TMTplus Introduction to Scientific Programming

Mahdi Farnaghi, Mahdi Khodadadzadeh & Robert Ohuru

March 2021

Note from the teaching staff

The materials that we use for this course are somewhat new and have been heavily reworked. We will likely have made mistakes in this work or have glossed over issues that deserved better or different treatment. Where you find issues worth noting please do report these to us as it will allow to repair and improve.

By the way, welcome to the Exercise book for Introduction to Scientific Programming! We believe this book is self-explanatory, so go ahead and practice.

Exercise 4

Lists & iteration

Computer programs are especially suited to conduct repetitive tasks consistently. Human beings normally grow tired and bored on such work, but computers do not. A common case where repetitive tasks occur is when we have multiple data elements in some kind of container. The list as discussed in class is a prominent container type in Python. We will later see how versatile lists actually are. So, this exercise set is about lists and their elements, and how to successively work on the elements.

The exercises

The main goal of each exercise is to use the Python programming language to review concepts you have learned during the lectures.

More than before, we encourage you to experiment with the code that you have in your PyCharm projects, and go beyond what we ask you as exercise. Try pose yourself the question what else you could do to learn further.

4.1 Working with list expressions

Ex 4.1 Given the list of movies

```
movies = [ "The␣Addams␣Family", "Ghostbusters", "Jurassic␣Park",
           "Pulp␣Fiction", "Home␣Alone", "The␣Matrix"]
```

complete the table below.

index	movies[index]
0	The Addams Family
4	
5	
-1	
-6	
-10	
9	

Ex 4.2

Which are the movies that correspond with the following sliced segments?

1. `movies[0:2]`
2. `movies[3:5]`
3. `movies[:]`
4. `movies[3:]`
5. `movies[:5]`
6. `movies[-1:]`
7. `movies[-3:]`
8. `movies[-4:-2]`
9. `movies[-4:5]`
10. `movies[-4:0]`
11. `movies[9:-4]`

Ex 4.3

It is possible to use steps in slicing with `[start:stop:step]`. With this in mind, determine the movie lists produced in these slides:

1. `movies[0::2]`
2. `movies[::3]`
3. `movies[0:5:1]`
4. `movies[0:5:2]`
5. `movies[0:5:]`

6. `movies[-4::3]`
7. `movies[-5:-1:2]`
8. `movies[-3:-9:1]`
9. `movies[::1]`
10. `movies[::-1]`
11. `movies[::2]`
12. `movies[::-2]`

Ex 4.4 The `in` keyword allow us to check whether an element occurs in a list. Please run the following code in context of the movies list, and verify the results. What do you conclude?

```
print('Star Wars' in movies)
print('Jurassic' in movies)
print('Jurassic Park' in movies)
print('jurassic park' in movies)
print('Ghostbusters' in movies)
```

Ex 4.5 Lists are mutable, which means that we can change them. Please change the movies list according to the below requirements. Note: Refer to the lecture slides on lists.

- Replace the Jurassic Park movie by the movie Star Wars.
- Replace the last three movies by Gladiator, Casablanca and The Godfather.

Ex 4.6 Lists have methods too, and these allow us to append, insert, and remove members t/from the list. Please change the movies list using the methods for the below requirements.

- Append movie The Shawshank Redemption.
- Extend the movies list with the following: Forrest Gump, The Dark Knight, Fight Club, and Saving Private Ryan.
- Insert movie Back to the Future into position 0.
- Remove movie Forrest Gump from the movies list.

Ex 4.7 The `del` keyword can be used to delete an item or a group of items from a list. With the information of slides from the lectures, please delete the following items:

- `movies[0]`
- `movies[0:2]`
- `movies[:]`
- delete variable `movies`

4.2 More advanced list expressions

Ex 4.8

Scrutinize the following list of lists:

```
tv_shows = [    ["Sherlock␣Holmes", "Drama" ,2009],
                ["Planet␣Earth", "Documentary", 2006],
                ["Cosmos", "Documentary", 2014],
                ["Dr.␣Who","Science␣fiction", 2005],
                ["Stranger␣Things","Science␣fiction", 2016],
                ["Game␣of␣Thrones","Fantasy", 2011]
           ]
```

Now, print the information that corresponds with to the following index expressions.

- `tv_shows[1]`
- `tv_shows[1:3]`
- `tv_shows[2][0]`
- `tv_shows[2][1]`
- `tv_shows[2][2]`
- `tv_shows[2][1:]`
- `tv_shows[2][:]`

Ex 4.9

Tuples are like lists but they are immutable; you cannot change a data member, can you append or remove one. Once defined, they cannot be changed. Please try to run the following script.

```
movies = ("The␣Addams␣Family", "Ghostbusters", "Jurassic␣Park",
          "Pulp␣Fiction", "Home␣Alone", "The␣Matrix")
print(movies[2])
movies[2]="Star␣Wars"
```

What are your conclusions?