Week 4 – Day 3

XP Gold

Exercise :

* Don’t forget to push on [Github](https://github.com/)

Exercise 1

1. Ask the user to type in his/her favorite fruit(s) (one or several fruits).  
   **Hint** : Use the input built in method. Ask the user to separate the fruits with a single space, eg. "apple mango cherry".
2. Store the favorite fruit(s) in a list. (How can we ‘convert’ a string of words into a list of words?).
3. Now that we have a list of fruits, ask the user to type in the name of any fruit.
   * If the user’s input is a fruit name existing in the list above, print “You chose one of your favorite fruits! Enjoy!”.
   * If the user’s input is NOT a fruit name existing in the list above, print, “You chose a new fruit. I hope you enjoy it too!”.
4. **Bonus**: Display the list in a user friendly way : add the word “**and**” before the last fruit in the list – but only if there are more than 1 favorites!

Exercise 2

1. Here is a list of popular car manufacturers: Volkswagen, Toyota, Ford Motor, Honda, Chevrolet
2. Paste it into your code, and store it in a variable.
3. Convert it into a list using Python (don’t do it by hand!).
4. Print out a message saying how many manufacturers/companies are in the list.
5. Print the list of manufacturers in reverse/descending order (Z-A).
6. Using loops or list comprehension:
   1. Find out how many manufacturers’ names have the letter ‘o’ in them.
   2. Find out how many manufacturers’ names do not have the letter ‘i’ in them.
   3. Print out the above information with meaningful output messages.
7. **Bonus:** There are a few duplicates in the list:
   1. Remove these programmatically. (Hint: you can use sets to help you).
   2. Print out the companies without duplicates, in a comma-separated list with no line-breaks (eg. “Acura, Alfa Romeo, Aston Martin, …”), and also print out a message saying how many companies are now in the list.
8. **Bonus:** Print out the list of manufacturers in ascending order (A-Z), but reverse the letters of each manufacturer’s name.

Exercise 3

1. Write a program that inserts an item at a defined index.  
   **Hint**: use a function

Exercise 4

1. Write a program that counts the number of spaces in a string.  
   **Hint**: use a function

Exercise 5

1. Write a program that calculates the number of upper case letters and lower case letters in a string.  
   **Hint**: use a function

Exercise :

* Don’t forget to push on [Github](https://github.com/)

Exercise 1

1. Given a list of integers and strings, put all the integers in one list, and all the strings in another one.

Exercise 2

1. Write a program that put each word of a string in a list without using the split function.  
   **Hint**: use a function

Exercise 3

1. Write a program that prints the longest word in a list.  
   **Hint**: use a function

Exercise 4

1. Convert a string into password format.

Example:

input : "mypassword"

output: "\*\*\*\*\*\*\*\*\*\*\*"

Exercise 5

1. Make a list called sandwich\_orders and fill it with the names of various sandwiches .
2. Then make an empty list called finished\_sandwiches. Loop through the list of sandwich orders and print a message for each order, such as I made your tuna sandwich.
3. As each sandwich is made, move it to the list of finished sandwiches.
4. After all the sandwiches have been made, print a message listing each sandwich that was made.

Exercise 6

1. Using the list sandwich\_orders from Exercise 5, make sure the sandwich ‘pastrami’ appears in the list at least three times.
2. Add code near the beginning of your program to print a message saying the deli has run out of pastrami, and then use a while loop to remove all occurrences of ‘pastrami’ from sandwich\_orders.
3. Make sure no pastrami sandwiches end up in finished\_sandwiches.

