Week 1 Exercises: Strings and Arithmetic (65 minutes)

- Activity Introduction
- Jupyter Interface Overview
- Ability Sorting Poll
- Breakouts

Session 1

Purpose of Today's Session

This session is an introduction to the structured study sessions and includes exercises pertaining to strings and arithmetic in Python.

HCs

Readings

Study Guide

Review the following sections from  **learnpython.org:**

- **Hello, World!**
- **Variables and Types**
- **Basic Operators**
- **String Formatting**
- **Functions**

You need not review anything related to lists unless you would like to. While reviewing, be sure to do the exercises - you cannot learn to code by just reading!

Pre-Class Work

◉ **Individual**
Group

Background Readings for Faculty

Same as for students.

Classroom Activities

Week 1 Exercises: Strings and Arithmetic 65m

NOTES FOR FACULTY

This is a problem solving session devoted to string and arithmetic exercises. Be sure to move students to appropriate breakout groups. To aid in this, use the included poll of students' ability levels before launching the breakouts. You have been provided with a "hard copy" of the exercises in the "breakout notes content," to be displayed if there are problems launching Cocalc notebooks.

Activity Introduction LAYOUT: 2-UP 5m

Briefly explain to students the format of the structured study sessions: students work on Python exercises in small breakout groups with students of similar proficiency, calling on the peer tutors for help as needed. You should be prepared to move students around in the groups as appropriate. Students may work on problems appropriate to their level, but should continue working through the entire session. Students should also periodically check in with you to report on their progress.

Note that students will be expected to know what a "function" is for the second week of Formal Analyses, so be sure to check that students can tackle the related parts of these exercises.

Activity Learning Goal Slide

Slide

Activity Learning Goal

Write Python scripts pertaining to strings and arithmetic

Jupyter Interface Overview LAYOUT: 1-UP 8m

Before administering the poll and sending students to breakouts to work on problems, give a very brief overview of the interface including the following:

- Show students how to run code using either the menu bar, or simply pressing Command-Enter.
- Show students how to create new cells.
- Distinguish between code and markdown cells. (They will not need this for the exercises, but it is important for students to know how to create text, and to turn these worksheets into a finished looking report.)

Feel free to improvise, but a copy of the worksheet has been provided for your use.

Resource

URL

Week 1 Exercises

Ability Sorting Poll LAYOUT: 1-UP 2m

The included poll should help you sort students into appropriate groups. It is desired for students of similar ability levels to work together. NOTE: THIS POLL WILL NOT AUTOMATICALLY POPULATE THE BREAKOUT GROUPS. YOU MUST TAKE NOTE OF THE RESULTS AND USE THEM TO MANUALLY SHUFFLE STUDENTS (for example, take a screenshot). Warn students that it may take a few minutes to sort them during the beginning of the breakouts.

Poll

Ability Sorting

Multiple Choice

How would you rate your Python programming skill?

-Absolute beginner

-Somewhat experienced beginner

-Intermediate

Instructor can see all responses

Breakouts LAYOUT: 1-UP 50m

Breakout

Week 1 Exercises

Groups

8 groups

unlimited

Breakout Notes

BREAKOUT NOTES CONTENT

Session 1: Strings and Arithmetic

1. **(Beginner) Input and printing:**
    1. **Use input() to request a name, then print it on the screen.**
    2. **Use input() to request a first name, then request a last name. Now print the full name with a single print statement.**
    3. **Use input() to get two integers n and m. Call print(m+n). What happened? Fix this so that the actual sum of the numbers is printed. Hint: What is the "type" of any input from input()? How can we change it?**
    4. **Now ask for an integer n, and a name. Print "yourname is printing this n times" and print that message n times without using a loop. Hint: try a newline character.**
    5. **Do the same as in (d), but each time you print, indent n spaces as well.**

6. **Define a function called print_three() that takes a string as an input and then prints the string three times. Below the function definition, use input() to request a name, then call print_three, using the name as the input to the function.**

2. **(Beginner) Arithmetic and equality:**

    1. **Request four integers, give them variable names such as m, n , p, q, and ensure that the variable values are of integer type. Compute the quantity p*(m-n)^5 and output the result.**

    2. **Repeat the above, but this time define a function named my_computation() that takes the four integers as input, and outputs (returns) the quantity p*(m-n)^5. Print this output.**

    3. **Do the same as above, but instead of computing the quantity given above, output whether or not m+n = p+q. Note: This can be done with a conditional statement, or a straightforward use of the print() function.**

3. **(Advanced Beginner) Manipulation of integers:**

    1. **Check whether an input is an integer, and output whether or not it is even. Next, check whether an integer input is divisible by 4. Hint: % operator.**

    2. **Find the sum of the digits of an inputted integer less than 1000 without reading the individual digits.** Optional: **Complete this task for any integer that is reasonable to enter by hand.**

4. **(Intermediate) Project Euler #17: If the numbers 1 to 5 are written out in words: one, two, three, four, five, then there are 3 + 3 + 5 + 4 + 4 = 19 letters used in total.**

If all the numbers from 1 to 1000 (one thousand) inclusive were written out in words, how many letters would be used? NOTE: Do not count spaces or hyphens. For example, 342 (three hundred and forty-two) contains 23 letters and 115 (one hundred and fifteen) contains 20 letters. The use of "and" when writing out numbers is in compliance with British usage.

◉ Same notes duplicated for all groups
Different notes for each group

ALL GROUPS Week 1 Exerci

After teaching all sections of this class...

please feel free to submit your feedback on this LP