# Tutorial 07 – Lists

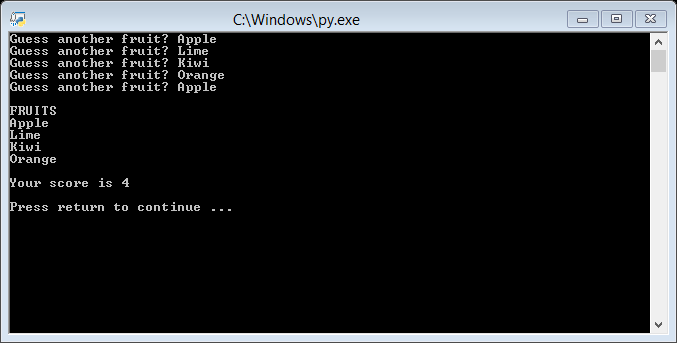
This tutorial is aimed at familiarising you with using lists. It will also test you on the use of list-based functions and loops.

Do not worry if you can’t do all the exercises, especially the difficult ones. Give them a try and if you are stuck, ask your tutor.

For each exercise you should make a copy of the example you are working on **before you modify it**. In that way you will have both your new version and the original version to compare it with.

You should make a record all of your work in your COMP1753 logbook.

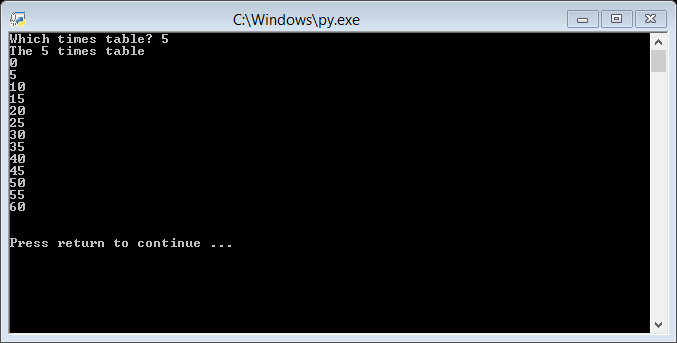
1. Download the examples and last week’s solutions. Check through the solutions and make sure you could do them all**.** Unzip this week’s examples.
2. Make a copy of 06Fruits\_listItemMethods and change the name to 02Fruits\_favourites. Now modify it so that it prompts the user for 2 extra fruits – their favourite fruit which is inserted at the start of the list and their least favourite fruit which is added to the end. The list should be printed out twice, the first time with the order unchanged and the second time sorted.
3. Write a program which prompts the user for their name and then prints the output ‘Hello [name], your favourite fruit is [fruit]’ where [name] is input by the user and [fruit] is picked from a list of fruit names, using a randomly generated index. Run it a number of times to check that it works and the randomly generated index always corresponds to a list item.
4. Now build a program which starts with an empty list and uses a while loop to repeatedly prompt the user for a new fruit which will be added to the list. If the fruit is already in the list, or if the user enters an empty string, the loop should terminate. When the loop terminates, print out the list of fruits and the user’s score, which is the number of different fruits they have identified. [Hint: use while True: to run the loop indefinitely and break to terminate it.] NB this could be the basis of a simple game if, in addition, it checked that the user’s fruits are genuine. The output might look like this:



1. Write the following two Python functions which are passed a list as a parameter:
   1. **first()** should return the first item in the list;
   2. **last()** should return the last item in the list.

Now make a copy of 05Fruits\_listMethods and include your two methods. Then modify the output so that it just prints the first and last items of the fruits list, rather than the whole list, after the list has sorted or reversed. [Note that the first() and last() functions don’t need to work if the list is empty.]

1. Make a copy of 07Randoms\_sorted and change the name to 06TimesTable. Now modify it so it prompts the user for a number between 1 and 10, calculates the corresponding times table, stored in a list, and then prints the list out using print\_list(). For example, if the user enters 5 at the prompt, the program will calculate the 5 times table with the following output (13 values, starting from 0 times 5 and ending with 12 times 5):



1. Using list slicing, write the following two Python functions which are passed a list as a parameter:
   1. **even()** should return the even numbered items in the list (starting from 0);
   2. **odd()** should return the odd numbered items in the list (starting from 1).

Now make a copy of your solution to exercise 6 and include these two functions to print the even and odd values in the times table.

1. Read the w3schools pages mentioned in the lecture:
   1. <https://www.w3schools.com/python/python_lists.asp>
   2. <https://www.w3schools.com/python/python_sets.asp>