

COM4013 – Introduction to Software Development Week 6 – Lists and Good Coding Practice

Always select “Enable Editing” if prompted by Microsoft Word

Lab Exercises

*Always refer to the lecture notes for examples and guidance.
Also look at your previous work, many exercises are similar.*

THERE ARE NO STAGES IN THIS WORKSHEET

I use the term integer, string and float, and Boolean to infer what the input/output looks like. Unlike many other languages Python does not expect a data type before the variable names.

So if I ask that you declare an integer num1 to 1, then do as follows:

```
num1 = 1
```

For declaring a string and assigning the value hello

```
greeting = "hello"
```

For floats we can declare it like this

```
Money = 2.50
```

For Booleans (bool) we can declare it like this

```
isHeavy = True
```

Simple Lists

- Create a new Jupyter Notebook project called **Lab 6 Exercises**. Refer to Week 1's lecture/worksheet/video if you have forgotten how to create a new project/file/program or use the shell code.
- Create a List called `weekdays` containing the days of the week ("Sunday", "Monday", etc.). Read the lecture notes "Creating a List" or "Accessing List Elements: Examples".
- Display each of the 7 List elements using 7 `print()` statements in the style shown in the lecture notes on the slide "Accessing List Elements: Examples".
- Then use a `for` loop to display the 7 elements, following the style shown on the lecture notes slide "'for' Loops and Lists" This shows how well suited `for` loops are for Lists.

All Strings are Sequences

- Declare a variable called `firstName` and set the value to your first name.
- As the Python String is a Sequence we can access the individual letters of `firstName` by using the same indexing techniques as we do with Lists.
- Write the following code to print the first letter of `firstName`
`print(name[0]) # Prints the first letter of your name`

Strings as Lists

- Create a new cell in Jupyter Notebook (use the shell code).
- This exercises uses the `weekdays` List again (Copy the weekdays initialisation code into this cell). This time we want to display only the days that begin with 'S' or 'T'. This is easy once you realise that *strings can be used like a List of characters*:
- If we have a string called `dayString` that contains the word "Monday"
- Then `dayString[0]` is 'M', `dayString[1]` is 'o', `dayString[2]` is 'n', etc.
- Also `len(dayString)` is 6
- Use the earlier `weekdays` List (copy and paste it into the cell). However, if you created a new .py file for some reason then you will need to declare the `weekdays` List again.
- Write another `for` loop that goes through the 7 `weekdays` List elements, but instead of displaying them inside the loop write this line inside the loop (but use the correct variable names for your project):

```
dayString = weekdays[i];
```

This simply copies the weekday into a local variable so it is easier to work with

- After that line, still inside your loop, use an if statement to test whether the first character of the string is 'S' or 'T'.
- If the first character is 'S' or 'T', then display that day of the week. The correct output should be (assuming you started your week days on Sunday):

```
Sunday
Tuesday
Thursday
Saturday
```

Strings as Lists 2

- Create a new cell in Jupyter Notebook (use the shell code).
- This exercises uses the `weekdays` List again (Copy the weekdays initialisation code into this cell). This time we want to display only the first letter of each weekday within the List.
- Write a for loop to print every letter of your name on its very own line and in upper case. Use the `len()` function to work out the size of your weekdays List.

```
S
M
T
W
T
F
S
```

- Hint: You will need one `[index of the List element][index of the first letter]`

Sum of All List Elements

- Create a new cell in Jupyter Notebook (use the shell code).
- Declare a List called `numbers` and assign the following values to it:

```
1, 3, 4, 6, 7, 8, 19, 2, 4, 6, 7, 8, 13, 17, 11
```
- Using a for loop and a variable called `sum`, write a program that calculates the sum of all of the elements in the `numbers` List.
- Print the following statement to the screen, and note that **88** is not the correct answer...

```
The sum of your array elements is: 88
```

