Lab 5

This lab is designed to introduce you to `project.py` for P5 and introduce some fundamental looping patterns you'll be using. Completing this lab will improves your odds of success on the projects and the exams. Doing it at the designated lab time will mean you have help right away when you get stuck, and will help you make regularly scheduled progress. Enjoy!

Project API

The `project.py` file helps give you access to the dataset you'll use this week, `hurricanes.csv`. Start by looking at the dataset here: https://github.com/tylerharter/cs301-projects/blob/master/spring19/p5/hurricanes.csv. This data is a summary of statistics pulled from Wikipedia: https://en.wikipedia.org/wiki/List_of_United_States_hurricanes. Look through the dataset for a recent hurricane, such as hurricane Maria, and briefly familiarize yourself with some of the numbers. The data shows name, year, max wind speed (in MPH), damage (in dollars), and deaths. Note that the death stats are usually direct deaths, meaning they don't count deaths that occur after the storm due to, say, infrastructure damage.

Often, we'll often organize data by assigning numbers (called indexes) to different parts of the data (e.g., rows or columns in a table). In Computer Science, indexing typically starts with the number 0 (zero); i.e., when you have a sequence of things, you'll start counting them from 0 (zero) instead of 1 (one). Thus, you should **ignore the numbers show by GitHub to the left of the rows**. From the perspective of `project.py`, the indexes of Baker, Camille, and Eloise are 0, 1, and 2 respectively (and so on).

Download
hurricanes.csv
and
project.py
to a `lab5` directory that you create, and start a new notebook in that directory for some scratch work.

Run the following in cells to explore the API:

```
import project
dir(project)
```

Spend a little time reading about each of the six functions that don't begin with two underscores. E.g., run this to learn about `count`:

```
project.count.__doc__
```

You may also open up the `project.py` file directly to learn about the functions provided. E.g., you might see this:

```
def count():
    """This function will return the number of records in the dataset"""
    return len(__hurricane__)
```

You don't need to understand the code in the functions, but the strings in triple quotes (called *docstrings*) explain what each function does. As it turns out, all `project.count.__doc__` is doing is giving you the docstring for the `count` function.

Try using the project API, by running the following (in each case, make sure you have the `hurricanes.csv` file open in GitHub and find where the data returned by the function call is coming from):

1. `project.get_name(0)`
2. `project.get_year(0)`
3. `project.get_mph(0)`
4. `project.get_deaths(0)`
5. `project.get_damage(0)`
6. `project.get_damage(1)`
7. `project.get_name(project.count())`

For 6, note that the damage amount ends with "B". In this dataset, "K" represents one thousand, "M" represents one million, and "B" represents one billion. Write a function that takes a string, and returns True if (and only if) the parameter passed to it ends in one of these suffixes.

Oops, example 7 failed! Can you change the code so that you get the name of the last hurricane (in this case, "Omar")?

Loop Warmups

You're going to need to write lots of loops for this project. We'll walk you through some examples here that will help you later.

1. Using `for` and `range`

Run this snippet and observe the output:

```
i = 0
while i <= 5:
    print(i)
    i += 1
```

Your job is to replace the `???` parts below to create a loop that does the same thing:

```
for ??? in range(???):
    ???
```

Make sure the last number printed is exactly the same with both code snippets!

2. When to use `range`

Consider these two loops:

Loop A:

```
s = "bahahaha"
for x in s:
    pass # TODO
```

Loop B:

```
s = "bahahaha"
for i in range(len(s)):
    pass # TODO
```

Now imagine two different problems.

1. you need to print every letter in `s` on its own line
2. you need to print the index of every "h" in `s` on its own line

Which loop is the easier starting point for each problem? Give it a try, and discuss with your neighbour.

3. Looping over indexes

You want a loop that prints the index of every row index in `hurricanes.csv` (from 0 to 110, inclusive):

```
for idx in range(???):
    print(idx)
```

Your job is to replace the `???` parts below with a call to one of the functions in the `project` module.

3. Looping over values

Complete the following loop so it prints the name of every hurricane in the dataset:

```
for idx in range(???):
    name = ???
    print(name)
```

Both places where `???` occurs should be replaced with calls to functions in `project`.

4. Filtering data

Your job is to replace the `???` parts below so that the name of every hurricane that occurred in 2018 in printed.

```
for i in range(???):
    if ???:
        print(project.get_name(i))
```

5. Finding a maximum

Relpace `???` so that the code does what the comments say it should do:

```
def f(n):
    return 3 + n % 7

# we want to find the integer n in the range of 0 to 10
# such that f(n) is largest.
best_n = 0
for n in range(11):
    if ???:
        best_n = n

print(best_n)
```

Other Practice: Textbook and Online Exercises

If you're looking for additional practice before starting the project, you can find it online or in Think Python. Below are some suggestions for getting started.

Think Python

The textbook has an excellent suggestion to get used to strings: "Write a function that takes a string as an argument and displays the letters backward, one per line." Do this once using a `while` loop, and once using a `for` loop.

Coding Bat

The website CodingBat (https://codingbat.com/python) has great interactive coding exercises for beginners. Try to complete both the 'String-1' and 'String-2' examples. You can also practice loops with these problems:

- https://codingbat.com/python
- https://codingbat.com/prob/p193507
- https://codingbat.com/prob/p165097
- https://codingbat.com/prob/p118366