Challenge 1

First of all I had to install mongodb on my mac via brew. Once the local mongodb installation was over I started the service using the command 'brew services start mongodb-community@5.0'

Afterwards, I tested the routine and the script using the Jupyter notebook.

Step 1

Import the libraries

In [1]:
```
import pymongo
```

In [3]:
```
from pymongo import MongoClient
```

Step 2

set up a collection into "miramar database". We created several entries (documents) to meet the challenge goals

In [3]:
```
client = MongoClient ('localhost',27017)
```

In [4]:
```
db=client["miramar"]
mycol = db["SCANNER11"]
input1 = [{ "deviceid": "h23ab", "timestamp": "1583934256", "devices":"22345q789123456a" },
          { "deviceid": "z23ae", "timestamp": "1583934261", "devices":"123456789123456a" },
          { "deviceid": "d23ab", "timestamp": "1583934262", "devices":"1f3456789123456a" },
          { "deviceid": "823ae", "timestamp": "1583934082", "devices":"223456789123456a" },
          { "deviceid": "a23ab", "timestamp": "1583934081", "devices":"22345q789123456a" },
          { "deviceid": "b23ab", "timestamp": "1583934072", "devices":"123456789123456a" },
          { "deviceid": "c23ab", "timestamp": "1583934022", "devices":"1234567rty23456a" },
          { "deviceid": "d23ab", "timestamp": "1583934029", "devices":"123qw6789123456a" },
          { "deviceid": "e23ab", "timestamp": "1583934009", "devices":"123456789123456a" },
          { "deviceid": "f23ab", "timestamp": "1583934007", "devices":"12345678912yiu6a" },
          { "deviceid": "g23ab", "timestamp": "1583934007", "devices":"123456789rt3456a" },
          { "deviceid": "f23ab", "timestamp": "1583934001", "devices":"1234qq789123456a" },
          { "deviceid": "123ab", "timestamp": "1583934000", "devices":"1234567891poo56a" }]
x = mycol.insert_many(input1)
```

Step 3

Test that we have a cursor to work with

In [5]:
```
mycol.find({})
```

Out[5]:
```
<pymongo.cursor.Cursor at 0x7fc8acd842d0>
```

Step 4

create a simple function that can be used to iterate the database. The idea is firts to have an empty container. Afterwards we iterate the collection to store all the 'devices' strings inside the db.

In [6]:
```
lista_devices=[]
for i in list(mycol.find({})):
    lista_devices.append(i['devices'])
```

check the output

In [7]:
```
lista_devices
```

Out[7]:
```
['22345q789123456a',
 '123456789123456a',
 '1f3456789123456a',
 '223456789123456a',
 '22345q789123456a',
 '123456789123456a',
 '1234567rty23456a',
 '123qw6789123456a',
 '123456789123456a',
 '12345678912yiu6a',
 '123456789rt3456a',
 '1234qq789123456a',
 '1234567891poo56a']
```

check only unique 'devices' strings

In [8]:
```
list(set(lista_devices))
```

Out[8]:
```
['123456789123456a',
 '123456789rt3456a',
 '1234567891poo56a',
 '12345678912yiu6a',
 '22345q789123456a',
 '123qw6789123456a',
 '1234qq789123456a',
 '1f3456789123456a',
 '1234567rty23456a',
 '223456789123456a']
```

Step 5

create a printout to visualize the differences between actual db entries and removed duplicates

In [9]:
```
print('number of devices in database: ' + str(len(lista_devices)))
print('number of unique devices in database: ' + str(len(set(lista_devices))))
```

```
number of devices in database: 13
number of unique devices in database: 10
```

Step 6

export the mongodb collection by using the command from shell terminal 'mongoexport --collection=SCANNER11 --db=miramar --out=events.json'

The json file has been placed into the delivarable folder

Step 7

create a python script which connect to the db and return a list accordingly to specs

