IMD0905 - Data Science I

Lesson #4 - Introduction to Python I

Ivanovitch Silva

August, 2018

Agenda

- Modules
- Iterations
- List comprehension

top100.csv

Spotify®

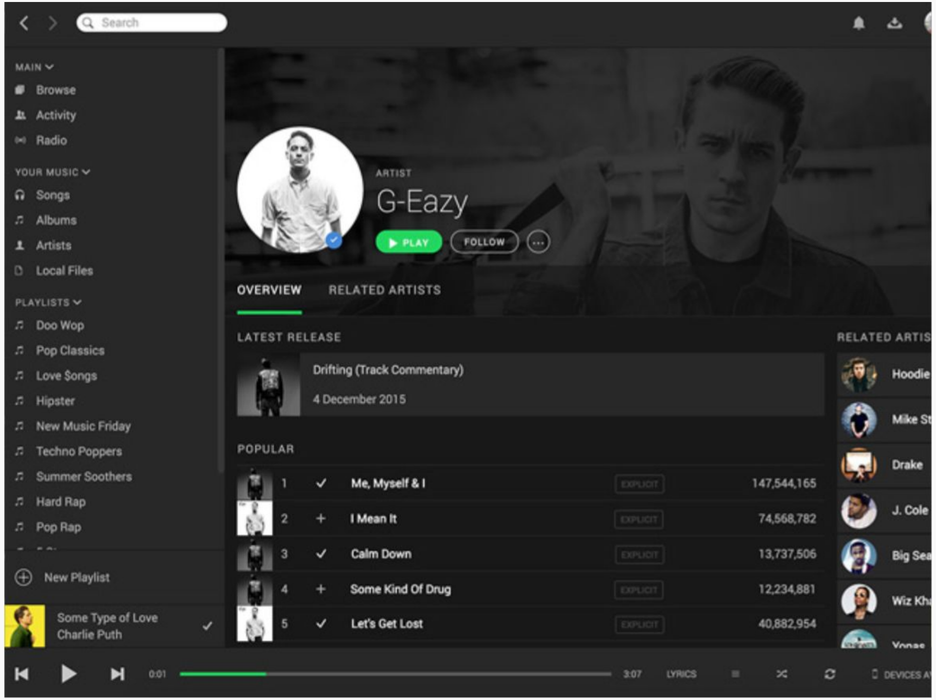
Update the repository

```
git clone https://github.com/ivanovitchm/IMD0905_datascience_one.git
```

Or ….

```
git pull
```

Spotify



Drifting (Track Commentary)
4 December 2015

POPULAR

1	✓	Me, Myself & I	EXPLICIT	147,544,165
2	+	I Mean It	EXPLICIT	74,568,782
3	✓	Calm Down	EXPLICIT	13,737,506
4	+	Some Kind Of Drug	EXPLICIT	12,234,881
5	✓	Let's Get Lost	EXPLICIT	40,882,954

