For Loops
Grades 6-12
Designed by Robolink

Summary:
Loops are used to repeat the same lines of code for as many times as needed. In this lesson, students will learn how to include while loops and for loops in their CoDrone EDU programs.

Guiding Question(s):
- What is a loop?
- How is a loop used in programming?
- How can loops be used in a CoDrone EDU's flight?

Learning Objectives:
Students will be able to:
- Identify the purpose of loops
- Write and execute different types of For loops
- Identify when to use a For loop in a program
- Identify the components of the range command
- Use the range command to count up, down, and in unique increments
- Describe other ways to count and print using a for loop
- Utilize loop knowledge to create flight patterns for different types of polygons
- Explain what is a private or local variable and how they are used in loops

Step for Success:
- What are Loops?
- Anatomy of a FOR Loop
- First FOR Loops
- A More Efficient Square
- Challenge: Fly a Polygon

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

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

Engagement: (Introduction)

- Ask students to talk to the people near them and come up with definitions for while loops and for loops. When everyone is finished, have them share their answers with the class. Show the video below and ask for thoughts, observations, and questions.
  Video: Intro to Programming: Loops

Exploration: (Activity)

- Have students complete Loops on Robolink Basecamp. This includes the Tower of Terror challenge at the end of the lesson. If students are having problems, they need to talk to classmates before asking the teacher!

Explanation: (Recap)

- Ask students to use pseudocode to explain their Tower of Terror program to a partner, using both their own words and the appropriate academic language.

Elaboration: (Extension)

- Students have programmed their drone to fly in a square a few times now, but what about other shapes? Ask students to come up with a three-sided regular (equilateral and equiangular) shape, and draw it on the board. Keep increasing the number, at least up to an octagon!
- Have students program their drone to fly in a regular triangle using loops. After they are successful, they should move on to a square, then pentagon, hexagon, etc. See who can program the highest number of successful shapes in the time allotted!

Evaluation:

- In a journal or on a worksheet, have students answer the following questions. An example of engineering portfolios is available on Google Sites.

1. In your own words, what is a loop, and why would a programmer use one in their code?
2. What is the difference between a while loop and a for loop?
3. Explain one of your shapes programs using pseudocode. Why did you choose to make your program the way you did?
4. Did you have any problems running any of your codes? If so, what did you do to fix them?

5. What did you learn?

Related Vocabulary: condition, drone, for loop, increment, landing, library, loops, maximum, minimum, movement command, negative, pitch, positive, private variable, Python, roll, takeoff, throttle, while loop, 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.

CSTA:

2-CS-03: Systematically identify and fix problems with computing devices and their components.
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.