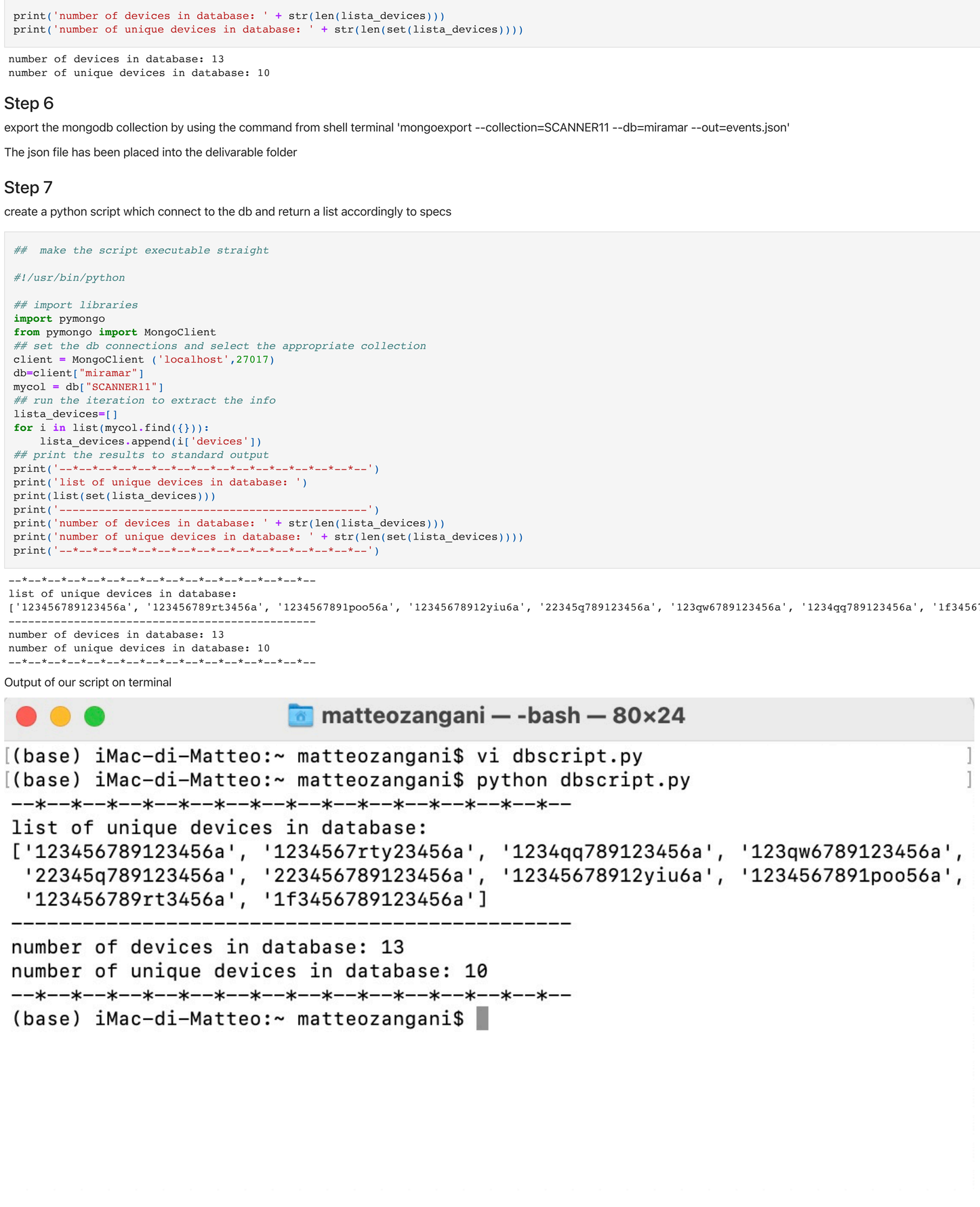
In [15]:
```
##  make the script executable straight

#!/usr/bin/python

## import libraries
import pymongo
from pymongo import MongoClient
## set the db connections and select the appropriate collection
client = MongoClient ('localhost',27017)
db=client["miramar"]
mycol = db["SCANNER11"]
## run the iteration to extract the info
lista_devices=[]
for i in list(mycol.find({})):
    lista_devices.append(i['devices'])
## print the results to standard output
print('--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--')
print('list of unique devices in database: ')
print(list(set(lista_devices)))
print('-----------------------------------------------')
print('number of devices in database: ' + str(len(lista_devices)))
print('number of unique devices in database: ' + str(len(set(lista_devices))))
print('--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--')
```

```
--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--
list of unique devices in database:
['123456789123456a', '123456789rt3456a', '1234567891poo56a', '12345678912yiu6a', '22345q789123456a', '123qw6789123456a', '1234qq789123456a', '1f3456789123456a', '1234567rty23456a', '223456789123456a']
-----------------------------------------------
number of devices in database: 13
number of unique devices in database: 10
--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--
```

Output of our script on terminal



Datasets dropped inside the google drive folder

In [ ]:
```
databae in json format events.json
script dbscript.py
pdf of jupyter notebook
jupyter notebook ipynb format
```