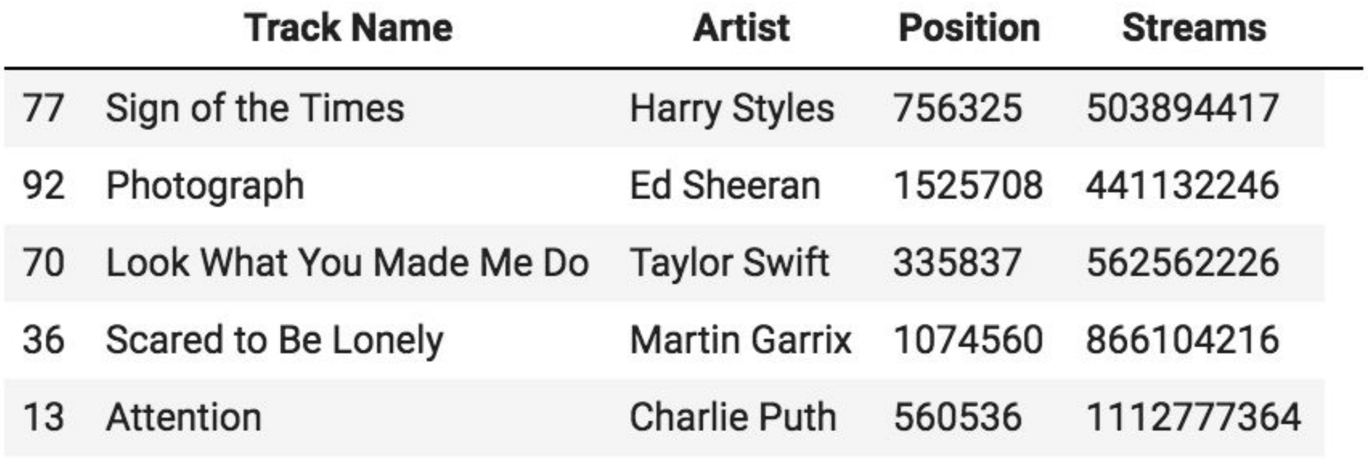
As of January 2018, Spotify has over 70 million paying users

What are the average total streams for each song in the top 100?

Which song was the most popular song of 2017?

