主义”不仅包含智力因素，还包含感觉、情感等因素，强调具象思维的力量；更好的学习是需要花时间的，讨论也可以促进学习，并且还要积极寻找联系、重点发挥学习环境的连通性。

四 儿童学编程与新型学习文化的形成

佩珀特深知，“计算机能够做什么”与“计算机允许被做什么”之间存在巨大的鸿沟。即使是在计算机随处可见的今天，每个儿童接触计算机的方式、用计算机做的事情仍然千差万别，而这种分化很可能造成将来社会角色的进一步分化。让儿童学编程是为了让儿童真正用计算机做有价值的事情，从而让计算机发挥积极、正面的作用。儿童学编程不应该被看作是脱离教育实际的“空中楼阁”，也不是制造教育焦虑的“新赛道”，更不是所谓少数具有编程天赋者的“特权”。佩珀特认为计算机教育的发展不能被简单地理解为“一个又一个该死的产品”，其真正重要而深远的影响是播撒下有教育意义的计算机文化种子，让儿童学编程背后的文化得以成长——只有理解和培育这种文化，计算机才能在将来对教育产生建设性的影响。这不仅需要从从事计算机编程教育研究的工作人员做出努力，学校管理者、教师、家长乃至整个社会都应该做出相应的转变，共同营造助学的氛围，才能真正支持儿童学好编程。

1 学校：助学的“小学校”模式更有利于学习

佩珀特^[3]在《头脑风暴：儿童、计算机和有利的想法》一书中曾激进地批判“学校教育”，也因此引发了许多争议。经过一段时间的实践，他反思自己应该以“发展的眼光看待学校”——实际上，他本人并不反对“学校教育”，他反对的仅仅是学校中不利于学习的“那一种特定的教育”。《儿童机器》一书对学校进行了更为详细、透彻的分析，他认为只有深入了解传统学校模式的不合理之处，才能更好地改进教育。

佩珀特用“同化”（Assimilation）和“顺应”（Accommodation）⁵来描述计算机进入学校的过程：计算机本来是具有颠覆和变革作用的工具，但进入学校之后反而被同化为固化学校既有做法的工具。当时有人质疑让计算机进入学校的做法，认为计算机对教育来说是投入多但收益少的事情，佩珀特认为当时看不到成效是可以解释的：从表面来看是量的问题，生机比远远不够；而深层原因是计算机在学校中究竟被如何使用的问题。计算机辅助教学（Computer Aided Instruction, CAI）是同化的表现之一，这种让计算机对儿童编程的做法背后是错误的学习理念，而以 LOGO 为代表的儿童对计算机编程的进步主义教育技术（Progressive Educational Technology, PET）是顺应的表现——前者对教育的改变是部分的，而后者是对学校的本质发起了挑战。回顾计算机进入学校的历史，不难推理出当下编程教育进入学校大概率也要经历从同化到顺应的过程。最初的同化可能收效甚微，但仍是一种有益的尝试，更重要的是如何更快地进入顺应的过程，让编程教育发挥出改变思维、改变学习的力量。

佩珀特认为编程带给教育的变革不是一蹴而就的，因此应该允许学校有过渡和妥协模式的存在，但如果传统学校固步自封，不能适应未来的变化，那么将来可能会被“小学校”模式替代。所谓“小学校”，是一个由怀着相似知识观、学习观、教育观的教师、学生、家长自愿进入的学习共同体，它可以是实体的也可以是虚拟的。传统学校不再是学习的唯一选择，多种多样的“小学校”成为了学习方法进化的最佳实验室，所有人都可以自由加入“小学校”学习感兴趣的内容，而不必被强加任何思维模式和学习内容。

