(http://csmatters.org) 2 - 8

0b10 - 0b1000

Comparison, Logical Operators, and Conditional Execution

Unit 2. Developing Programming Tools

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

Lesson Summary

Summary

Students will learn how programs can solve problems using the various types of conditional statements in Python programs.

Learning Objectives

CSP Objectives

- *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.B - Explain how a program or code segment functions.
  - LO CRD-2.C - Identify input(s) to a program.
  - LO CRD-2.E - Develop a program using a development process.
  - LO CRD-2.G - Describe the purpose of a code segment or program by writing documentation.
  - LO CRD-2.I - For errors in an algorithm or program: a. Identify the error. b. Correct the error.
  - LO CRD-2.J - Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program.
- *EU AAP-1 - To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.*
  - LO AAP-1.A - Represent a value with a variable.
  - LO AAP-1.B - Determine the value of a variable as a result of an assignment.
  - LO AAP-1.D - For data abstraction: a. Develop data abstraction using lists to store multiple elements. b. Explain how the use of data abstraction manages complexity in program code.
- *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.A - Express an algorithm that uses sequencing without using a programming language.
  - LO AAP-2.B - Represent a step-by-step algorithmic process using sequential code statements.
  - LO AAP-2.E - For relationships between two variables, expressions, or values: a. Write expressions using relational operators. b. Evaluate expressions that use relational operators.
  - LO AAP-2.F - For relationships between Boolean values: a. Write expressions using logical operators. b. Evaluate expressions that use logic operators.
  - LO AAP-2.G - Express an algorithm that uses selection without using a programming language.
  - LO AAP-2.H - For selection: a. Write conditional statements. b. Determine the result of conditional statements.
  - LO AAP-2.I - For nested selection: a. Write nested conditional statements. b. Determine the result of nested conditional statements.
  - LO AAP-2.L - Compare multiple algorithms to determine if they yield the same side effect or result.
  - LO AAP-2.M - For algorithms: a. Create algorithms. b. Combine and modify existing algorithms.

Math Common Core Practice:

- MP1: Make sense of problems and persevere in solving them.
- MP2: Reason abstractly and quantitatively.
- 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 12.2 - Determine central ideas and conclusions in the text
- 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
- RST 12.10 - Read and comprehend science/technical texts
- WHST 12.2 - Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes

Key Concepts

Decisions in programs are made using conditional statements.

Outcomes

- Students will understand how conditional logic controls program flow.
- Students will be able to solve problems that require conditional logic.
- Students will be able to look for answers in documentation (API)
- Students will be introduced to comparison and logical operators, and conditionals on the exam reference sheet.

Essential Questions

- How are programs developed to help people, organizations or society solve problems?
- How do computer programs implement algorithms?
- How do people develop and test computer programs?
- Which mathematical and logical concepts are fundamental to computer programming?

How are comparison operators and Boolean expressions used with conditional statements?

Teacher Resources

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

In the Lesson Resources folder:

- Nested If Statement Setup

Reference Texts:

- RUNESTONE BOOK: How to Think Like a Computer Scientist - http://interactivepython.org/runestone/static/thinkcspy/Selection/toctree.html (http://interactivepython.org/runestone/static/thinkcspy/Selection/toctree.html)
- Python for Everybody- http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf (http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf)

Online interpreters:

- http://www.compileonline.com/execute_python3_online.php (http://www.compileonline.com/execute_python3_online.php)
- http://www.pythontutor.com/visualize.html#mode=edit (http://www.pythontutor.com/visualize.html#mode=edit)
- http://labs.codecademy.com/#:workspace (http://labs.codecademy.com/#:workspace)

Lesson Plan

Getting Started (5 min)

Journal

Think of a decision you make in your daily life and how you make the decision. In your journal, write about your decision and the process you use to decide.

Teaching note: take a few minutes to have students share responses (whole class, elbow partners, small groups).

Introduction of Content (30 min)

Brief discussion of comparison operators [5 min]

==, !=, <, >, >=, <=

Briefly discuss logical operators [5 min]

Includes AND, OR, and NOT

Say: The College Board's exam reference sheet uses:

- Comparison operators a = b a ≠ b a > b a < b a ≥ b and a ≤ b that evaluate to a boolean value (true/false) and test the relationship between two variables, expressions, or values
- logical operators NOT, AND, and OR as does Python
- NOT evaluates to true if the condition is false; otherwise, it evaluates to false
- AND evaluates to true only if both operands are true
- OR evaluates to true if either or both operands are true
- The operands that are manipulated by logical operators are either boolean values or expressions that evaluate to boolean values.

