Homework #1 - Python Essentials

Re-submit Assignment

Due Sep 17, 2019 by 4pm **Points** 10 **Submitting** a file upload **File Types** html

Homework #1 - Python Essentials

Complete these exercises and submit a single Jupyter Notebook file (in .html format, not .ipynb) that contains your responses by **4PM on Tuesday September 17th**. Late assignments will be penalized up to 2 points per day (20%), unless prior arrangements have been made to submit the assignment after the deadline.

The notebook should be well organized. Each section should be **clearly labeled with the exercise (and part) that it addresses** (e.g., Exercise #1a, #1b, #2) in a Markdown cell block. Use (clear and concise) comments as needed to help describe each step of your process. All notebook cells that contain essential steps should be executed and the output should be visible, so as to demonstrate your successful completion of the exercise. If you cannot complete an exercise in its entirety, you should make an effort to demonstrate your intermediate progress in order to maximize partial credit, and move forward as best as possible. You may submit any written answers to the exercises in the notebook as text cells.

Academic Integrity: Each student is expected to submit his or her own original work. You may collaborate with your classmates on the concepts of the homework assignment, but you should not submit the same documentation for any part of the assignment. Submissions that contain significant similarities will be reported directly to the Office of Student Conduct.

Exercise #1 (3 Points)

One of the earliest forms of cryptography was the Caesar cipher, a form of substitution cipher in which each letter in the alphabet is shifted by a fixed number of positions (with wrapping). Caesar himself used a left shift of three (-3), but any shift can be used. For example, a Caesar cipher with a right shift of two (+2) would replace the letter 'A' with the letter 'C', the letter 'B' with the letter 'D', and so on, with the letters 'Y' and 'Z' being replaced with the letters 'A' and 'B', respectively. In such a scheme, the word 'ANALYTICS' would be encrypted as 'CPCNAVKEU'.

Step 1.1: Define a function **shift_alpha** that takes in an alphabet string (**A**) and a shift constant (**shift**, default value of 0), prints the original and shifted alphabet (each as a single string), and returns a dictionary that contains each original letter as the key and the encrypted letter as the value. For example, a shift of 1, would have the output:

```
{'A': 'B',
 'B': 'C',
 'C': 'D',
 ..more key value pairs..
 'Z': 'A'}
```

Step 1.2: Show the output of your function on the upper case alphabet on a shift of 0.

Step 1.3: Show the output of your function on the upper case alphabet on a shift of -11.

Step 1.4: Show the output of your function on the upper case alphabet on a shift of 7.

Hint: In the string module, there are built-in string constants ascii_lowercase and ascii_uppercase that contains lower case letters (a-z) and upper case letters (A-Z) that you can use in your solution. As a start, import ascii_lowercase and ascii_uppercase from the string module and print them out.

Exercise #2 (3 Points):

Step 2.1: Define a function **encrypt** that takes in a **word** and a **shift** (default value = 0), and returns a string containing the encrypted version of the word using a Caesar shift. The function should work properly on upper and lower case letters, and return any non-alphabetic characters *as is*. For example, an input of "HeLLo *" and a shift of 2 would look like:

```
encrypt("HeLLo *", 2)
```

```
Out[]: 'JgNNq *'
```

Step 2.2: Show the output of your function on "Ms. DaHLin :-)" with a shift of -2.

Step 2.3: Using a loop, show the output of your function on the following ***list*** of (case-sensitive) words/phrases with a Caesar shift of +5.

- Billie Eilish, Maryland, Route 1, LaTeX, twenty-first century

Exercise #3 (3 Points)

Now suppose you receive an *encrypted* message.

Step 3.1: Using your **encrypt** function from Exercise 2, write a function **decrypt** that takes in a **message**, and produces all possible translations of the message using a Caesar cipher. Report the decrypted message and the shift (in the range of [-12, 13]). Your output should be 26 shifted messages (including a shift of 0). For example:

```
decrypt("G jmtc Kq. Byfjgl'q ajyqq.")
```

```
Original message: G jmtc Kq. Byfjgl'q ajyqq.
1: H knud Lr. Czgkhm'r bkzrr.
2: I love Ms. Dahlin's class.
3: J mpwf Nt. Ebimjo't dmbtt.
4: K nqxg Ou. Fcjnkp'u encuu.
..more strings printed..
25: F ilsb Jp. Axeifk'p zixpp.
```

Step 3.2: Run your function on the following encrypted quotes from Greek philosophers. After each output, write in a markdown code block what the shift is that produced the decrypted message. In the example shown above, the shift that produces the decrypted message is a shift of 2.
3.2.a Aristotle: Swcnkva ku pqv cp cev, kv ku c jcdkv.
3.2.b Demosthenes: Dxlww zaazcefytetpd lcp zqepy esp mprtyytyr zq rcple pyepcactdpd.
3.2.c Hypatia: Fczy cm uh ohzifxgyhn, uhx nby zolnbyl qy nlupyf nby gily nlonb qy wuh wigjlybyhx. Ni ohxylmnuhx nby nbcham nbun uly un iol xiil cm nby vymn jlyjuluncih zil ohxylmnuhxcha nbimy nbun fcy vysihx.

HW #1 Rubric