Variables: Data Types
Grades 6-12
Designed by Robolink

Summary:

Students may have learned about variables in their math class, but they are slightly different in computer science. In this lesson, they will learn about variables and how they work in programming. They will then manipulate variables in their programs in order to make shapes of multiple sizes.

Guiding Question(s):

- How, and why, are variables used in programming?
- How can variables be manipulated in a program?
- How can variables be used to program your CoDrone EDU to fly?
- How can variables change the outcome of your program?

Learning Objectives:

- Define what a variable is in programming
- Explain how variables can be used in a program
- Describe the benefits of using variables in your programs
- List the different types of data in python programs
- Explain why we have different data types and provide examples of when to use the different data types

Steps to Success

1. Why Variables?
2. What's in a Name? Variable Naming Best Practices
3. Types of Variables
4. Practice Exercises with Variables
5. Going in Squares Again
6. Challenge: The Drone Slide

Materials needed:

- CoDrone EDU, remote, and USB cable for each student or student group
- Laptop with Internet access and PyCharmEDU installed for each student or student group
- Charged CoDrone EDU batteries and extra chargers

- Tape measure (only for math extension)

Curriculum Overview

Lesson Title: Variables **Time:** 1 hour 30 minutes

Engagement: (Introduction)

- Ask students what a variable is in either math or science. What do they do? How do they work? Next, ask students to talk to the people next to them to come up with a definition of a variable from computer science. Show the video below and ask for observations, thoughts, and questions. How did their definition change?
  Video: CS Principles: Intro to Variables - Part I

Exploration: (Activity)

- Have students complete lesson Variables on Robolink Basecamp, including the Variable Square challenge at the end. Warn students that they will probably be testing and modifying their programs a lot with this lesson!

Explanation: (Recap)

- Ask students to use pseudocode to explain to a partner how they programmed their drone to fly in a square, using both their own words and academic language. They should also explain the benefits of using variables for this program.

Elaboration: (Extension)

- Have students create a square (or other shape!) program with the drone's time as a variable. They will need to find a way to increase the variable's value in the middle of the program, so the shape keeps getting bigger.
- As an extension to this challenge, students can try using decimal values in their variable. Is there a significant difference between 1 second, 1.5 seconds, and 2 seconds?

CCSS Math extension (middle school):

- Have students work in pairs, with one flying the drone and one marking its angles as it turns, as well as its height. The pairs should then measure the distance between sides of their assigned polygon as well as its height to find their area, volume, and surface area.

- Next, have students change the values of the variables in their program to smaller or larger numbers, and then repeat the previous step. How does manipulating the variables change the area, volume, and surface area of their polygon?

CCSS Math extension (high school):

- Have students change the variables in their polygon program to fly their CoDrone EDU in a rotation, reflection, or translation of the original shape.

Evaluation:

- Have students answer the following questions in their engineering journals. A Robolink example is available on [Google Sites]:
    1. In your own words, explain how to use a variable in a program and what its benefits are.
    2. Include the pseudocode for your time variable program. This can be either through words or a flow chart.
    3. Did you have any problems running your code or using your drone? If so, what did you do to fix them?
    4. What did you learn?
    5. (only if middle school polygon calculation challenge was completed) Explain what happened when you changed the values of the variables in your polygon's programs. Were your volume, area, and surface area larger or smaller? How did this relate to the new values?
    6. (only if high school polygon challenge was completed) Explain how you changed the values of your variables to fly your CoDrone EDU in a rotation, reflection, or translation of your original shape. Did your program work out the way you originally thought it would? Why or why not?

Related Vocabulary: Boolean, call, declare, drone, float, initialize, int, landing, library, movement command, negative, pitch, positive, Python, roll, string, takeoff, throttle, value, variable, yaw

Standards:

CCSS:

ELA-LITERACY.RST.6-8.3: Follow precisely a multistep procedure when carrying out experiments, taking measurements, or performing technical tasks.
ELA-LITERACY.RST.9-10.3: Follow precisely a complex multistep procedure when

carrying out experiments, taking measurements, or performing technical tasks, attending to special cases or exceptions defined in the text.
ELA-LITERACY.RST.11-12.3: Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks; analyze the specific results based on explanations in the text.
MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.
MATH.PRACTICE.MP5: Use appropriate tools strategically.
MATH.PRACTICE.MP7: Look for and make use of structure.
MATH.CONTENT.6.G.A.4: Represent three-dimensional figures using nets made up of rectangles and triangles, and use the nets to find the surface area of these figures. Apply these techniques in the context of solving real-world and mathematical problems.
MATH.CONTENT.7.G.B.6: Solve real-world and mathematical problems involving area, volume and surface area of two- and three-dimensional objects composed of triangles, quadrilaterals, polygons, cubes, and right prisms.
MATH.CONTENT.HSG.CO.A.5: Given a geometric figure and a rotation, reflection, or translation, draw the transformed figure using, e.g., graph paper, tracing paper, or geometry software. Specify a sequence of transformations that will carry a given figure onto another.

NGSS:
MS-ETS1-2: Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.

CSTA:
2-CS-03: Systematically identify and fix problems with computing devices and their components.
2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.
2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.
2-AP-19: Document programs in order to make them easier to follow, test, and debug.
3B-AP-16: Demonstrate code reuse by creating programming solutions using libraries and APIs.

ISTE:
5D: Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.
6A: Students choose the appropriate platforms and tools for meeting the desired objectives of their creation or communication.