

Fundamentals of Programming



Overview:

In this unit, students will learn the basics of block based programming. They will create an account and explore the Scratch website. Students will take two different approaches to creating a program, experimenting through open exploration and following step by step directions. Learning activities will provide students with a foundational understanding of how to use simple features within the Scratch environment. After designing their first project, they will learn how to share their projects to a studio and provide meaningful feedback to others. Finally, they will develop strategies for testing and debugging problems within a Scratch program.



Student Agency (expressing, connecting, questioning, and bringing meaning to their learning):

Activities in this unit are designed to allow students to learn through exploration, experimentation, and collaboration. Students are encouraged to try new things and take responsible risks. They have choice in what features in Scratch to explore and can start the process of expressing themselves through programming. They are also given support through step-by-step instruction and asked to think about when they feel the most creative. They question and experiment to learn how things work within the Scratch program. Students gain perspective as they work in pairs or small groups to share their creations and learn how to offer clear, constructive criticism to one another.



Pathway:

Coding/Computational Thinking



Duration:

This unit contains 4 lessons and will span 7 or 8 classes (40 minutes each).



Grade Levels:

This Unit is recommended for third grade, but can be used as an introduction for any grade level getting started with programming.



CC BY-NC-SA 4.0. This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0>

Authorization to reproduce this lesson plan in whole or in part is granted. Suggested Citation: Shad Wachter (2019). TL Unit 1 - Introduction to Scratch. South Fayette Township School District.

Fundamentals of Programming

 <u>Competencies & Practices</u>	 <u>Student Artifacts/Evidence</u>
Collaboration	Students share ideas and discoveries with others, offer feedback, and develop strategies for programming.
Incremental and Iterative	Students build their programs step by step, experimenting, and running the program after changes are made.
Testing and Debugging	Students find problems in one another's projects and offer suggestions. Students develop strategies for testing and debugging programs.



Essential Question:

- What can you create with Scratch?
- How can critique improve a project?
- How do you fix a problem?



Concepts/Practices:

- **Experimenting and Iterating** - The computational practice of developing a little bit, then trying it out, then developing some more.
- **Testing and Debugging** - The computational practice of making sure things work and finding and solving problems when they arise.
- **Events** - The computational concept of one thing causing another thing to happen.
- **Sequence** - The computational concept of identifying a series of steps for a task.
- **Loops** - The computational concept of running the same sequence multiple times.
- **Parallelism** - The computational concept of making things happen at the same time.



Habits of Mind:

- Taking responsible risks
- Responding to the world with wonderment and awe



CC BY-NC-SA 4.0. This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0>

Authorization to reproduce this lesson plan in whole or in part is granted. Suggested Citation: Shad Wachter (2019). TL Unit 1 - Introduction to Scratch. South Fayette Township School District.

Fundamentals of Programming

- Thinking about your thinking
- Finding humor
- Thinking interdependently
- Thinking and communicating with clarity and precision



Unit Sequence:

1. **Lesson 1.1: Introduction to Scratch**
2. **Lesson 1.2: Step-by-Step**
3. **Lesson 1.3: Sharing and Critiquing**
4. **Lesson 1.4: Testing and Debugging**
5. **Lesson 1.5: Debugging Challenges**
6. **Lesson 1.6: Interactive Collage**



Standards

CTSA Grades 3-5

1B-AP-08: Compare and refine multiple algorithms for the same task and determine which is the most appropriate.

1B-AP-10: Create programs that include sequences, events, loops, and conditionals.

1B-AP-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

1B-AP-12: Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.

1B-AP-13: Use an iterative process to plan the development of a program by including others' perspectives and considering user preference.

1B-AP-15: Test and Debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

1B-AP-16: Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.

1B-AP-17: Describe choices made during program development using code comments, presentations, and demonstrations.

ISTE

1c: Students use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.

2b. Students engage in positive, safe, legal, and ethical behavior when using

5a. Students formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.



CC BY-NC-SA 4.0. This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0>

Authorization to reproduce this lesson plan in whole or in part is granted. Suggested Citation: Shad Wachter (2019). TL Unit 1 - Introduction to Scratch. South Fayette Township School District.

Fundamentals of Programming

5c. Students break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.

5d. Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.

7b. Students use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints.



CC BY-NC-SA 4.0. This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0>

Authorization to reproduce this lesson plan in whole or in part is granted. Suggested Citation: Shad Wachter (2019). TL Unit 1 - Introduction to Scratch. South Fayette Township School District.