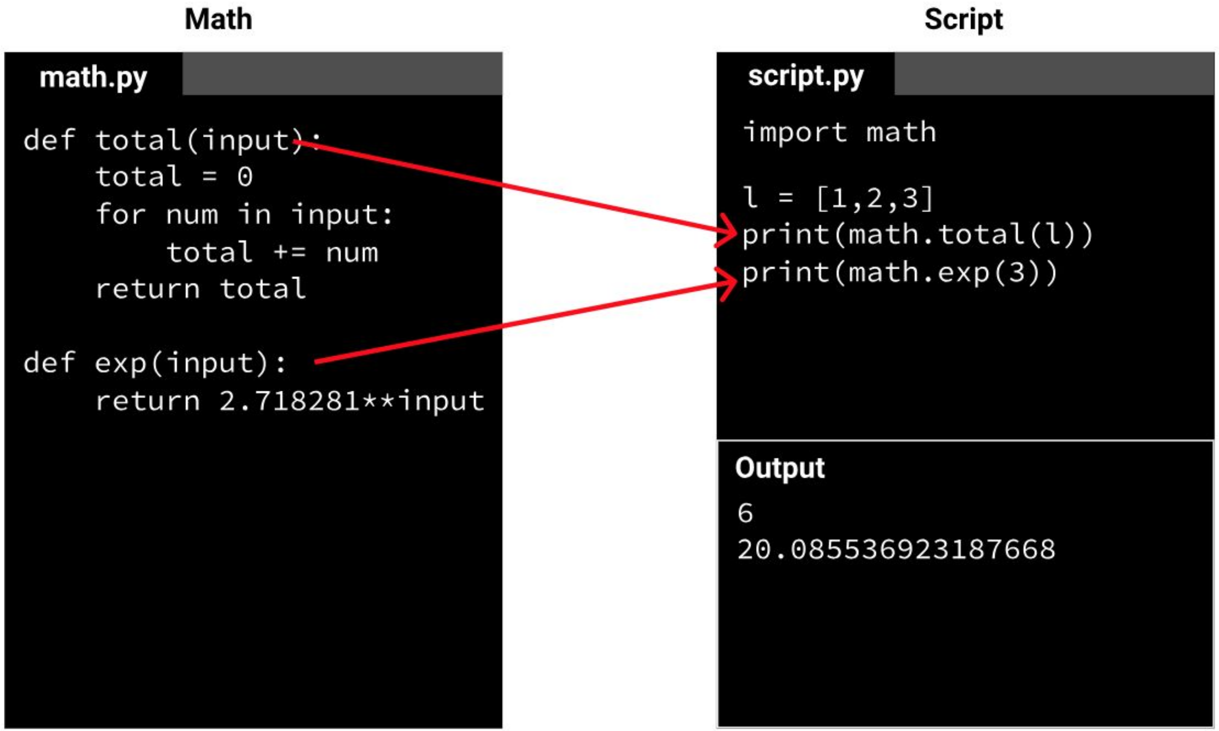
Spotify's Worldwide Daily Song Ranking

kaggle



	Track Name	Artist	Position	Streams
77	Sign of the Times	Harry Styles	756325	503894417
92	Photograph	Ed Sheeran	1525708	441132246
70	Look What You Made Me Do	Taylor Swift	335837	562562226
36	Scared to Be Lonely	Martin Garrix	1074560	866104216
13	Attention	Charlie Puth	560536	1112777364

Introduction to Modules



Math

math.py

```
def total(input):
    total = 0
    for num in input:
        total += num
    return total

def exp(input):
    return 2.718281**input
```

Script

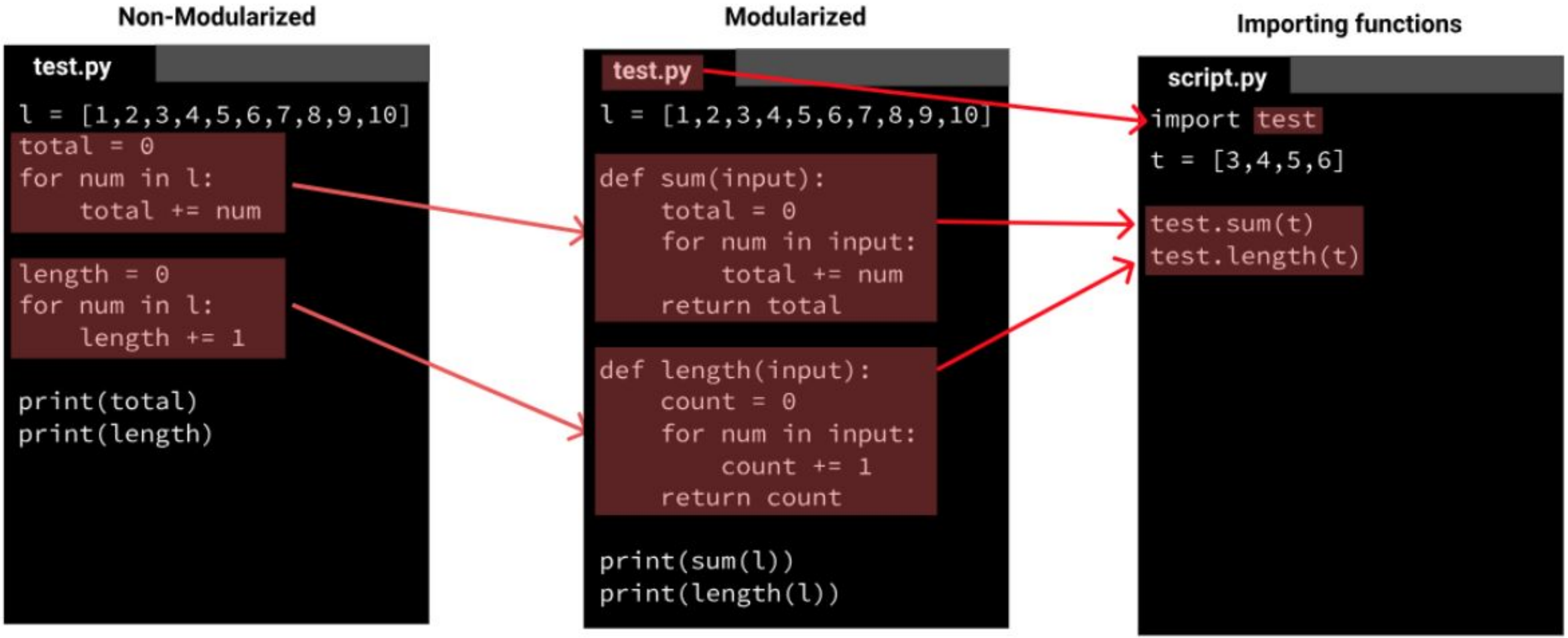
script.py

```
import math

l = [1,2,3]
print(math.total(l))
print(math.exp(3))
```