2 教师：应该成为编程教育中的共同学习者

佩珀特认为，教师应该成为计算机变革教育过程中的重要角色。但是，传统学校中的教师通常很难摆脱学校的束缚，以至于很多教师并不能算是真正的“教师”，而只能算是教学生如何应对标准化考试的“技师”；有的教师虽然心怀“诗和远方”，却不得不在岗位上应对眼前的“苟且”，这使得有些人对教师这一群体有负面印象，认为教师缺乏创新与变革的能力。造成传统学校中的教师不受欢迎的主要原因，在于学校强加给了教师许多角色。佩珀特认为要相信教师有足够的潜力，并给予教师充分学习和发展的机会。

佩珀特指出，通常情况下对教师的培训是将教师作为被动的接受者，让他们被动学习一些他们并不太感兴趣的技术和技巧。而要想帮助教师更好地成长，让教师真正获得发展的机会，首先要转变观念，将教师视作学习者，甚至更重要的是教师自己要把自己当成学习者。佩珀特曾为教师举办过 LOGO 工作坊，当他让教师用 LOGO 画画的时候，教师常常谈论他们的学生会多么喜欢这一切，这恰恰说明教师受自己身份和角色的影响而没有全身心地投入和体验，没有

⁵ 皮亚杰提出同化、顺应的概念，前者表示个体改变对世界的表征，以适应自身的思考方式；后者则表示个体改变自己的思考方式，以适应世界。佩珀特认为，同化和顺应的概念可以套用于计算机进入学校的过程中。

让自己成为一个真正的学习者，所以他们使用 LOGO 时想的都是如何把这些带给学生。这样一来，教师就没办法在这个过程中切实感受计算机与编程怎样改变学习，因而也无法将这种感受传递给学生。

在编程教育中，教师一定不要被自身角色所束缚，要拒绝“计算机教师”这种狭隘的标签，而是要成为编程学习中的共同学习者。这是因为，学习本身是一件具有吸引力的事情，教师作为水平高一些的学习者，应该与学生共同参与，而不是置身事外。在这个共同参与的过程中，教师不要害怕失去权威，要接受学生可能比自己更了解计算机、更懂编程；教师也不必假装什么都懂，遇到不会的问题，可以和学生一起探讨如何解决。在学习共同体中，教师只不过是其中水平较高的学习者，有时候承认自己的“无知”也是一种智慧，这不仅能刺激学生自主探索、发现解决问题的方法，还能从本质上改变师生之间的关系，营造积极、开放、包容的学习氛围。

3 教育研究者：鼓励以质性的视角从微观教育层面开展研究

佩珀特认为教育研究不必为了追求客观原则而与教育的对象保持过多的距离，不能脱离学生、教师来研究教育；所谓向科学范式靠近的控制变量的教育实验只能帮助改良学习的某个小方面，并不能起到深层变革的作用。况且真正有意义的事情不一定能通过简单的测量反映出来，就像佩珀特所记录的学习故事那样，儿童在学习 LOGO 的过程中发生的很多改变都是难以测量的。佩珀特认为作为教育研究者应该重视直觉和理解，这些经验能够让他们透过现象看到本质，发现究竟是什么因素真正在学习中发挥了作用，最好能够像一个优秀的人类学家那样，以质性的方法从生动的学习故事中捕捉成功学习的要素，而不是用冷冰冰的数字去测量、去评价。

佩珀特将对微世界的研究划归为“微观教育学”，认为这是值得关注的研究领域；他希望那些愿意在教育微世界中做出贡献的研究者，能够从教育研究的边缘逐渐走向中心位。微观教育研究虽然旨在发展和研究特定的微观世界，但其干预策略应为宏观教育服务。传统的教育研究往往更关注如何“正确使用”，而微观教育研究倡导关注“可能的使用”，这种关注焦点的转移更有利于实现教育的创新。佩珀特鼓励有创造力的教育研究者保持积极的心态，勇于将想法付诸实践，以研究促产品，为编程学习提供更多优质的选项。

