Paradigm Academic Press
Research and Advances in Education
ISSN 2788-7057
OCT. 2024 VOL.3, NO.10

The Impact of Children's Programming Education on the Development of Children's Cognitive Abilities

Xinsheng Liang[1]

[1] Second Needle Tick (Beijing) Network Technology Co., Ltd., China

Correspondence: Xinsheng Liang, Second Needle Tick (Beijing) Network Technology Co., Ltd., China.

doi:10.56397/RAE.2024.10.03

Abstract

This paper delves into the impact of children's programming education on the development of children's cognitive abilities. It begins by discussing the rise and development of children's programming education and the importance of cognitive abilities in the growth of children. It then analyzes in detail the characteristics and advantages of children's programming education, including teaching methods, teaching tools, and specific ways of cultivating children's cognitive abilities. Through empirical research, experiments were designed, and data were collected and analyzed to verify the positive impact of children's programming education on the development of children's cognitive abilities. At the same time, in response to the challenges faced by children's programming education, such as the shortage of teachers and the irrationality of curriculum settings, corresponding strategies have been proposed. Finally, the paper summarizes the significance of children's programming education in the development of children's cognitive abilities, emphasizes the necessity of further promoting and improving children's programming education, and looks forward to its future development trends.

Keywords: children's programming education, children's cognitive abilities, teaching methods, teaching tools, empirical research, challenges and strategies

1. Introduction

1.1 The Rise and Development of Children's Programming Education

1.1.1 The Origin and Development Course of Children's Programming Education Worldwide

The origin of children's programming education can be traced back to the 1960s when computer science was just emerging, and some educators began to consider how to introduce programming into the field of education. With the continuous development of computer technology, programming education has gradually gained attention globally.

In the 1980s and 1990s, some countries began to promote computer science education at the primary and secondary school stages, including programming courses. For example, the United States began to incorporate computer science into the school curriculum system during this period, cultivating students' computational thinking and problem-solving abilities.

Entering the 21st century, with the popularization of the Internet and the rapid development of information technology, children's programming education has received broader attention and promotion. Many countries have begun to regard programming education as an important means of cultivating future scientific and technological talents and have formulated relevant policies and curriculum standards.

In recent years, the global children's programming education has shown a vigorous development trend. More and more countries have included programming into the basic education curriculum system, and at the same time, various programming education institutions and online platforms have emerged like a spring breeze, providing

rich and diverse programming learning resources for children.

1.1.2 Policy Promotion and Market Status of Children's Programming Education in China

In terms of policy promotion, in recent years, the Chinese government has attached great importance to information technology education and has successively introduced a series of policy documents to promote the popularization of programming education in primary and secondary schools. For example, the "New Generation Artificial Intelligence Development Plan" issued by the State Council in 2017 proposed to set up artificial intelligence-related courses in the primary and secondary school stage and gradually promote programming education. Since then, local education departments have also responded in succession, formulating corresponding policy measures to promote the development of programming education.

In terms of market status, with the promotion of policies and parents' attention to the comprehensive quality education of children, the domestic children's programming education market has shown a trend of rapid growth. At present, the domestic children's programming education market is mainly composed of three forms: offline training institutions, online education platforms, and school courses. There are many offline training institutions that provide a rich curriculum system and teaching services; online education platforms are favored by more and more parents and children due to their convenient learning methods and rich teaching resources; in terms of school courses, more and more primary and secondary schools have begun to include programming into school-based courses or club activities, providing programming education for students.

1.2 The Importance of Cognitive Abilities for Children's Growth

1.2.1 The Key Role of Logical Thinking, Creativity, and Problem-Solving Abilities in Children's Learning and Life

Logical thinking is an indispensable ability for children's learning and life. It helps children understand the causal relationships, order, and patterns of things, thereby better mastering knowledge and skills. When learning subjects such as mathematics and science, logical thinking ability is particularly important. For example, when children solve mathematical problems, they need to use logical reasoning to analyze problems and find solutions. In daily life, logical thinking ability also helps children make reasonable decisions and judgments.

Creativity is an important driving force for children's development. It endows children with unique ways of thinking and innovative abilities, enabling them to propose novel solutions when facing problems. In fields such as art, literature, and science, creativity plays a crucial role. For example, children can give full play to their creativity in activities such as painting, writing, and inventing, showing their unique personality and talent.

