Lab 2 Conditionals

Instructor Guide

Overview

As students enter, they will verbally review over what *variables*, *printing*, *strings*, and *input* are, and they syntax for each. Then students will move into the **Pythons and Ladders Paper Programming**, receiving an introduction to *if-else conditional statements*. Following, there will short lecture to review conditionals and understand the structure of code more.

Students will then talk with their mentors about what *elif* is, and complete the challenges in **Week One Fun**! This lesson should begin to challenge the students syntactically and logically, mentors should be helpful but patient so students have the opportunity to try to understand these challenges on their own.

Learning Goals

- A boolean is a type, True or False
- If - else are conditional statements testing to see if something is true or not
- == tests for equality; whether something is True or False. If something is True, then the material within that if or elif statement (what is indented) is run
- You can have an if-statement without an else-statement, but you need an if-statement for an else-statement
- You ALWAYS use ==, instead of = inside of a conditional statement
- != is testing if something is not equal to something else
- Elif is used, if you wanted to test multiple equalities
- You enter the else statement if the conditionals in the previous if and elif's were False

Have comments, concerns, or feedback? We love to hear about ways to improve our curriculum! Fill out our Feedback Form or email cmu.teknowledge@gmail.com, and we will get back to you as soon as possible.

Personal Growth Goals

- Decreased Fixed Mindset: This is the first lab, in which they will have to start thinking logically about specific problems. Students can get discouraged if they fall behind because of a logical misunderstanding in conditionals, so instilling and helping the students believe they can complete these challenges will help decrease that mindset.
- Critical Thinking: Along with having to logic through some of these challenges, students should also start thinking about how these tools, can be applied in other situations.

Skills Required

- Multiplication
- Understanding of variables, print, input, and strings

Resources Required

- Computers for either every student or every pair of students
- Python 3 and a text editor needs to be installed on all the computers
- One mentor per 2-3 students
- A projector to project the central instructor's computer
- Pencils for each student

Instructor Preparation

1. Make sure all the computers students will use have Python and a text editor (right now, we use Pyzo) installed (check to see that students have a way to save/access files.
2. Load the following programming files onto each computer:
    a. 01_fun_question_game.py
3. Print out one copy of the **Pythons and Ladders Paper Programming** for each student.

In Depth Description of Lab Activities

Phase 1: Setup

1. Before the students arrive, open the following files in a text editor on each computer:
    a. 01_fun_question_game.py
2. Have one printed out **Pythons and Ladders Paper Programming** for each student on the desks, and pencils available if needed.

Have comments, concerns, or feedback? We love to hear about ways to improve our curriculum! Fill out our Feedback Form or email cmu.teknowledge@gmail.com, and we will get back to you as soon as possible.

Phase 2: Introduction

1. As students are coming in, have a verbal review with the class of the material taught in the previous lab.
    a. Questions to ask:
        i. What is a variable? What is a string? What does input return? Why do we put quotation marks around certain words in print statements? When is there a case when you don't put quotation marks around words in print statements? etc.

Phase 3: Pythons and Ladders Paper Programming

1. Teach the basics of:
    a. What an if-else statement is
    b. How basic math assignment operators work from the syntax guide (+=, *=)
2. Next students will try this paper programming on their own. There are a number of 'problems' that can occur in the puzzle such as infinite loops, regular loops, dead code. If students run into these problems their mentors can discuss these topics briefly with them, but the main purpose of this activity is to teach them about conditionals. The students should try to guess and check (side-note: this is an algorithm!) with different numbers until they find the correct path.
3. Ending this activity should be a class discussion about how this activity differs from Python code, namely, that you can jump around between lines of code and revisit already evaluated lines of code and such.
4. If the central instructor has time, they can discuss the following aspects of the activity: 0,5,6 lead to infinite loops, 7,8, incorporate loops, and the code on the right-hand-side leading up to the snake is dead code (it can never be reached).
5. Answer Key: 0 - infinite loop, 1 - gold, 2 - death, 3 - gold, 4 - bomb, 5&6 - infinite loop, 7&8 - dragon, 9&10 - gold