Exercise 7

1. Analyse this code before executing it. Write some commnts next to each line. Write the value of each variable and their changes, and add the final output. Try to understand the purpose of this program.

my\_list = [2, 24, 12, 354, 233]

for i in range(len(my\_list) - 1):

minimum = i

for j in range( i + 1, len(my\_list)):

if(my\_list[j] < my\_list[minimum]):

minimum = j

if(minimum != i):

my\_list[i], my\_list[minimum] = my\_list[minimum], my\_list[i]

print(my\_list)

Exercise 7

* Draw the following pattern using for loops:
* \*
* *\*\*\**
* **\*\*\*\*\***
* Draw the following pattern using for loops:
* \*
* \*\*
* *\*\*\**
* *\*\*\**\*
* **\*\*\*\*\***
* Draw the following pattern using for loops:
* \*
* \*\*
* \*\*\*
* \*\*\*\*
* \*\*\*\*\*
* \*\*\*\*\*
* \*\*\*\*
* \*\*\*
* \*\*

\*

Ninja:

Exercise 1 – Birthday Look-up

1. Create a variable called birthdays. Its value should be a dictionary.
2. Initialize this variable with birthdays of 5 people of your choice. For each entry in the dictionary, the key should be the person’s name, and the value should be their birthday. **Tip** : Use the format “YYYY/MM/DD”.
3. Print a welcome message to the user. Then tell him: “You can look up the birthday of the people in the list!”“
4. Ask the user to give you a person’s name and store his answer in a variable.
5. Get the birthday for the person’s name from the birthdays dictionary.
6. Print out the birthday with a nicely-formatted message.

Exercise 2 – Birthdays Advanced

1. Extend Exercise 1 (use a new python file).
2. Before asking the user to type in a person’s name, print out all of the names from the dictionary, to make it easier for them to choose.
3. If the person that the user types is not found in the dictionary, print an error message (“Sorry, we don’t have birthday information for “”)

Exercise 3 – Add Your Own Birthday

1. Extend Exercise 2 (in a new python file)
2. Insert this new code: before you offer the user to type a person’s name to look up, ask the user to add a birthday first:
   1. Ask the user for a person’s name – store it in a variable
   2. Ask the user for this person’s birthday (in the format “YYYY/MM/DD”) - store it in a variable.
   3. Now add this new data into your dictionary.
3. The rest of your code should follow (from Exercise 2).
4. Make sure that if the user types any name that exists in the dictionary – including the name that he entered himself – the corresponding birthday is found and displayed.

Exercise 4 – Math on a List

**!! This is the same Exercise as Day2/Exercise 1 XP NINJA). If you have already done it - Go directly to the Exercise 5 !!**

1. We want to get a list of 10 numbers from the user: Ask the user for an integer between -100 and 100 – repeat this question 10 times. Each answer from the user should be added into a list.  
   After asking the user 10 times, you should now have a list of integers.
2. Print a line(s) to separate our input section (getting the numbers from the user) from our output section (which we will describe below).
3. We will print some mathematical information about the list of numbers. Each time, print the answer with a helpful string message.  
   **NB:** While we are testing our code, it can get tedious to keep typing in 10 number each time! To save time, you can change ‘10’ to ‘2’ or ‘3’ while you are testing. Just be sure to change it back to 10 once your code works, so that you can test it more thoroughly, and so that the finished code will be correct.
   1. Print the list of numbers – printed in a single line.
   2. Print the sum of all the numbers.
   3. Print a list containing the first and the last number only.
   4. Print the numbers without any duplicates (Test this by typing in some duplicates).
   5. Print a list of all the numbers greater than 50 (There are at least 2 ways of doing questions 5-7. Can you do it in the most efficient way?)
   6. Print a list of all the numbers smaller than 10
   7. Print list of all of the numbers squared – eg. for [1, 2, 3] you would print “1 4 9”
   8. **Bonus**: Print the average of all the numbers
   9. **Bonus**: Print the largest number
   10. **Bonus**: The smallest number

Exercise 5 – Randoms

1. Extend Exercise 4 (copy it into a new file)
2. Instead of asking the user for 10 integers, generate 10 random integers yourself. Make sure that these random integers lie between -100 and 100.
3. Instead of always generating 10 integers, let the amount of integers also be random! Generate a random positive integer no smaller than 50.
4. Go back and check all of your output! Does your code work correctly for a list of unknown length, or does it only work correctly for a list that has 10 items in it