Problem-solving ability is a key ability for children to deal with challenges. It helps children maintain a positive attitude when facing difficulties and setbacks and find effective solutions. In learning and life, children will encounter various problems, such as learning difficulties, interpersonal relationship issues, and minor troubles in life. Having good problem-solving skills allows children to face challenges more confidently, grow, and make progress.

1.2.2 The Impact of Good Cognitive Abilities on Children's Future Career Development

In today's digital age, the development of technology is rapidly changing, and the demand for talent is also constantly changing. Children with good logical thinking, creativity, and problem-solving abilities will be more competitive in their future career development.

Children with strong logical thinking skills can better analyze problems, make plans, and make decisions in their future careers. They have broad development prospects in fields such as technology, finance, and management. For example, in careers such as software development, data analysis, and project management, logical thinking skills are essential.

Children with strong creativity can create more value in their future careers. They have unique advantages in fields such as art, design, and innovation. For example, in fields such as advertising design, product development, and creative industries, creativity is the core competitiveness.

Children with strong problem-solving skills can better cope with various challenges and difficulties in their future careers. They can play an important role in various industries and become core figures in teams. For example, in fields such as healthcare, engineering, and education, problem-solving skills are key professional qualities.

In summary, the rise and development of children's programming education provide a new way for children to cultivate cognitive abilities, and good cognitive abilities are of great significance for the growth and future career development of children.

2. Characteristics and Advantages of Children's Programming Education

As an innovative educational model, children's programming education has unique value in cultivating

children's cognitive abilities. The following will elaborate in detail from three aspects: teaching methods, teaching tools, and specific ways of cultivating children's cognitive abilities.

2.1 Teaching Methods

2.1.1 Application of Project-Based Learning in Children's Programming Education

Guiding children to learn programming knowledge and skills through practical projects: Project-based learning integrates programming knowledge into specific practical projects, allowing children to learn programming while completing projects. For example, in designing a simple game project, children need to understand the logic and rules of the game, master basic programming concepts such as variables, loops, and conditional judgments, and implement various game functions by writing code. In this process, children not only learn programming knowledge but also improve their problem-solving and creative abilities.

Cultivating children's teamwork and communication skills: In project-based learning, children usually need to form teams to complete projects together. Team members need to cooperate, communicate ideas, and coordinate actions. For example, in an animation production project, some children are responsible for drawing characters, some for writing scripts, and some for programming to achieve animation effects. Through teamwork, children learn to listen to others' opinions, express their own ideas, solve problems together, and improve their communication and collaboration skills.

2.1.2 Advantages of Gamified Teaching Methods

Increasing the fun and attractiveness of learning: Gamified teaching turns programming learning into an interesting game experience, stimulating children's interest in learning. For example, using programming game platforms, children can learn programming knowledge in games and gain a sense of achievement by completing level tasks. Elements such as animation, sound effects, and reward mechanisms in the game can increase the fun of learning, making children more actively participate in learning.

Stimulating children's learning motivation and enthusiasm: Gamified teaching stimulates children's learning motivation and enthusiasm through setting challenges and reward mechanisms. Children continuously challenge themselves in the game, pursuing higher scores and achievements, thereby improving their enthusiasm for learning. For example, in programming competition games, children can compete with other players and strive for better results. This competitive atmosphere can stimulate their motivation to learn.

2.2 Teaching Tools

2.2.1 Features and Applicable Age Stages of Graphical Programming Tools

Introduction to tools like Scratch and Blockly: Scratch and Blockly are widely used graphical programming tools characterized by their intuitive and understandable nature. These tools replace traditional code writing with graphical modules, allowing children to program by dragging and dropping modules. For example, in Scratch, children can use various graphical modules to create animations, games, stories, and other works. Scratch also provides a rich material library and community sharing functions, making it easy for children to access resources and showcase their works.

How to help children easily get started with programming: Graphical programming tools are very suitable for young children to start learning programming. They do not require children to master complex programming language syntax, reducing the learning threshold. Through a graphical interface and simple operations, children can quickly get started with programming and experience the fun of programming. For example, children can achieve effects such as character movement and animation playback by dragging modules, intuitively feeling the logic and functionality of programming. Graphical programming tools also provide immediate feedback features, allowing children to immediately see the results of their programs, thus better understanding programming concepts.