Phase 4: General Lecture

1. As a class, the teacher will then give a lecture on the following topics.
    a. Conditionals:
        i. Start off by asking: What did you learn about conditionals?
        ii. A conditional is …
        iii. Inside of a conditional, you are going to be testing the equality of two things.
        iv. If a conditional evaluates to True, then you can enter that conditional.
        v. Question: What does it mean to enter a conditional?
    b. The difference between == and =:
        i. If you noticed, we use a double equals (==) inside of single equals sign (=) to test equality within the if statement.

Have comments, concerns, or feedback? We love to hear about ways to improve our curriculum! Fill out our Feedback Form or email cmu.teknowledge@gmail.com, and we will get back to you as soon as possible.

ii. Remember the = is used to set a variable equal to something, the == is testing if two things are equal to each other.
iii. Question: Explain back the difference between =, and ==.

c. What != means:
   i. The !=, read as 'not equal,' tests if something is not equal to something else.
   ii. != is the opposite of ==.
d. Reviewing other conditionals:
   i. Similarly, >=, <=, >, < are used as witnessed during the paper programming to test conditions.
e. Reading Conditionals:
   i. The easiest way to understand a conditional is to read it like a sentence.
   ii. Example from previous code:
      1. if (yourAnswer == realAnswer): can be read as, 'Is yourAnswer equal to realAnswer?, if so enter this conditional, if not move on.'
f. Structure/Syntax of Code:
   i. Indentation:
      1. Just like taking notes on a computer, when something is a specific subset of something else, you indent it. For today, new indentations only come after a if, elif, or else statement.
   ii. Colons:
      1. Colons come at the end of conditional statements like if, elif, and else. Any statement that ends with a colon is followed by an indented block (line(s) of code).
   iii. Parentheses:
      1. Parentheses are necessary to enclose what information you want inside of a print() or input() statement. They are optional for conditional statements like if or elif.
g. Integers:
   i. Review integers using the syntax guide.
   ii. They act like regular numbers do, and can be stored in variables as well.
h. Math operators:
   i. Review basic math operators using the syntax guide.
   ii. Math operators in code look very similar and have the same purpose to change numbers as in algebra.
   iii. Explain what +=, and *= … do and why.

Phase 5: Week One Fun Activity

1. Then, the students will complete the first few challenges in the Week One Fun file, gaining help from the Syntax Guide or from their mentors if needed.

Have comments, concerns, or feedback? We love to hear about ways to improve our curriculum! Fill out our Feedback Form or email cmu.teknowledge@gmail.com, and we will get back to you as soon as possible.

Phase 6: Optional Take Home Activity

1. The students can take home this assignment and complete the rest of the challenges below if they would like!

Phase 7: Pack up | Review

1. Mentors should lead a discussion with their students based on the question: What do you think that you can do with these tools now?
2. This question may be useful to use this as a form of review, and can also be used to increase interest in the subject.

Lesson Plan

(:10) means that this part should be done by the tenth minute of the lesson

1. Setup (:0)
2. Introduction (:10)
3. Pythons and Ladders Paper Programming (:25)
4. General Lecture (:40)
5. Week One Fun Activity (:55)
6. Optional Take Home (:55)
7. Pack up | Review (:End)

Take Away

After completing this lab, students should feel comfortable with:

1. Explaining what a conditional is
2. Knowing the difference between ==, =, and !=
3. Know what if, elif, and else are used for
4. Understand when something will evaluate to True, and thus you enter into the statement

And be able to apply some of this information to a simple unique idea, and be able to write simple code using conditionals and other tools already learned.

Have comments, concerns, or feedback? We love to hear about ways to improve our curriculum! Fill out our Feedback Form or email cmu.teknowledge@gmail.com, and we will get back to you as soon as possible.