Example: Type various combinations of Boolean values (True / False) with Boolean operators (not, and, or) into the IDE (PyCharm)

Conditional Statements [20 min]

Show physical representation using a real-life example

Suggested Activity: What's in the box?

Materials: Two small opaque containers, one small item for each container, two post-it notes.

Set-Up Directions: Set up the activity before the class arrives, following the directions below:

1. Place one item in each container.
2. Using the post-it notes, label one box "green" and the other "black."
3. On the board, write the two colors (green and black) on the board to record the student tally.
4. On the board, write the following pseudocode:

```
if numofGreen > numofBlack:
    open green box
else:
    open black box
```

Activity Directions:

1. Ask: "If you could choose one of these colors, which one would it be? Please only vote once. Raise your hand if you would choose Green." Record the number of students who voted for Green on the board.
2. Ask: "Raise your hand if you would choose Black." Record those votes.
3. Show the students the pseudocode on the board and work through the if/else statement using the data collected from the class.
4. Open the box and display the item that the class voted for.

Discussion of Conditional Execution (if, if/else)

Show the example from How to Think Like a Computer Scientist text (ActiveCode:6 (ch05_4)) or a similar example. Briefly demonstrate how to read flowcharts while showing the example from How to Think Like a Computer Scientist. Point out that the API (https://docs.python.org/3/library/ (https://docs.python.org/3/library/)) is always available as a language reference to help translate from algorithms/flowcharts into code. The API describes many tools to make it easier to create programs and is an important tool for all programmers.

Hint

If desired an if-else statement that returns boolean values can be replaced by the conditional expression or vice versa.

Instead of:
if (x < y):
return True
else:
return False

we could use:
return x < y

Check for understanding:

Define: A code segment refers to a collection of program statements that are part of a program.
Have students answer the following questions:

1. What will be printed by the following code segment?

```
x=15
if x==25:
  print ('Pizza is yummy')
else:
   print ('My teacher is awesome')
```

2. What will be printed by the following code segment?

```
x=35
y=52
if x!=25 and y==52:
  print ('Pizza is yummy')
else:
 ('My teacher is awesome')
```

Use IDE (PyCharm) to show how to create an if/else statement

Suggested Coding Example:

```
n = input('Please enter your password: ')
if n=='P@s5w0d':
    print ('Welcome, correct user!')
else:
    print ('Incorrect, try again')
```

Exam Reference Sheet

Say: The College Board's exam reference sheet uses if and if-else statements like Python except it uses curly braces to indicate the block of statements controlled by if or if-else statements.

IF(condition)

{

}

IF(condition)

{

}

```
ELSE

{


}
```

Independent Activities

1 Complete the following Runestone Activities

- 7.6. Nested conditionals (https://runestone.academy/runestone/books/published/thinkcspy/Selection/Nestedconditionals.html)
- 7.7. Chained conditionals (https://runestone.academy/runestone/books/published/thinkcspy/Selection/Chainedconditionals.html)

2 If/else practice problem: In the IDE, write a program that will prompt a user to enter a value for a food item. Evaluate the variable `food`**.** If the value of `food` is equal to "potato salad," display "In Stock". If the value of `food` is not equal to "potato salad," display "Not in Stock".

Test your program with the following values for food:

pizza
popcorn
potato salad

Wrap Up (5 min)

Journal

Thinking about conditional execution, answer the following questions.

1. What difficulties, if any, did you encounter during the development of your program?
2. After running your independent program, were your predictions correct?
3. What conditional execution concepts do you need clarified?
4. How is it helpful to have a reference available for each programming language?

Options for Differentiated Instruction

Alternate Instructional Strategy for Guided Practice :

- For guided practice, have a list of instructions for how to write the example presented.

Alternate Instructional Strategy for Journal: Interactive Journaling

- Comment on the journal entry by asking check-for-understanding questions or clarifying any misconceptions students write about.

Evidence of Learning

Formative Assessment

- A variety of checking for understanding techniques
  - Temperature checks
  - Teacher review student's code
  - Thumbs up/ thumbs down
  - Questioning thoughout the lesson (whole group / small group / individual)
- Quick quizzes
- Interactive journaling

Summative Assessment

- Students create a small program demonstrating conditional execution
- Reflective journal entry

Authored by: CS Matters in Maryland
Website: csmatters.org (http://csmatters.org)
Email: csmattersinmaryland@gmail.com (mailto:csmattersinmaryland@gmail.com)