Assignment 2

Status: Completed
Your identity is visible during marking.

Deadline: 17 Sep 2018, 23:55

Weekly assignment A2

Part A

What type of values are these?

1. `3.25 + 4`
2. `["abc", 34, 4.34, 23]`
3. `True`
4. `"Hello" + "World"`

Part B

Choose the Data Type

Choose a suitable data type to represent the following values, justify your selection.

1. Month name
2. Week number
3. Recorded temperatures on different days of a month
4. Average temperature in a month

Part C

Tracking Errors

Point out errors and explain them, if any, in the following program segments without executing them.

1. ```
   >>> a = 5
   >>> b = 'string'
   >>> c = 5.9
   ```
   ```

```
>>> print(a + " squared equals " + b + " squared " + c + " squared.")
```

2. ```
   >>> a_list = [4,5,6,7,8,9]
   >>> print(a_list['0'])
   ```

3. ```
   >>> b = 3
   >>> c = 4
   >>> (b**2 + c**2)**0.5 = d
   >>> print(d)
   ```

4. ```
   >>> my_list = [23, "hi", 2.4]
   >>> for item in my_list:
   ...  print(item)
   ```

5. ```
   >>> num=24
   >>> a=34.999
   >>> result=num*(13/a**2)+1.0
   >>> print("Result:"result)
   ```

Part D

Write Programs

1. Write a program which will find and print a list of all such integers which are divisible by 7 but are not multiple of 5 from a list of integers in the range 1000 to 2000 (both included). Solve this exercise using a *for*-loop, and not a list comprehension. Hint: To obtain a list of integers between 1000 and 2000, you can use the built-in `range()` function, e.g.:

   ```
   a_list = list(range(1000,2001))
   ```

2. Suppose you are given the following list:

   ```
   a_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
   ```

   Write a program to compute a list of cubes of the items in the above list using the list comprehension. The output of the program should be:

   ```
   [1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]
   ```

3. If you are given three sticks, you may or may not be able to arrange them in a triangle. For example, if one of the sticks is 12 inches long and the other two are one inch long, you will not

be able to get the short sticks to meet in the middle. For any three lengths, there is a simple test to see if it is possible to form a triangle:

If any of the three lengths is greater than the sum of the other two, then you cannot form a triangle. Otherwise, you can. (If the sum of two lengths equals the third, they form what is called a “degenerate” triangle.)

Write a program which takes lengths of three sticks from the keyboard and then prints either “Yes” or “No”, depending on whether you can or cannot form a triangle from sticks with the entered lengths.

Submission

For part A, B, and C produce and upload one (pdf or txt) file (containing all parts), and for part D upload the script files (i.e. py files) one for each of the programs. Name the files for Part D like `partD_1.py`, `partD_2.py`, and `partD_3.py`

Best of Luck!

Hemanth Kumar Battula , 14 Sep 2018 15:37

File name: PartABC.txt (2,1 KB)

Status set to: To be marked

Hemanth Kumar Battula , 14 Sep 2018 15:38

File name: PartD_1.py (159 B)

Status set to: To be marked

Hemanth Kumar Battula , 14 Sep 2018 15:38

File name: PartD_2.py (73 B)

Status set to: To be marked

Hemanth Kumar Battula , 14 Sep 2018 15:38

File name: PartD_3.py (433 B)

Status set to: To be marked

Chatrine Qwaider , 25 Sep 2018 13:16

Status set to: Completed

Comment: Thanks for well-done work.