Output

```
6
20.085536923187668
```

Introduction to Modules



Non-Modularized

test.py

```
l = [1,2,3,4,5,6,7,8,9,10]
total = 0
for num in l:
    total += num

length = 0
for num in l:
    length += 1

print(total)
print(length)
```

Modularized

test.py

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

def sum(input):
    total = 0
    for num in input:
        total += num
    return total

def length(input):
    count = 0
    for num in input:
        count += 1
    return count

print(sum(l))
print(length(l))
```

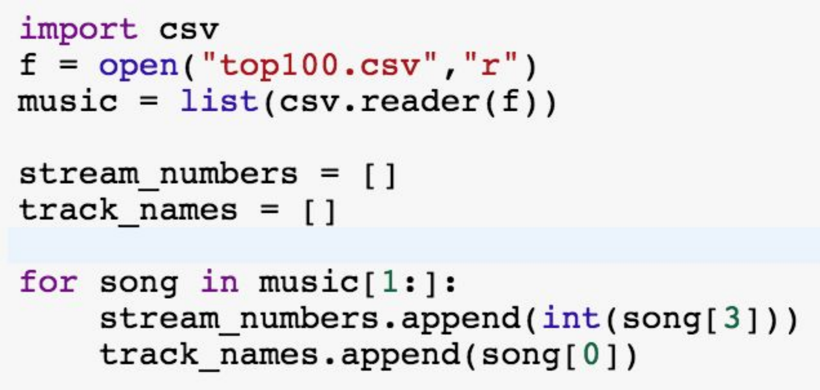
Importing functions

script.py

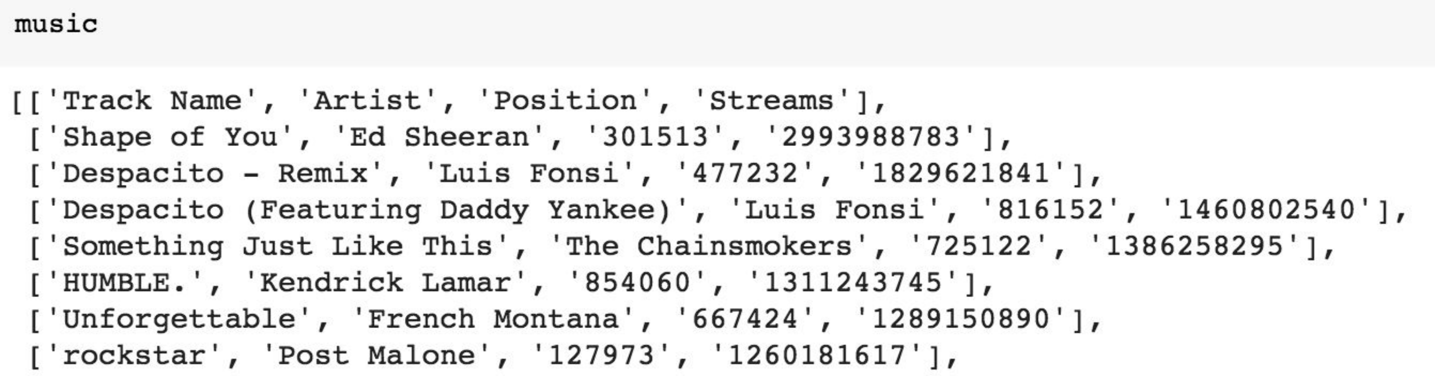
```
import test
t = [3,4,5,6]