4 家长：需要对编程抱有支持理解的态度进行理性选择

虽然《儿童机器》一书对于家长作用的论述并不多，但是父母在儿童学习中扮演的角色不容小觑，尤其是对于计算机的学习来说，父母的支持理解很重要。编程教育需要调动各方积极参与，以获得更好的文化共鸣^[4]，这里面就包括父母的积极参与。佩珀特认为父母大多倾向于相信学习编程能促进数学的学习，因为人们普遍认为，计算机是和数学有关系的。但是，儿童可能会比他们的父母更加熟悉和热爱计算机，家长面对这种热爱难免会产生忧虑，有的家长会觉得被疏远、被排斥，还有的家长甚至认为这种人造世界会削弱童年的纯真与自然性，从而对计算机和编程学习持消极态度。

佩珀特解释道，儿童本来具有独立行动的本能，是愿意主动学习的，但由于没有可以借助的工具，只能一次次被学习的依赖性所挫败，如不得不通过向成年人提问来获得解答。当他们发现计算机这种工具可以帮助自己实现学习的自由、自己可以通过探索来解答疑惑，便会牢牢抓住它。家长应该明智地加入用计算机来学习的队伍中，如果可能，也可以和孩子共同学习编程。当父母对儿童正在用计算机所做的事情有了更多的了解和认识，对于儿童学习编程的信心也会增强。但是，面对鱼龙混杂的儿童编程教育行业，家长也应理性选择产品，因为有些产品

只是迎合了软件行业赚钱的模式：成本低廉，且容易营销，却并不一定适用于教育。如果家长不了解什么是真正有益处的编程教育，就很容易被各种营销手段所左右，不仅会抑制编程教育产品本应发挥的积极作用，还会助长错误的学习方法。

佩珀特从来都不是单纯为了让儿童学编程而学编程，学编程的目的是让儿童能够重新对学习、对知识、对自己进行思考。更进一步说，佩珀特希望改变的是关于学习的理念，形成基于计算机的学习文化，到时“儿童学编程”将成为一件再自然不过的事。正如佩珀特所言：“当第一的方法不是迎头赶上，而是在新的方向上领先。”而学编程正是一个可以获得领先机会的方向，这不仅是属于儿童的机会，也是属于每个人的机会，更是进行教育变革的机会。

参考文献

- [1] Papert S. The children's machine: Rethinking school in the age of the computer[M]. New York: Basic Book, 1993.
[2] 汪琼. 儿童为什么学编程?——Mindstorms: Children, computers, and powerful ideas 一书的启示[J]. 现代教育技术, 2020, (2): 122-126.
[3] Papert S. Mindstorms: Children, computers and powerful ideas[M]. New York: Basic Books, 1980.
[4] Papert S. The connected family: Bridging the digital generation gap[M]. New York: Taylor Trade Publishing, 1996.

How to Make Children Learn Programming Better?

——Review of Papert's Book *The Children's Machine: Rethinking School in the Age of the Computer*

JI Jiu-mei WANG Qiong

(Graduate School of Education, Peking University, Beijing, China 100871)

Abstract: In the book named *The Children's Machine: Rethinking School in the Age of the Computer*, Papert mentioned that "learning ability" was the most important ability in the future, and learning programming was a way to effectively improve learning ability. The essence of letting children learn programming was to exercise children's learning ability. In order to make children learn programming better, Papert held that a natural programming language suitable for children should be developed, and programming should be extended beyond the computer screen using children's favorite "objects". In addition, it was also necessary to uphold the educational philosophies of "Mathetic" and "Constructionism" to guide children's programming education from the level of learning theory. Papert pointed out that when the concept of learning changed and the computer-based learning culture was formed, children's learning programming will become a natural thing, and this was inseparable from the joint effort of schools, teachers, educational researchers and parents.

Keywords: child programming; programming language; "Mathetic"; "Constructionism"; "Microworld"

*基金项目：本文受教育部-中国移动科研基金 2018 年度项目“中小学编程教育与人工智能工程素养研究”（项目编号：MCM20180611）资助。

作者简介：纪九梅，硕士，研究方向为编程教育、MOOC 学习、人工智能教育政策，邮箱为 jijiumei@pku.edu.cn。

收稿日期：2020 年 7 月 20 日

编辑：小米

125