# Basic Programming - Seminar 4

Alexandra Ciobotaru

3 December 2021

The scope of this seminar is to get familiar with functions, sets and dictionaries in Python and get a sense on how they are used when programming. This seminar is exercise-based.

## 1 Functions

### 1.1 Exercise 1

Remember pseudocode from Lesson 1?

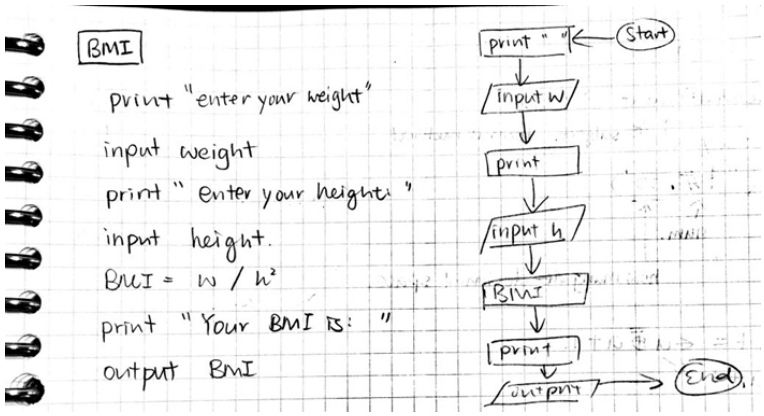


Figure 1: Compute BMI - pseudocode

Write a function that takes as an input two values: weight and height, and computes BMI using the formula:

$$BMI = \frac{weight}{height^2} \tag{1}$$

Test it by making a short script that takes weight and height as input from keyboard.

### 1.2 Exercise 2

Create a function that computes the user's BMI using the function created at Exercise 1 and compares it to the threshold value of 30. If the BMI exceeds 30, the function should return the message "It would be best for you to lose weight."

Also, if the BMI is under 18.5 the function should return the message "It would be best for you to put on some weight", and otherwise, the function should return the message "Congratulations, your weight / height ratio is perfectly normal!".

### 1.3 Exercise 3

Create a function that takes as an input a list of numbers, and computes and returns the mean of those numbers, using the formula:

$$mean = \frac{sum}{length} \tag{2}$$

You can use `sum` and `len` built-in methods for computing this formula.

Test your function with a list of numbers.

Compute the mean by importing the statistics module (`import statistics`) and using the `statistics.mean()` method. Are both your computations the same?

[Optional] Compute median using `statistics.median()` method.

## 2 Sets

### 2.1 Exercise 4

Create a set called "fruits" that contains the following strings: "apple", "banana", "cherry". Check if "apple" is present in the fruits set, and if it is present, print the following statement: "Yes, apple is a fruit!".

### 2.2 Exercise 5

Create a function that takes as an input two sets and returns the intersection of those sets (a new set of identical items from the two sets). Test it using the sets listed in the image listed in Figure 2.

Hint: Use the `intersection()` method of a set.

### 2.3 Exercise 6

Create a function that takes as an input two sets and returns the reunion of those sets (a new set containing only unique items from two sets). Test it using

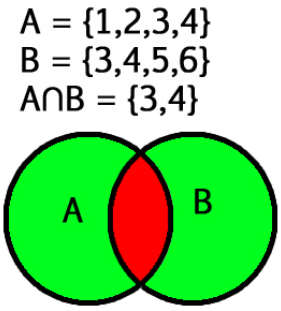


Figure 2: Intersection of sets

the sets listed in the image listed in Figure 3.

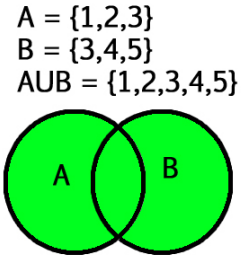


Figure 3: Reunion of sets

Hint: Use the `union()` method of a set.

# 3 Dictionaries

## 3.1 Exercise 7

Convert two lists into a dictionary. Below are the two lists. Write a Python program to convert them into a dictionary in a way that the item from list1 is the key and the item from list2 is the value:

```
keys = ['Ten', 'Twenty', 'Thirty']
values = [10, 20, 30]
```

Expected output:

```
{'Ten': 10, 'Twenty': 20, 'Thirty': 30}
```

Hint: use the `zip` function to aggregate the lists, and then cast it into a dictionary.

### 3.2 Exercise 8

Write a script that adds a key to a dictionary.

```
Sample Dictionary : {0: 10, 1: 20}
Expected Result : {0: 10, 1: 20, 2: 30}
```

Hint: use the `update()` method.

### 3.3 Exercise 9

Write a Python program to concatenate the following dictionaries to create a new one:

```
dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50, 6:60}
```

Expected Result:

```
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

Hint: create an accumulator (empty dictionary) and append to it by iterating over the dictionaries in a for loop.

### 3.4 Exercise 10

Write a Python function that checks whether a given key already exists in a dictionary. Test it using the following dictionary:

```
d = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

Hint: write a condition to test if the key is in the dictionary and return the response accordingly.

Congratulations, you have finished your fourth seminar! You're getting closer to mastering Python!