---------- Forwarded message ----------  
From: **Jonathan Soma** <[js4571@columbia.edu](mailto:js4571@columbia.edu)>  
Date: 24 May 2017 at 20:44  
Subject: Foundations Homework 2  
To: Jonathan Soma <[js4571@columbia.edu](mailto:js4571@columbia.edu)>  
Cc: Jager William Hartman <[jwh2163@columbia.edu](mailto:jwh2163@columbia.edu)>, Georgia Kromrei <[kromreig@gmail.com](mailto:kromreig@gmail.com)>

Hi everyone!

This is **one homework**, but I've split it up into **THREE SECTIONS**. Parts One and Two build up your basic skills with list and dictionaries, while Part Three actually puts them to use on Spotify data. **Read the entire homework before starting.**

Please turn it in by emailing it to me + Jager + Georgia by Sunday at midnight. I'll send you the other homeworks tomorrow or Saturday (one for more Class 1 review, the other just more of this), which will be due next Thursday.

**ADVICE**

As always, don't worry if you don't finish.

If you find Part One easy, I recommend trying Part Three. Once you hit a wall there, you can hop back to Part Two.

As you are struggling *(not if!)*, remember that the more practice you get the more skilled you'll become. And be sure to make use of TAs and Slack as much as possible, and take breaks when your brain hurts!

**Notes from class:**

I uploaded what we did in class on Slack, but there are also a few new pages on the site

**Functions and methods:**<http://jonathansoma.com/lede/foundations-2017/classes/data%20structures/functions-and-methods/>

**Lists:** <http://jonathansoma.com/lede/foundations-2017/classes/data%20structures/lists/>

**Dictionaries:** <http://jonathansoma.com/lede/foundations-2017/classes/data%20structures/dictionaries/>

**Overview of what we did:** <http://jonathansoma.com/lede/foundations-2017/classes/data%20structures/overview/>

There is content in there that we didn't cover in class, and it might be easier to read those pages instead of Stack Overflow.

*(You'll notice I mention cats named Callery and Napoleon - these are based on old notes, they both got adopted!)*

**HOMEWORK PART ONE**

Please create a new file named homework-2-part1-lastname.py

The first line should be a comment with your full name

The second line should be the date

The third line should be a comment "Homework 2, Part 1"

**PART ONE: Lists**

1) Make a list of the following numbers: 22, 90, 0, -10, 3, 22, and 48

2) Display the number of elements in the list.

3) Display the 4th element of this list.

4) Display the sum of the 2nd and 4th element of the list.

5) Display the 2nd-largest value in the list.

6) Display the last element of the original unsorted list

7) For each number, display a number: if your original number is less than 10, multiply it by thirty. If it's also even, add six. If it's greater than 50 subtract ten. If it's not negative ten, subtract one. (For example: 2 is less than 10, so 2 \* 30 = 60, 2 is also even, so 60 + 6 = 66, 2 is not negative ten, so 66 - 1 = 65.)

8) Display the sum of all of the numbers divided by two.

**PART ONE: Dictionaries**

8) Sometimes dictionaries are used to describe multiple aspects of a single object. Like, say, a movie. Define a dictionary called movie that works with the following code.

**print("My favorite movie is", movie['title'], "which was released in", movie['year'], "and was directed by", movie['director'])**

9) On the lines after that, add entries to the movie dictionary for budget and revenue (you'll use code like movie['budget'] = 1000), and print out the difference between the two.

10) If the movie cost more to make than it made in theaters, print "It was a flop". If the film's revenue was more than five times the amount it cost to make, print "That was a good investment."

11) Sometimes dictionaries are used to describe the same aspects of many different objects. Make ONE dictionary that describes the population of the boroughs of NYC. Manhattan has 1.6 million residents, Brooklyn has 2.6m, Bronx has 1.4m, Queens has 2.3m and Staten Island has 470,000. (Tip: keeping it all in either millions or thousands is a good idea)

12) Display the population of Brooklyn.

13) Display the combined population of all five boroughs.

14) Display what percent of NYC's population lives in Manhattan.

**PART ONE: Tips**

There are two ways to sort a list! One is just for display, and one sorts the list permanently. Keep an eye out for which one you're using.

Programmers are weird about counting. What number do they start with?

There might be a magic way to get the last item of a list in Python (or to start counting from the end).

