Python Fundamentals Review

In this file, we will review all of the material that has been covered in "Python Fundamentals 1" and "Python Fundamentals 2." This is *not* for a grade and is meant to be used as a checkpoint for yourself.

The material covered in this review (and in the Python Fundamentals sections) is foundational. If you have questions, please ask. It will be important to understand this material for the rest of this course. Some of these are meant to be a little challenging to stretch you – If there are problems that you can't do, don't despair – Instead of getting worried, ask questions! As if we haven't said this enough times: IF YOU NEED HELP, PLEASE ASK QUESTIONS!!!

Types

We have discussed various types that are used within Python. This section will ask you about these types. For each question with multiple answers, circle all of the types that apply.

1. Which of the following are numerical types

- string
- list
- int
- float
- tuple
- dictionary
- bool

2. Which of the following would you typically use to store text?

- string
- list
- int
- float
- tuple
- dictionary
- bool

3. Which of the following would you use to store keys and values?

- string
- list
- int
- float
- tuple
- dictionary
- bool

4. Which of the following would you use to store items that you wanted to iterate over?

- string
- list
- int
- float
- tuple
- dictionary
- bool

5. What do each of the following types do when they are multiplied by 2?

- string
- list
- int
- float
- tuple
- dictionary
- bool

6. What is the type and value of each of the following commands? If you don't know, can you think of a way to find out?

- `2 * 3`
- `2 * 3.0`
- `"abc" * 3`
- `list("abc") * 2`
- `str(int(float("12.34")))`
- `[1, 2, 3] + ["one", "two", "three"]`
- `dict(zip([1, 2, 3], ["one", "two", "three"]))`
- `1 > 0`
- `1 < 0`

7. What do we mean when we talk about "tab completion" for a type?

Writing python code and using Spyder

The information covered in this section is probably a bit less "fundamental" than what was covered in the previous section, but knowing the answers to this section can help you write your code more efficiently.

8. What is Spyder?

9. How do you open Spyder?

10. How could you find out more information about a function in Spyder? For example, if you wanted to know more about the `print` function where could you find that information?

11. What is it when you leave notes to yourself within your code? How do you make these notes?

12. What is a cell block and how do we make one? How do we run only a single block at a time? How would we run an entire file at once?

13. How do we change our "working directory" to the data bootcamp folder?

Practice problems

These are practice problems that apply the concepts we have discussed so far

14. Suppose that we borrow 200 for one year at an interest rate of 5 percent. If we pay interest plus principal at the end of the year, what is our total payment? Compute this using variables `principal` and `i` (for interest rate). Print your answer.

15. Imagine that now you wanted to compute the interest plus principal for multiple interest rates. How would you store multiple interest rates? Now print the answer for a principal of 200 and interest rates of 1, 2, 5, and 10 percent (Hint: easiest way is to do this in a loop).

16. (Challenging) This will be your first chance at doing "text analysis" in this course. Enter the code below into Python. It is a poem by Edgar Allan Poe. It is secretly dedicated to Poe's friend. The name is written by taking one letter from each line – In particular, if you are on line i then use the ith letter (if you were on line 7 then you would use the 7th letter, not including spaces, which is s). Your job is to print the name of his friend (letter by letter is fine) that it was dedicated to. I will give two hints: use the string method for splitting (either `.split(/n)` or `.splitlines()`) to break the code up into lines and then the second hint is that you will want to ignore the spaces and punctuation (find a string method that you can use get rid of them).

```
txt = """For her this rhyme is penned, whose luminous eyes,
Brightly expressive as the twins of Loeda.
Shall find her own sweet name, that, nestling lies
Upon, the page, enwrapped from every reader.
Search narrowly the lines! - they hold a treasure
Divine-a talisman-an amulet
That must be worn at heart. Search well the measure
The words-the syllables! Do not forget
The trivialest point, or you may lose your labor!
And yet there is in this no Gordian knot
Which one might not undo without a saber,
If one could merely comprehend the plot.
Enwritten upon the leaf where now are peering
Eyes scintillating soul, there lies perdus
```

```
Three eloquent words oft uttered in the hearing
Of poets, by poets as the name is a poet's, too.
Its letters, although naturally lying
Like the knight Pinto-Mendez Ferdinando
Still form a synonym for Truth. Cease trying!
You will not read the riddle, though you do the best you can do."""
```

17. Sum all of the numbers less than 200. What is the value?

18. Create a list that computes the cumulative sum from 0 to 10 (not including 10). The answer should be `[ 0,  1,  3,  6, 10, 15, 21, 28, 36, 45]`. Save this list as `cs` which we will use in the next couple problems.

19. How would you get the 4th element (6) from `cs`?

20. How would you get the first 4 elements from `cs`?

21. How would you get the last 4 elements from `cs`?

22. Write a function which takes a first name and last name and returns the combination of the names. For example, if `firstname = "Chase"` and `lastname = "Coleman"` then the function should return `"Chase Coleman"`.

23. Write a function which takes two first names and returns the names in alphabetical order. For example, if `name1 = "Chase"` and `name2 = "Chad"` then the function should return `("Chad", "Chase")`.

24. Take the list of growth rates `g = [0.02, 0.07, 0.08]`. Write a list comprehension that multiplies each element by 100. Do the same exercise by using a loop and storing them in a new list.

25. Think about everything we've done so far. If you have questions, raise your hand or write an email to us. ASK QUESTIONS.