

CMPUT 175 - Lab 1: Dictionaries, Strings, Files

Goal: Practice using Python dictionaries, formatting strings, and reading/writing to text files.

- Useful list methods: `append()`, `sort()`
- Useful string methods and operators: `format()` or `%`, `lower()`, `split()`, `upper()`, `[:]`, `+`
- Useful dictionary methods: `items()`, `keys()`, `values()`

There are many more methods, operators, and functions that you can look up in the lecture slides or in the Python 3 online documentation: <https://docs.python.org/3/index.html>

Another reference that may be helpful for understanding how to use `format()` is:

<https://www.programiz.com/python-programming/methods/string/format>

Exercise 1: Python Warm Up (must complete)

Problem: Follow the steps below to create a program for Bluebell Greenhouses that prints a purchase receipt. In the process, practice using dictionaries and formatting strings.

1. Bluebell Greenhouses sells the following Spring flower bulbs. Create a dictionary that stores this information. Which data should be the keys (must be unique), and which should be the values?

Flower Bulb Name	Price Per Bulb
daffodil	\$ 0.35
tulip	\$ 0.33
crocus	\$ 0.25
hyacinth	\$ 0.75
bluebell	\$ 0.50

2. Mary has a standing order with Bluebell Greenhouses for 50 daffodil bulbs and 100 tulip bulbs every year. Create a new dictionary that stores this information.
3. Demand for tulips this year has dramatically outpaced demand. As a result, the price of tulip bulbs has increased by 25%. Update the price of tulip bulbs in the appropriate dictionary. (Round the price per bulb to 2 decimal places.)
4. This year, Mary would also like to try planting hyacinths. Add 30 hyacinth bulbs to the dictionary that is storing her order.
5. Display Mary's purchase order for this year on the screen. Each line should be formatted as follows:

bulb code		*	number of bulbs	=	\$	subtotal
field width: 5 left-aligned			field width: 4 right-aligned			field width: 6 right-aligned 2 decimal places

where the code for each bulb name is the first three letters of its name, all in capital letters. The lines should be printed so that the bulb codes are in alphabetical order.

6. Calculate the total number of bulbs that Mary purchased this year, as well as the total cost of her order. Include this information at the bottom of her purchase order. Format the total cost float value so that it is right-aligned in a field width of 6, with 2 decimal places.

Exercise 1: Sample Output

You have purchased the following bulbs:

```
DAF    *   50 = $ 17.50
HYA    *   30 = $ 22.50
TUL    *  100 = $ 41.00
```

Thank you for purchasing 180 bulbs from Bluebell Greenhouses.
Your total comes to \$ 81.00.

Exercise 2: Student's Choice (solve either Exercise 2A or 2B)

Exercise 2A:

Create a Python program that reads the data stored in the provided **rainfall.txt**, where each line in the text file contains the name of a city, followed by whitespace, followed by the city's annual rainfall (in mm). Process this data so that it is grouped by annual rainfall into the following categories: [50-60 mm), [60-70 mm), [70-80 mm), [80-90 mm), [90-100 mm], and then sorted from lowest to highest rainfall within each category. Write this processed data to a new file called **rainfallfmt.txt**, so that under each category the city name is centered in a field that is 25 characters wide and is in all uppercase letters. The city name should be followed by its rainfall, right-aligned in a field that is 5 characters wide with 1 digit to the right of the decimal point.

Exercise 2A: Sample Output:

```
[50-60 mm)
[60-70 mm)
        AKRON           65.6
        ALTON           69.7
[70-80 mm)
        ALGONA          78.0
[80-90 mm)
        BRITT           80.1
        CARROLL         84.7
        ANKENY          84.8
...
```

Exercise 2B:

Create a Python program that reads the data stored in the provided **earthquake.txt**, where each line in the text file contains an earthquake magnitude, the date of the earthquake, the time, latitude, longitude, depth, and region, all separated by whitespace. Use the data to create lists of earthquake magnitudes and their dates, one for each region. Write these lists (in any order) to a new file called **earthquakefmt.txt**, formatted to match the sample output below.

Exercise 2B: Sample Output:

```
[ALASKA, [2006/10/19, 2.8], [2006/10/18, 2.6], [2006/10/18, 2.7],  
[2006/10/18, 2.7], [2006/10/18, 2.8]]
```

```
[HAWAII, [2006/10/19, 2.5], [2006/10/20, 3.1]]
```

```
[PANAMA, [2006/10/18, 5.0]]
```

```
[MISSOURI, [2006/10/18, 3.4]]
```

```
[INDONESIA, [2006/10/20, 4.9]]
```

```
[VANUATU, [2006/10/18, 6.2]]
```

```
[MEXICO, [2006/10/20, 2.8], [2006/10/18, 3.3]]
```