List from User Input

- Create a new cell in Jupyter Notebook (use the shell code).
- Ask the user for a List size and let them input an integer. Don't worry about validating the input, assume it will be sensible.
- Declare a List called `userList` of the size the user asked for. You need to specify the size of the List, but do not attempt to specify the actual values it contains (this needs to be an empty List).
- Use a `for` loop to step through and input each array element from the user. That means inside the loop you would need a line like this, but make sure you use the correct variable name for your code:

```
userList[i] = # Regular input code for an integer input
```

- Create a new cell in Jupyter Notebook (use the shell code).
- Write a second `for` loop to display the contents of this List *in reverse order*
 - This is similar to the earlier exercise to display a List, but this time you need a loop that decreases. **Be very careful** about choosing the start value and ending condition for this loop. The wrong choice will result in a crash or won't show all the numbers. Read the lecture notes "Out of Range Indexes" and look carefully at the range of valid indexes used. Ask your tutor if you get problems here.
- Add a simple `print()` statement after the above code.
- Optional: There is an easy way to sort a simple List like this into increasing order. However, you will need to research available Python List methods. Can you find it? If so then display the sorted List.

Review time

Use the remainder of this lab session to go back and review previous labs that you were unsure about or didn't complete. Try the advanced sections that you didn't do first time round, or attempt to reach Stage 2 to push your skill further. Also use this time to ask questions of your tutor on topics you feel weak or need extra help on.

Advanced Tasks

Plus Minus

This is an interview question that I liked the look of – You should be able to complete this one at this point of time.

HINT: You MAY need one piece of code I covered in last week's review...
Something new that I've never touched on before.

Given an array of integers, calculate the ratios of its elements that are *positive*, *negative*, and *zero*. Print the decimal value of each fraction on a new line with **6** places after the decimal.

Note: This challenge introduces precision problems. The test cases are scaled to six decimal places, though answers with absolute error of up to 10^{-4} are acceptable.

Example (This is test case 1 for this program)

`arr = [1, 1, 0, -1, -1]`

There are **n = 5** elements, two positive, two negative and one zero. Their ratios are **2/5 = 0.400000**, **2/5 = 0.400000** and **1/5 = 0.200000**. Results are printed

as:

0.400000

0.400000

0.200000

Function Description

Complete the *plusMinus* function below.

```
def plusMinus(arr):
    # Write your code here for the function specified
    pass

if __name__ == '__main__':
    arr = [1,1,0,-1,-1]
    plusMinus(arr)
```

plusMinus has the following parameter(s):

- *List arr[n]*: a List of integers

Print

Print the ratios of positive, negative and zero values in the array. Each value should be printed on a separate line with **6** digits after the decimal. The function should not return a value.

Input Format

The first line contains an integer, **n**, the size of the array.

The second line contains **n** space-separated integers that describe **arr[n]**.

Constraints

$0 < n \leq 100$

$-100 \leq \text{arr}[i] \leq 100$

Output Format

Print the following **3** lines, each to **6** decimals:

1. proportion of positive values
2. proportion of negative values
3. proportion of zeros

Sample Input (This is test case 2 for this program (arr = [-4...]))

STDIN	Function
-----	-----
6	arr[] size n = 6
-4 3 -9 0 4 1	arr = [-4, 3, -9, 0, 4, 1]

Sample Output

0.500000
0.333333
0.166667

Explanation

There are **3** positive numbers, **2** negative numbers, and **1** zero in the array.

The proportions of occurrence are positive: $3/6 = 0.500000$, negative: $2/6 = 0.333333$ and zeros: $1/6 = 0.166667$.

- ◆ Draw a **circle** of X's using nested for loops, with the user specifying the radius. The example below is illustrative, you will need a formula:

$$2\sqrt{R^2 - L^2}$$

Using this formula adds a little difficulty because $L = 0$ on the centre line, but we have to start drawing at the top line. For that reason you might want to start your loop counter at a negative value.

[illegible]

THAT IS ALL FOR NOW

Submission

1. Upload your work to the GitHub classroom - Here
 - a. [Classroom Link for Week 6](#)