2.2.2 Gradual Introduction of Code Programming Tools

Application of Python, Java, and other languages in children's programming education: As children grow older and their programming skills improve, code programming tools can be gradually introduced. Python and Java are two commonly used programming languages with powerful functions and wide application fields. In children's programming education, Python has attracted much attention due to its simple and easy-to-learn syntax. Children can write various programs by learning Python, such as data analysis, artificial intelligence applications, game development, etc. Java holds an important position in enterprise-level application development, and learning Java can lay the foundation for children's future career development.

Cultivating children's abstract thinking and programming standard awareness: Code programming requires children to have higher abstract thinking abilities and programming standard awareness. When using code programming tools, children need to abstract specific problems into programming problems, using abstract

concepts such as variables, functions, and classes to organize code. For example, when writing a mathematical calculation program, children need to abstract the mathematical problem into algorithms and data structures in the code, using variables to store data, and using functions to implement specific calculation functions. At the same time, code programming also requires children to follow certain programming standards, such as variable naming standards and code indentation standards, to improve the readability and maintainability of the code.

2.3 Specific Ways to Cultivate Children's Cognitive Abilities

2.3.1 Cultivation of Logical Thinking

Cultivating logical thinking through the structure of programming such as sequence, loops, and conditional judgments: The structure of programming such as sequence, loops, and conditional judgments is an effective tool for cultivating logical thinking. The sequence structure requires children to execute program steps in a certain order, cultivating their orderly thinking. Loop structures allow children to repeatedly execute a block of code, cultivating their iterative thinking. Conditional judgment structures require children to execute different code branches according to different conditions, cultivating their decision-making thinking. For example, when writing a program to calculate factorials, children can use loop structures to repeatedly perform multiplication operations and use conditional judgment structures to determine whether the input number is negative. Through the learning and application of these programming structures, children can improve their logical thinking skills.

Case Analysis: Solving mathematical problems or logical reasoning problems with programming: Programming can help children solve various mathematical problems and logical reasoning problems, further cultivating their logical thinking. For example, when solving mathematical pattern finding problems, children can use loops and conditional judgment structures in programming to traverse data and find patterns. When solving logical reasoning problems, children can use variables and conditional judgment structures in programming to represent logical relationships and make inferences and judgments. By solving practical problems with programming, children can apply abstract logical thinking to specific problems, improving their problem-solving abilities.

2.3.2 Stimulation of Creativity

Encouraging children to freely create programming works: Children's programming education should encourage children to use their imagination and creativity to freely create programming works. Children can choose different themes and projects for creation based on their interests and ideas. For example, children can design their own games, animations, music works, etc. In the creative process, they can try different programming techniques and ideas, showcasing their personality and talent.

Sharing and displaying works to cultivate innovation awareness and self-confidence: Sharing and displaying works is an important link in stimulating children's creativity. Children can share their works with classmates, teachers, and parents to obtain their feedback and suggestions. By sharing and displaying, children can understand the ideas and viewpoints of different people, broaden their horizons, and further stimulate innovation awareness. At the same time, receiving recognition and praise from others can also enhance children's self-confidence and encourage them to continue creating more outstanding works.

2.3.3 Enhancement of Problem-Solving Abilities

Facing errors and challenges in programming to cultivate children's abilities to analyze problems and find solutions: In the process of programming, children often encounter various errors and challenges, such as program running errors, incomplete function implementation, etc. Faced with these problems, children need to learn to analyze the causes of the problems and find solutions. For example, when a program has a running error, children can find the location and cause of the error by viewing error messages and debugging the code. Then, they can try different solutions, such as modifying the code and adjusting the algorithm, until the problem is resolved. Through this process, children can improve their abilities to analyze problems and find solutions.

Demonstration of problem-solving process in practical projects: Demonstrating the problem-solving process in practical projects can help children better understand the methods and steps of solving problems. Teachers can guide children to analyze problems in the project, propose solutions, and implement them step by step. For example, in a robot programming project, children may encounter the problem of the robot not being able to walk according to the predetermined route. Teachers can guide children to analyze the causes of the problem, which may be sensor failure, program logic errors, etc. Then, children can solve the problem by checking the sensors and debugging the program. Demonstrating the problem-solving process can teach children how to systematically analyze problems, propose solutions, and continuously try and improve, enhancing their problem-solving abilities.

