C3

Built-In Data Structures, Functions & Files

CHAPTER ASSIGNMENT

Name _____ Class _____ No. _____ Date _____

> Answer every question in the space given. Show your working. You may use the chapter page to help you.

Questions from the Guru Jazzy chapter page (based on *Python for Data Analysis*, 3rd edition, Wes McKinney, O'Reilly).

3.1 Tuples

1. Store a student record as a tuple (name, class, score). Unpack it into three variables and print a sentence with an f-string.

2. Given scores = (55, 72, 91, 64, 88), use unpacking with *middle to get the first, last and everything in between.

3.2 Lists and Slicing

1. Before using the visualiser, predict seq[2:], seq[:-2], seq[1::3] and seq[-1:-5:-1] on paper. Then check each one.

```
seq = [7, 2, 3, 7, 5, 6, 0, 1]
```

2. Find a slice that returns an empty list even though seq is not empty. Explain why it is empty.

3. Sort a list of words by their last letter using sort(key=...) and a lambda.

3.3 Dictionaries and Sets

1. Group a list of students' names by the length of the name using setdefault, then again with defaultdict.

2. Two clubs have member lists with some overlap. Use sets to find who is in both, who is only in the first, and everyone in total.

3.4 Sequence Functions and Comprehensions

1. Given a list of names, build a dict of name → length with a dict comprehension, keeping only names with an "a" in them.

2. Flatten [(1, 2, 3), (4, 5, 6), (7, 8, 9)] into one list with a nested comprehension, then write the same thing as two for loops.

3.5 Functions

1. Add a function to clean_ops that collapses repeated spaces, so "South Carolina" comes out with one space.

```
import re

def remove_punctuation(value):
    return re.sub("[!#?]", "", value)

clean_ops = [str.strip, remove_punctuation, str.title]

def clean_strings(strings, ops):
    result = []
    for value in strings:
        for func in ops:
            value = func(value)
        result.append(value)
    return result

states = ["   Alabama ", "Georgia!", "georgia", "FlOrIda", "south   carolina##"]
print(clean_strings(states, clean_ops))
```

2. Write stats(numbers) that returns the minimum, maximum and mean, and unpack the result into three variables.

3.6 Generators and Errors

1. Notice "A" appears twice in the groupby output. Why? Fix it by sorting names by first letter first.

```
import itertools
def first_letter(x):
    return x[0]

names = ["Alan", "Adam", "Wes", "Will", "Albert", "Steven"]
for letter, group in itertools.groupby(names, first_letter):
    print(letter, list(group))
```

```
A ['Alan', 'Adam']
W ['Wes', 'Will']
A ['Albert']
S ['Steven']
```

2. Write a generator squares(n) that yields 1, 4, 9 ... up to n squared. Loop over it, then pass it to sum().

3.7 Files and the Operating System

1. Write the multiplication table for 7 to a file, one line per row, then read it back and print only the even results.

2. Open the same file in "x" mode and read the error you get. When would that behaviour protect you?

End of assignment · Check your answers against the chapter at gurujazzy.com/subjects/books/python-for-data-analysis/chapter-3-built-in-data-structures-functions-and-files/