Assignment 3
Data structures

UAH, Departamento de Automática, ATC-SOL
`http://atc1.aut.uah.es`

Week 3

Objectives:

- Manipulate lists
- Manipulate tuples
- Manipulate dictionaries
- Slice notation
- Handle complex data structures

The following exercises have been collected from `http://introtopython.org/lists_tuples.html#exercis` and `http://introtopython.org/dictionaries.html`.

Exercise 1

Store the first ten letters of the alphabet in a list. With this list, perform the following tasks:

- Use a slice to print out the first three letters of the alphabet.
- Use a slice to print out any three letters from the middle of your list.
- Use a slice to print out the letters from any point in the middle of your list, to the end.

Exercise 2

Your goal in this exercise is to prove that copying a list protects the original list.

- Make a list with three people's names in it.
- Use a slice to make a copy of the entire list.
- Add at least two new names to the new copy of the list.
- Make a loop that prints out all of the names in the original list, along with a message that this is the original list.
- Make a loop that prints out all of the names in the copied list, along with a message that this is the copied list.

Exercise 3

Store a sentence in a variable, making sure you use the word Python at least twice in the sentence.

1. Use the in keyword to prove that the word Python is actually in the sentence.
2. Use the `find()` function to show where the word Python first appears in the sentence.
3. Use the `rfind()` function to show the last place Python appears in the sentence.
4. Use the `count()` function to show how many times the word Python appears in your sentence.
5. Use the `split()` function to break your sentence into a list of words. Print the raw list, and use a loop to print each word on its own line.
6. Use the `replace()` function to change Python to Ruby in your sentence.

Exercise 4

A gymnast can earn a score between 1 and 10 from each judge; nothing lower, nothing higher. All scores are integer values; there are no decimal scores from a single judge.

1. Store the possible scores a gymnast can earn from one judge in a tuple.
2. Print out the sentence, “The lowest possible score is ___, and the highest possible score is ___.” Use the values from your tuple.
3. Print out a series of sentences, “A judge can give a gymnast _ points.”
    - Don’t worry if your first sentence reads “A judge can give a gymnast 1 points.”
    - However, you get 1000 bonus internet points if you can use a for loop, and have correct grammar.

Exercise 5

Create a dictionary to hold information about pets. Each key is an animal’s name, and each value is the kind of animal. For example, ’ziggy’: ’canary’

1. Put at least 3 key-value pairs in your dictionary.
2. Use a for loop to print out a series of statements such as “Willie is a dog.”

Exercise 6

Think of a question you could ask your friends. Create a dictionary where each key is a person’s name, and each value is that person’s response to your question.

1. Store at least three responses in your dictionary.
2. Use a for loop to print out a series of statements listing each person’s name, and their response.

Exercise 7

Make a copy of your program from Pet Names.

1. Use a for loop to print out a series of statements such as “Willie is a dog.”
2. Modify one of the values in your dictionary. You could clarify to name a breed, or you could change an animal from a cat to a dog.
    a) Use a for loop to print out a series of statements such as “Willie is a dog.”
3. Add a new key-value pair to your dictionary.
    a) Use a for loop to print out a series of statements such as “Willie is a dog.”
4. Remove one of the key-value pairs from your dictionary.
    a) Use a for loop to print out a series of statements such as “Willie is a dog.”

Bonus: Use a function to do all of the looping and printing in this problem.

Exercise 8

Make a dictionary where the keys are the names of weight lifting exercises, and the values are the number of times you did that exercise.

1. Use a for loop to print out a series of statements such as “I did 10 bench presses”.
2. Modify one of the values in your dictionary, to represent doing more of that exercise.
    - Use a for loop to print out a series of statements such as “I did 10 bench presses”.
3. Add a new key-value pair to your dictionary. - - Use a for loop to print out a series of statements such as “I did 10 bench presses”.
4. Remove one of the key-value pairs from your dictionary. - - Use a for loop to print out a series of statements such as Ï did 10 bench presses".

Bonus: Use a function to do all of the looping and printing in this problem.

Exercise 9

Wikipedia has a list of the tallest mountains in the world, with each mountain’s elevation. Pick five mountains from this list.

1. Create a dictionary with the mountain names as keys, and the elevations as values.
2. Print out just the mountains’ names, by looping through the keys of your dictionary.
3. Print out just the mountains’ elevations, by looping through the values of your dictionary.

4. Print out a series of statements telling how tall each mountain is: “Everest is 8848 meters tall.”
5. Revise your output, if necessary.
   - Make sure there is an introductory sentence describing the output for each loop you write.
   - Make sure there is a blank line between each group of statements.
6. Revise your final output so that the information is listed in alphabetical order by each mountain’s name.

Exercise 9

This is an extension of exercise 8. Make sure you save this program under a different filename so that you can go back to your original program if you need to.

1. The list of tallest mountains in the world provided all elevations in meters. Convert each of these elevations to feet, given that a meter is approximately 3.28 feet. You can do these calculations by hand at this point.
2. Create a new dictionary, where the keys of the dictionary are still the mountains’ names. This time however, the values of the dictionary should be a list of each mountain’s elevation in meters, and then in feet: ’everest’: [8848, 29029]
3. Print out just the mountains’ names, by looping through the keys of your dictionary.
4. Print out just the mountains’ elevations in meters, by looping through the values of your dictionary and pulling out the first number from each list.
5. Print out just the mountains’ elevations in feet, by looping through the values of your dictionary and pulling out the second number from each list.
6. Print out a series of statements telling how tall each mountain is: .^{Ev}erest is 8848 meters tall, or 29029 feet."

Bonus: Start with your original program from Mountain Heights. Write a function that reads through the elevations in meters, and returns a list of elevations in feet. Use this list to create the nested dictionary described above.

Exercise 10

This is one more extension of exercise 8. Create a new dictionary, where the keys of the dictionary are once again the mountains’ names. This time, the values of the dictionary are another dictionary. This dictionary should contain the elevation in either meters or feet, and the range that contains the mountain. For example: ’everest’: ’elevation’: 8848, ’range’: ’himalaya’.

1. Print out just the mountains’ names.
2. Print out just the mountains’ elevations.

3. Print out just the range for each mountain.
4. Print out a series of statements that say everything you know about each mountain: “Everest is an 8848-meter tall mountain in the Himalaya range.”