In summary, children's programming education has unique characteristics and advantages, providing strong support for the growth and development of children through effective teaching methods, tools, and cultivation of children's cognitive abilities.

3. Empirical Research

The impact of children's programming education on the development of children's cognitive abilities is a widely concerned research topic. To deeply explore this issue, we have conducted a series of empirical researches, including experimental design, data collection, and analysis.

3.1 Experimental Design

The selection and grouping of research subjects are crucial. We screened children from different ages, genders, and learning backgrounds to ensure that the research results have a broad representativeness. Specifically, we selected children from different age groups in kindergartens, primary schools, and junior high schools, including boys and girls, and considered children's learning backgrounds in different subject areas such as mathematics, science, and art. Through random grouping, these children were divided into experimental and control groups. Children in the experimental group will receive children's programming education, while children in the control group will not receive any programming education, in order to accurately assess the impact of programming education on children's cognitive abilities in subsequent comparisons.

The control of experimental variables is key to ensuring the reliability of the research results. We determined variables such as the content, duration, and teaching methods of programming education. In terms of content, we designed programming courses suitable for different stages according to the age and cognitive level of children, including graphical programming and code programming. In terms of duration, we set a reasonable programming learning time to avoid adverse effects on children due to too long or too short a learning time. In terms of teaching methods, we adopted various methods such as project-based learning and gamified teaching to improve children's interest and participation in learning. At the same time, we also took a series of measures to exclude the impact of other factors on the development of children's cognitive abilities. For example, we ensured that the learning and living conditions of children in the experimental and control groups were basically the same except for programming education during the experiment. We also strictly controlled the experimental environment to reduce the impact of external interference factors on the experimental results.

The selection of measurement tools needs to be scientific and effective. We used standardized cognitive ability test tools, such as Raven's Progressive Matrices, creativity tests, etc., to objectively assess the level of children's cognitive abilities. At the same time, we also combined multiple data collection methods such as teacher evaluation and parent feedback to fully understand the performance and progress of children in the programming learning process. Teacher evaluation can evaluate children's programming skills, cognitive abilities, and learning attitudes from a professional perspective. Parent feedback can provide information on children's performance and changes in the home environment, providing richer information for the study.

3.2 Data Collection and Analysis

Data collection before the experiment for baseline testing is to understand the initial cognitive ability level of children. We conducted cognitive ability tests on children in both the experimental and control groups, including tests on logical thinking, creativity, and problem-solving abilities. Through these tests, we can obtain data on children's cognitive abilities before the experiment, providing a basis for subsequent comparative analysis.

Data collection during the experiment mainly includes observing children's performance and progress in programming learning, as well as recording children's thinking processes and strategies in solving problems. We collected various performance data of children in programming learning through classroom observation, homework analysis, group discussions, etc. At the same time, we also asked children to record their thinking processes and strategies when solving programming problems, in order to deeply understand their ways of thinking and problem-solving abilities.

Data collection after the experiment for effectiveness evaluation is to verify the positive impact of children's programming education on the development of cognitive abilities. We conducted cognitive ability tests on children in both the experimental and control groups again and compared the differences in test results between the two groups. At the same time, we also conducted in-depth analysis of the data collected during the experiment, including data on children's learning performance, thinking processes, and strategies. Through data analysis, we can determine the specific impact of children's programming education on the development of children's cognitive abilities and explore its impact mechanism.

Through scientific and reasonable experimental design and comprehensive and in-depth data collection and analysis, we can more accurately evaluate the impact of children's programming education on the development of children's cognitive abilities, providing a strong basis for the promotion of children's programming education.

4. Challenges and Strategies

4.1 The Problem of Teacher Shortage

4.1.1 Analysis of the Current Situation and Demand for Teachers in Children's Programming Education

Shortage of professional programming teachers: At present, the field of children's programming education is facing a serious shortage of professional programming teachers. With the rapid development of children's programming education, the demand for programming teachers is growing explosively. However, the existing number of programming teachers is far from meeting the market demand. On the one hand, programming education has not received enough attention in the traditional education system, resulting in a limited number of trained programming teachers. On the other hand, children's programming education has its particularities, requiring teachers not only to have a solid foundation in programming knowledge but also to understand child psychology, pedagogy, and other aspects, and be able to teach according to the characteristics of children. These interdisciplinary requirements make qualified programming teachers even rarer.

