Iteration: While Loops

Unit 2. Developing Programming Tools

Revision Date: Jan 11, 2020
Duration: 1 50-minute session

Lesson Summary

Summary

Students continue learning about iteration by using `while` loops and nested iteration. Students work through a guided tutorial on `while` loops and learn more turtle graphics features. They also have the opportunity to utilize the pair programming model to facilitate collaboratively writing programs using `for` loops, `while` loops, and turtle graphics. Throughout the lesson, students are given the opportunity to use their journal as a reflective tool.

Outcomes

- Students will know that iteration can be performed in Python using `while` loops.
- Students will understand that `for` and `while` loops can be written (nested) inside of other `for` and `while` loops.
- Students will be introduced to the REPEAT loops on the exam reference sheet.
- Students will understand and utilize the pair programming model as means to facilitate collaboration.

Outline

1. Getting Started (5 min)
2. Guided Activity (15 min) - `for` and `while` loops with turtle graphics
3. Collaborative Activity (25 min) - Shared coding of an iterative program
4. Wrap Up (5 min) - Journal Activity

Learning Objectives

CSP Objectives

- *EU CRD-1 - Incorporating multiple perspectives through collaboration improves computing innovations as they are developed.*
    - LO CRD-1.B - Explain how computing innovations are developed by groups of people.
    - LO CRD-1.C - Demonstrate effective interpersonal skills during collaboration.

- *EU CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.*
    - LO CRD-2.E - Develop a program using a development process.

- *EU AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.*
    - LO AAP-2.J - Express an algorithm that uses iteration without using a programming language.
    - LO AAP-2.K - For iteration: a. Write iteration statements. b. Determine the result or side-effect of iteration statements.

Math Common Core Practice:

- MP1: Make sense of problems and persevere in solving them.
- MP2: Reason abstractly and quantitatively.

- MP5: Use appropriate tools strategically.
- MP6: Attend to precision.
- MP7: Look for and make use of structure.
- MP8: Look for and express regularity in repeated reasoning.

Common Core ELA:

- RST 12.3 - Precisely follow a complex multistep procedure
- RST 12.4 - Determine the meaning of symbols, key terms, and other domain-specific words and phrases
- RST 12.9 - Synthesize information from a range of sources
- WHST 12.2 - Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes

NGSS Practices:

- 3. Planning and carrying out investigations
- 5. Using mathematics and computational thinking
- 6. Constructing explanations (for science) and designing solutions (engineering)
- 8. Obtaining, evaluation, and communicating information

Key Concepts

- Students will describe how iteration can be performed in Python using `while` loops.
- Students will demonstrate that `for` and `while` loops can be written (nested) inside of other `for` and `while` loops.
- Students will describe programs that are developed for creative expression, to satisfy personal curiosity, or to create new knowledge and that these programs may be developed with different standards or methods than programs developed for widespread distribution.
- Students will understand and utilize the pair programming model to facilitate collaborative work.

Essential Questions

- How can computing and the use of computational tools foster creative expression?
- How are programs developed to help people, organizations or society solve problems?
- How are programs used for creative expression, to satisfy personal curiosity or to create new knowledge?
- How do people develop and test computer programs?
- Which mathematical and logical concepts are fundamental to computer programming?

Teacher Resources

Student computer usage for this lesson is: **required**

In the Lesson Resources folder:

- Guided activity: "Turtle Graphics Guided Activity: The while loop" in the Lesson Resources folder

Other:

- How to Think Like a Computer Scientist (http://interactivepython.org/runestone/static/thinkcspy/index.html) (Ch.8 More About Iteration (http://interactivepython.org/runestone/static/thinkcspy/MoreAboutIteration/toctree.html))
- CS1 Python Programming Projects Archive http://www.cse.msu.edu/~cse231/PracticeOfComputingUsingPython/ (http://www.cse.msu.edu/~cse231/PracticeOfComputingUsingPython/)

Lesson Plan

Getting Started (5 min)

What will be displayed at the end of this program?

```
        y=0
        for x in range (0, 8):
            y += 1
        print (y)
```

Introduction of Content (15 min)

- Use the guided activity ("Turtle Graphics Guided Activity: The while loop" in the lesson folder) to discuss while loops and nested iteration.
- To save time, have these programs open in your IDE before class starts.

Collaborative Coding Activity (25 min)

Define the Pair Programming Model: As the name implies, pair programming is where two programmers share one computer. One programmer codes while the other observes the code as it is typed in. The two programmers switch roles frequently. Common models such as pair programming exist to facilitate collaboration.

1. Show the Pair Programming.ppt in the Teacher Resources folder
2. Discuss the potential pros/cons of pair programming.

Introduce the Activity:

1. Go to the slide 4 in the Pair Programming.ppt to introduce the coding assignment.
2. Pair the students up. If you have an odd number of students, make one group of 3. Have students work in pairs to complete the coding activity that uses while loops and nested iteration. Each pair will write a Python program called SumOfInts.py to calculate the sum of integers, and then modify it using a nested while loop.
3. Once each pair completes this portion, display slide 5 in the Pair Programming.ppt for the nested loop portion.

Solutions to both programs are in the slideshow.

Class Discussion (+10 min)

Open Repeat Loops.ppt in the Teacher Resourses.

Say: The College Board's exam reference sheet uses REPEAT loops instead of for and while loops. The REPEAT n TIMES { } executes the block of statements n times.

The College Board's REPEAT UNTIL(condition) { } structure executes the block of statements without ending until the Boolean expression condition evaluates to true. If the condition never evaluates to true the loop coninue indefinitely (called an infinite loop).

In the REPEAT UNTIL(condition), the Condition is evaluated before the loop body is executed so if the condition evaluates to true the loop body is never execued.

Ask: Which of the two REPEAT loops is most like the Python for loop and which is most like the Python while loop?

Display the example REPEAT loops questions in the slideshow, one at a time. Give the students a moment to determine the answer, then prompt students for his/her response "How many chose Answer A?" Then do the same for B, C, D. Then give correct answer (answer is in the Notes section of the ppt).

Wrap Up (5 min)

Journal: In your journal, summarize the process that you used with your partner to create the collaborative project. How did each partner contribute? Did you have any differences, if so, how did you resolve them? Was there any benefits to working in a pair ie able to identify programming and/or algorithm errors?

Options for Differentiated Instruction

1. Give students a copy of the guided activity instructions so they can follow along.
2. Possibly give students a code stem to work from for the collaborative project (while ensuring that they have the opportunity to problem-solve by thinking about and implementing an appropriate loop structure).

Evidence of Learning

Formative Assessment

Various checking-for-understanding techniques:

- Temperature checks
- Teacher reviews of student's code
- Thumbs up/ thumbs down
- Questioning throughout the lesson (whole group / small group / individual)

Quick quizzes

Peer review

Interactive journaling

Summative Assessment

Students will work collaboratively to develop a program that uses nested iteration and turtle graphics.