Pencil Code Topic: Making a First Program

Objectives: Students will be able to:

- Use **speed**, **fd**, **lt**, **rt**, **bk**, and **pen** commands to draw shapes in Pencil Code.
- Test, experiment, and learn about **functions** using immediate commands.
- Plan and build a program to match a pencil sketch on paper.
- Save and share programs on their own website with http://pencilcode.net/.

Learning Objectives: Students will be able to:

- Use a computing tool and techniques to create artifacts.
- Develop an algorithm to solve a problem.

Essential Knowledge: Students need to know:

- How to launch a web browser and how to navigate to a website.

Outline:

- Starter:
    - Video Lesson Starter (5 minutes): Watch Online Tutorial, David Bau
    - Or Explanation (5 Minutes): Walk through a first “House” program with Pencil Code.
- Activity 1 (15 minutes): Students recreate their own “House” program and save it.
- Challenge: Advanced students add a door or window or other other shapes.
- Activity 2 (15 minutes): Following handout with some suggested drawings, create a “sailboat”.
- Activity 3 (15 minutes): Graph out own drawing and use primitives to create artifact.
- Wrap up (10 minutes): Discussion about problem-solving with code.

Details:

- Explanation/Starter:
    - Modifying the program text changes the behavior of the turtle.
    - Commands can be run individually.
    - Commands are functions with arguments. The function name is the first word.
    - Words are case-sensitive and should be lowercase. Spaces are also important.
    - A simple program is a “script” that sequences commands in linear order.
- Activity details:
    - First mimic the steps to make a program; then figure the steps to follow a plan for a program; then create plan for a new program and then create it independently.
    - Syntax: (1) function names must be lowercase (2) include a space between the function and the argument; (3) no indenting; and (4) no space if the argument is in parentheses.
- Wrap up: As a class discuss the following:
    - Testing and experimentation is useful when following a plan.
    - Smaller turns make obtuse angles and larger turns make acute angles.
    - Invisible moves and variations: pen on, pen off, pen thickness, and favorite colors.
    - Why long or repetitive programs become hard to read, how to use #-comment lines.