test.sum(t)
test.length(t)
```

Introduction to Modules

Sections
2.1 - 2.6



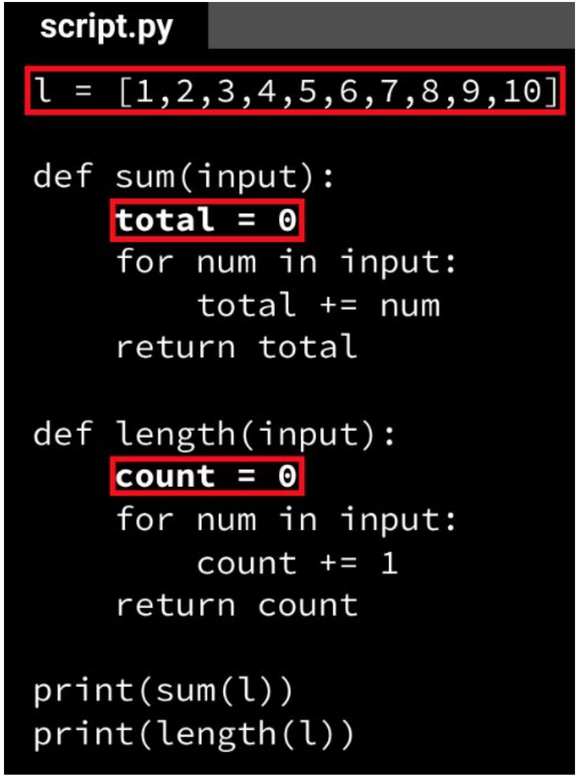
```
import csv
f = open("top100.csv","r")
music = list(csv.reader(f))

stream_numbers = []
track_names = []

for song in music[1:]:
    stream_numbers.append(int(song[3]))
    track_names.append(song[0])
```



```
music
```

```
[['Track Name', 'Artist', 'Position', 'Streams'],
 ['Shape of You', 'Ed Sheeran', '301513', '2993988783'],
 ['Despacito - Remix', 'Luis Fonsi', '477232', '1829621841'],
 ['Despacito (Featuring Daddy Yankee)', 'Luis Fonsi', '816152', '1460802540'],
 ['Something Just Like This', 'The Chainsmokers', '725122', '1386258295'],
 ['HUMBLE.', 'Kendrick Lamar', '854060', '1311243745'],
 ['Unforgettable', 'French Montana', '667424', '1289150890'],
 ['rockstar', 'Post Malone', '127973', '1260181617'],
```

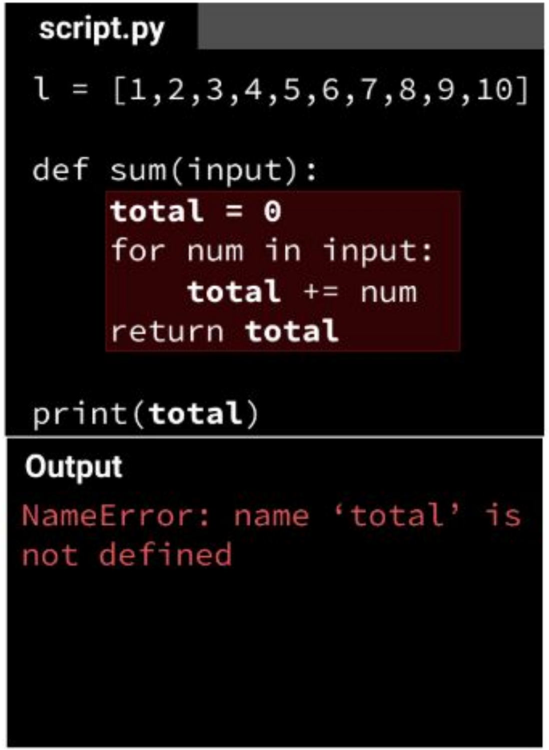
Local and Global Variables



script.py

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

def sum(input):
    total = 0
    for num in input:
        total += num
    return total

def length(input):
    count = 0
    for num in input:
        count += 1
    return count

print(sum(l))
print(length(l))
```

