Lab 3.1 Adventure

Instructor Guide

Overview

Students will be given a verbal review of conditionals (what they learned in the last lab) and then work through some review challenges.

Students will then move into learning about functions, and function flow through the use of the **Trailhead** activity. The day is wrapped up with students having the ability to create their own adventure game in the **My Fun Adventure** activity! Students should use the **Fun Adventure** example to help them think of what they would like to create, but there should also be a focus on making something unique to their own interests.

Learning Goals

- Creating and calling functions without parameters
  - Syntactically how to create and call functions without parameters
- Function flow in depth
- How to follow functions and what a function's purpose is
- Local variables are variables created within a function
- >=, <=, >, < are used in conditional statements to test if something is greater than or equal to, less than or equal to, greater than, and less than respectively
- A helper function is a function that is called within another function to complete a task, and make your code clearer to read
- Understand when to use helper functions

Personal Growth Goals

- Creativity: Students will have the opportunity to explore and create something unique for themselves. This is an opportunity for students to realize firsthand how diverse and unique coding can be.

Skills Required

- Complete understanding of print, input, variables, strings, conditionals, and the syntax for those skills

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

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. 03_01_01_review_challenge.py
    b. 03_01_02_trailhead.py
    c. 03__01_03_fun_adventure_example.py
    d. 03_01_04__fun_my_adventure.py

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. 03_01_01_review_challenge.py
    b. 03_01_02_trailhead.py
    c. 03__01_03_fun_adventure_example.py
    d. 03_01_04__fun_my_adventure.py

Phase 2: Introduction | Review

1. As students are entering into the classroom, verbally review with them the following topics:

    a. If-elif-else:
        i. "In what situation would you enter an if statement?"
        ii. "Give an example of when you would enter an else statement"
    b. ==,!=:
        i. What is the difference between ==, and =?
        ii. What does != mean?
        iii. Give an example of a situation where a conditional that uses != returns True.
2. Additionally, if students arrive early they can complete the review challenge.
    a. Students should look at the code and predict what would be printed out to the console if they were to run the code.

Phase 3: Trailhead Activity

1. This activity should be run by the main instructor. In this activity, the students are going to learn about functions for the first time.
2. In this activity students are going to be creating and following different trails, some already created, and some in which they need to create on their own.
3. The instructor should show the function flow of the program.
4. The following portion of the activity should be taught by a central instructor, who will write up code while his/her computer is projecting for the whole class to see:
    a. A function is code used to perform specific or multiple tasks.
        i. def functionName(): To use a function, you must first define that function.
        ii. def startTrail(): Is telling the program that there is a function called startTrail that exists.
        iii. functionName(): Calls a function
        iv. startTrail() is what calls the defined function def startTrail(): and tells the program to run that function
    b. Take the students through the function flow if 'red' is chosen as the trail.
    c. Explain why the redTrail is a new function, and where it is called from.
        i. Talk about the purposes and uses of helper functions, and how redTrail is a helper function.
5. Then have the students work through the challenges on their own, with help from their mentors if needed.
6. Before moving on, if some students could not get the bonus challenge, have a student who did get it explain how to do it, and then show it projected on the central instructor's computer.

Phase 4: Fun Adventure Activity

1. The students should spend a few minutes quickly playing through the Fun Adventure activity. They should give it a few tries, and then try to code trace through the activity to see how it was created. The purpose of this activity is to give some kind of inspiration for when students are creating their own adventure! Any questions during this activity can be directed toward their mentors or the Syntax Guide.

Phase 5: My Fun Adventure Activity

1. Students will now create their own adventure game! Based off of inspiration from the Fun Adventure activity, students should create their own adventure, and play it!
2. They can receive help from the Syntax Guide and the mentors if needed.
3. Push creating one function at a time, and testing their code abundantly to make sure their code works and isn't filled with bugs.

Phase 6: 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 | Review (:10)
3. Trailhead Activity (:25)
4. Fun Adventure Activity (:30)
5. My Fun Adventure Activity(:55)
6. Pack up | Review (:End)

Take Away

After completing this lab, students should be able to write code that can connect multiple functions together. Students should also be able to create functions, call functions, and be able to use conditionals in connection with inputs to affect the flow of a function.