Requirements for teachers' interdisciplinary knowledge and teaching abilities: Children's programming education requires teachers to have interdisciplinary knowledge and teaching abilities. First, teachers need to master programming professional knowledge, including knowledge of programming languages, algorithms, data structures, etc. At the same time, teachers also need to understand knowledge in child psychology, pedagogy, and other aspects, and be able to teach according to children's cognitive characteristics and learning patterns. In addition, teachers also need to have good communication skills, teamwork skills, and innovation abilities, and be able to communicate and cooperate effectively with students, parents, and other educators, and continuously innovate teaching methods and curriculum content.

4.1.2 Strategies to Solve the Teacher Shortage Problem

Strengthen teacher training to improve teachers' programming teaching level: To solve the problem of teacher shortage, existing teachers can be trained. By organizing programming training courses, seminars, and workshops, teachers' programming professional knowledge and teaching abilities can be improved. Training content can include knowledge of programming languages, programming tools, teaching methods, curriculum design, etc. At the same time, industry experts and outstanding programming teachers can be invited to give lectures and share experiences, providing learning and communication opportunities for teachers. In addition, a long-term mechanism for teacher training can be established to regularly train and assess teachers to ensure continuous improvement of their teaching level.

Introduce online educational resources to make up for the shortage of teachers: Online educational resources can provide strong support for children's programming education and make up for the shortage of teachers. By introducing online programming courses, teaching videos, interactive platforms, and other resources, students can engage in independent learning without professional programming teachers. At the same time, teachers can also use online educational resources to assist in teaching and improve teaching effectiveness. For example, teachers can play online teaching videos in class to guide students in learning; or use online interactive platforms to organize students for programming practice and communication. In addition, remote teaching can also be realized through online educational resources to provide programming education services for students in remote areas.

Encourage colleges and universities to establish related majors to cultivate more talents in programming education. To fundamentally solve the problem of teacher shortage, it is necessary to encourage higher education institutions to offer relevant majors to cultivate more professionals in programming education. Universities can offer majors such as children's programming education and educational technology, training professionals with both programming knowledge and teaching capabilities. In terms of curriculum design, it can cover knowledge in programming languages, algorithms, data structures, child psychology, pedagogy, etc., focusing on students' practical abilities and innovative capabilities. At the same time, universities can also cooperate with enterprises and educational institutions to establish internship bases and employment channels, providing students with practical opportunities and job security.

4.2 The Issue of Unreasonable Curriculum Settings

4.2.1 Common Problems in the Curriculum Setting of Children's Programming Education

Lack of systematic and coherent curriculum content: Currently, there is a problem with the lack of systematic and coherent content in children's programming education courses. The curriculum content of some programming education institutions is quite fragmented, lacking overall planning and design. There is a lack of connection and transition between courses, which can lead to knowledge gaps and repetitive learning in the learning process. In addition, the update speed of curriculum content is also slow and cannot keep up with the development and changes of programming technology.

Unreasonable difficulty settings, not suitable for children of different age groups: The target of children's programming education is children of different age groups, and their cognitive levels and learning abilities vary. However, the difficulty setting of current programming education courses is often not reasonable and cannot well adapt to the needs of children of different age groups. Some courses are too difficult, beyond the cognitive

ability range of children, leading to learning difficulties and loss of interest; while some courses are too easy, which cannot meet the learning needs of students, wasting their time and energy.

4.2.2 Suggestions for Optimizing Curriculum Settings

Establish a scientific curriculum system covering different levels and fields of programming knowledge: To solve the problem of the lack of systematic and coherent curriculum content, it is necessary to establish a scientific curriculum system. The curriculum system should cover programming knowledge at different levels and in various fields, including programming foundations, advanced programming, and programming applications. There should be clear connections and transitions between courses, forming an organic whole. At the same time, the update speed of the curriculum content should also keep up with the development of programming technology, introducing new programming knowledge and technology in a timely manner.

Reasonably set the difficulty and progress of the curriculum according to the cognitive development laws of children: To adapt to the needs of children at different age stages, it is necessary to reasonably set the difficulty and progress of the curriculum according to the cognitive development laws of children. For younger children, simple graphical programming can be introduced to cultivate their programming interest and logical thinking skills; as they grow older, code programming can be gradually introduced to improve their programming skills and problem-solving abilities. In terms of curriculum progress, it should also be reasonably adjusted according to the learning situation of students to avoid too fast or too slow teaching progress.