🟥 Accessible Area



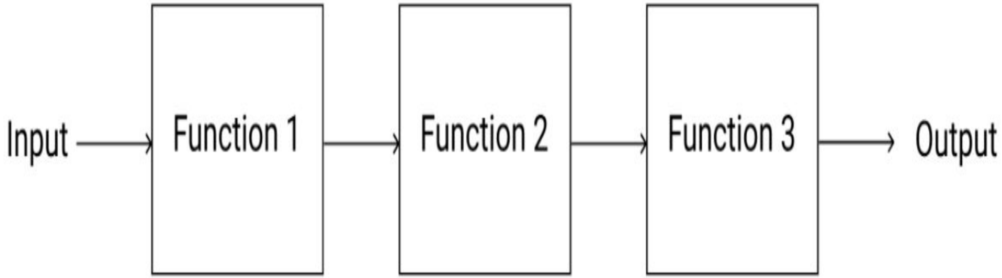
script.py

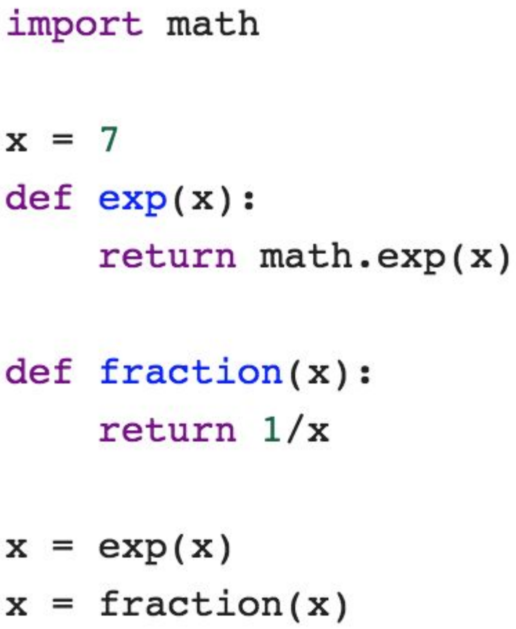
```python
l = [1,2,3,4,5,6,7,8,9,10]

def sum(input):
    total = 0
    for num in input:
        total += num
    return total

print(total)
```

Output

```
NameError: name ‘total’ is
not defined
```

Using programming paradigms (functional)

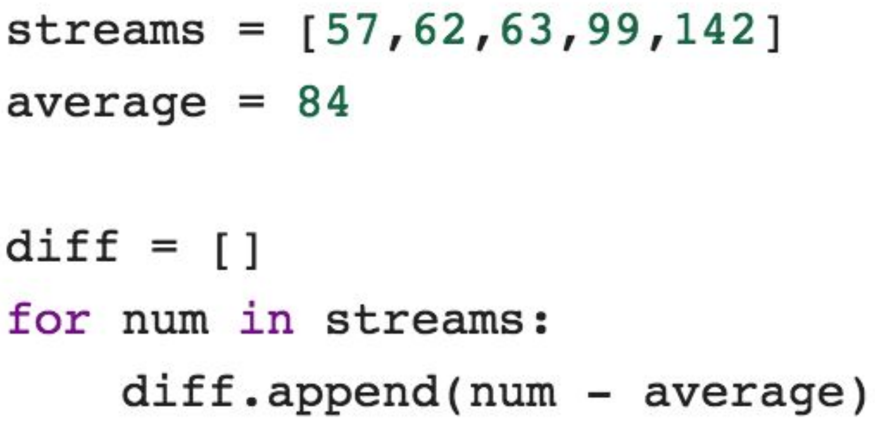




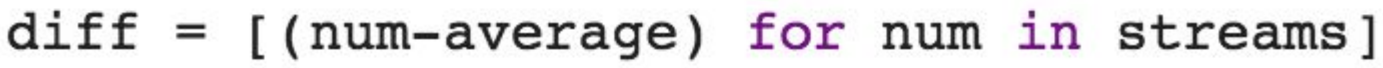
```
import math

x = 7
def exp(x):
    return math.exp(x)

def fraction(x):
    return 1/x

x = exp(x)
x = fraction(x)
```