When dealing with multiple numbers - population, for example - be sure to keep them all at the same level. If Brooklyn has 1.4 million people and Staten Island as 470,000, storing their population as 1.4 and 470000 isn't going to let you compare them accurately!

**HOMEWORK PART TWO**

Please create a new file named homework-2-part2-lastname.py

The first line should be a comment with your full name

The second line should be the date

The third line should be a comment "Homework 2, Part 2"

**PART TWO: Lists**

1) Make a list called "countries" - it should contain seven different countries and NOT be in alphabetical order

2) Using a for loop, print each element of the list

3) Sort the list permanently.

4) Display the first element of the list.

5) Display the second-to-last element of the list.

6) Delete one of the countries from the list using its name.

7) Using a for loop, print each country's name in all caps.

**PART TWO: Dictionaries**

These will require LATITUDE and LONGITUDE. You can ask Google for latitude and longitude by typing in \*coordinates of CITYNAME\*. You can also use <http://itouchmap.com/latlong.html>. Store the latitude and longitude without the N/S/E/W - if the latitude is S, make it negative. If the longitude is W, make it negative. See here for explanation: <https://answers.yahoo.com/question/index?qid=20080211182008AAMdUe8>

1) Make a dictionary called 'tree' that responds to 'name', 'species', 'age', 'location\_name', 'latitude' and 'longitude'. Pick a tree from: <https://en.wikipedia.org/wiki/List_of_trees>

2) Print the sentence "{name} is a {years} year old tree that is in {location\_name}"

3) The coordinates of New York City are 40.7128° N, 74.0059° W. Check to see if the tree is south of NYC, and print "The tree {name} in {location} is south of NYC" if it is. If it isn't, print "The tree {name} in {location} is north of NYC"

4) Ask the user how old they are. If they are older than the tree, display "you are {XXX} years older than {name}." If they are younger than the tree, display "{name} was {XXX} years old when you were born."

**PART TWO: Lists of dictionaries**

1) Make a list of dictionaries of five places across the world - (1) Moscow, (2) Tehran, (3) Falkland Islands, (4) Seoul, and (5) Santiago. Each dictionary should include each city's name and latitude/longitude (see note above).

2) Loop through the list, printing each city's name and whether it is above or below the equator (How do you know? Think hard about the latitude.). When you get to the Falkland Islands, also display the message "The Falkland Islands are a biogeographical part of the mild Antarctic zone," which is a sentence I stole from Wikipedia.

3) Loop through the list, printing whether each city is north of south of your tree from the previous section.

**HOMEWORK PART THREE**

Visit the following URL, then download the file by either clicking Download ZIP on the top right-hand corner or Raw on the right-hand side.

<https://gist.github.com/jsoma/8cd1b74392975201705caf52f8fd0911>

Rename the file to homework-2-part3-lastname.py

The first line should be a comment with your full name

The second line should be the date

The third line should be a comment "Homework 2, Part 3"

**PART THREE: Applying your skills**

It's more Spotify data, hooray! Explore it the same way we did in class - if you see a dictionary, examine its keys. Pick a key that might be useful, then go inside. If you see a list, take a look at the first element of the list. Keep going until you find what you're looking for.

**Note:** Don't read the Spotify API documentation, that's cheating!

These are probably not in order of difficulty, feel free to skip around.

1. How many songs are there?

2. List the name of each song, along with its popularity.

3. How long would it take, in minutes, to listen to all of these songs in a row?

4. For each song, list every artist that collaborated on it.

5. How many songs are from albums, and how many are from singles?

6. What percentage of these songs are marked as explicit?

7. I'd like to listen to one of the songs! Is there maybe a URL where I can listen to it?

**PART THREE: Tips**

If you get the error "SyntaxError: Non-ASCII character '\xe2' in file complex.py on line 4, but no encoding declared" it's because you're using Python 2

You'll probably want to use for loops, and continually add to a variable you're using for counting or totaling. But you might need to be more specific than you think: let's say I'm telling a robot to count the number of brown eggs in my egg basket. Robots are dumb, so it's important that I tell it we're starting with 0 eggs.

--

Jonathan Soma

Director, The Lede Program

Columbia University Graduate School of Journalism

2950 Broadway, New York, NY, 10027

[212-851-1816](tel:%28212%29%20851-1816) | [js4571@columbia.edu](mailto:js4571@columbia.edu)