Combine practical application scenarios to design creative and interesting course content: To improve students' interest and participation, it is necessary to combine practical application scenarios and design creative and interesting course content. Programming knowledge can be combined with practical application scenarios such as games, animations, and robots, allowing students to learn programming knowledge in practice. At the same time, programming competitions, project presentations, and other activities can also be organized to stimulate students' enthusiasm for learning and creativity. In addition, the design of course content should also focus on cultivating students' team cooperation and communication skills, allowing students to learn in cooperation and make progress together.

5. Conclusion

5.1 Review the Role of Children's Programming Education in Cultivating Logical Thinking, Creativity, and Problem-Solving Abilities

Children's programming education plays a key role in cultivating children's logical thinking through its unique teaching methods and curriculum. In learning programming, children need to understand and apply programming structures such as sequence, loops, and conditional judgments, which encourages them to think about problems in an organized manner, analyze causal relationships, and develop reasonable solutions. For example, when designing a simple game program, children need to plan the game's process, determine the action sequence and triggering conditions of the characters, a process that exercises their logical thinking skills.

The stimulation of creativity is another important outcome of children's programming education. Programming provides children with a platform for free creation, where they can use their imagination to design unique games, animations, or stories. In this process, children not only learn to use programming tools to realize their ideas but also cultivate an innovative spirit that dares to try new ideas and break through traditions. For example, some children have created games with unique gameplay or creative animations during their programming studies, demonstrating their creativity.

The improvement of problem-solving abilities is also a significant effect of children's programming education. In the process of programming, children inevitably encounter various problems, such as program errors and incomplete functions. Faced with these problems, they need to learn to analyze the root causes, try different solutions, and debug repeatedly. This experience cultivates children's patience and perseverance and also improves their abilities to analyze and solve problems. For example, when children encounter a situation where the program cannot run in programming, they need to troubleshoot errors step by step, which may be syntactic errors, logical errors, or parameter setting issues. By continuously trying and adjusting, they eventually solve the problem.

5.2 Emphasize the Importance of Cognitive Abilities for the Comprehensive Development of Children

Good logical thinking ability is indispensable for children in learning various subjects and in daily life. When learning subjects such as mathematics and science, logical thinking helps children understand abstract concepts, reasoning, and proof. In daily life, logical thinking helps children make wise decisions and solve practical problems. For example, when solving mathematical problems, children need to use logical reasoning to find solutions; when arranging their own learning and life plans, they also need to use logical thinking for rational planning.

Creativity is equally crucial for the comprehensive development of children. In today's society, the ability to innovate is the core competitiveness for both individuals and nations. Cultivating children's creativity can give them an advantage in future learning and work. Creativity can also enrich children's spiritual world, providing more space for expression and creation in the arts, literature, and other fields. For example, creative children may show unique talents in painting, writing, and other areas, adding color to their growth.

Problem-solving abilities are an important guarantee for children to face challenges and achieve self-growth. In the growth process, children will encounter various problems, and having good problem-solving abilities can make them more confident in facing difficulties and actively seeking solutions. This ability not only helps children succeed academically but also cultivates their resilience and adaptability, laying a solid foundation for their future development. For example, when facing academic difficulties, interpersonal relationship issues, and other challenges, children who are good at problem-solving can better adjust their mentality and behavior to find effective solutions.

References

Bennett, S., & Mateas, M., (2015). Programming by choice: A qualitative study of game programming in an after-school setting. *Journal of Computers in Education*, *2*(1), 51-74.

Brennan, K., & Resnick, M., (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association (AERA)*.

Grover, S., & Pea, R., (2013). Computational thinking in K-12: A review of the state of the field. *Educational Research Review*, *12*, 1-12.

Lee, I., & Tsai, C., (2010). Exploring teachers' perceived self efficacy and technological pedagogical content knowledge with respect to educational use of the World Wide Web. *Instructional Science*, *38*(1), 1-21.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Silver, M., (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60-67.

Wing, J. M., (2006). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717-3725.

Yao, L., & Resnick, M., (2017). The impact of programming languages on learning: A case study of Scratch and Python. *Journal of Educational Computing Research*, *55*(5), 649-673.

Copyrights