Sections
2.7 - 2.10

List Comprehension

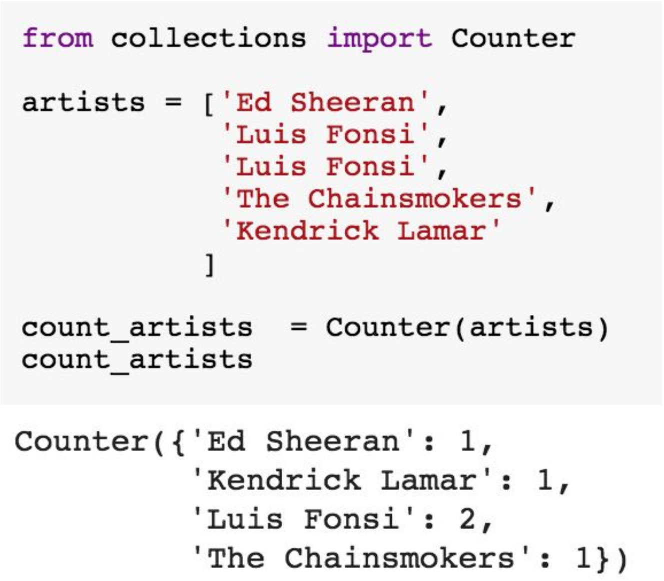


```
streams = [57,62,63,99,142]
average = 84

diff = []
for num in streams:
    diff.append(num - average)
```



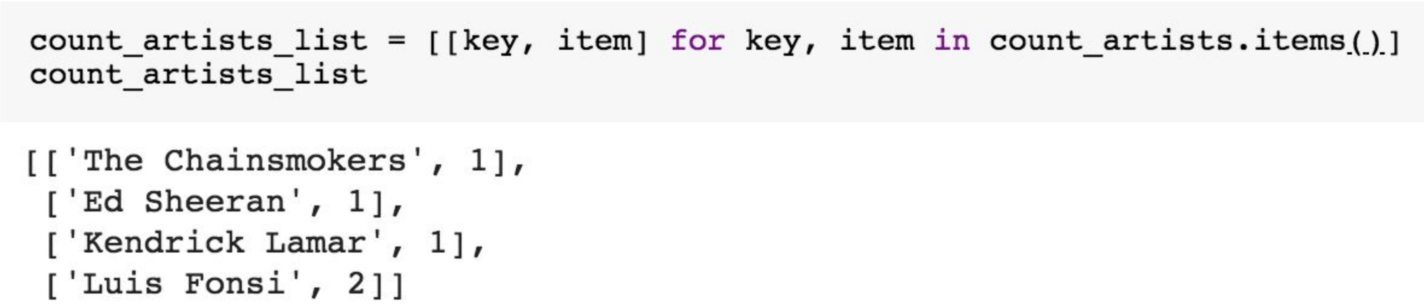
```
diff = [(num-average) for num in streams]
```

Getting the Artist Count Using a Function and List Comprehension



```
from collections import Counter

artists = ['Ed Sheeran',
           'Luis Fonsi',
           'Luis Fonsi',
           'The Chainsmokers',
           'Kendrick Lamar'
          ]

count_artists  = Counter(artists)
count_artists
```

```
Counter({'Ed Sheeran': 1,
         'Kendrick Lamar': 1,
         'Luis Fonsi': 2,
         'The Chainsmokers': 1})
```

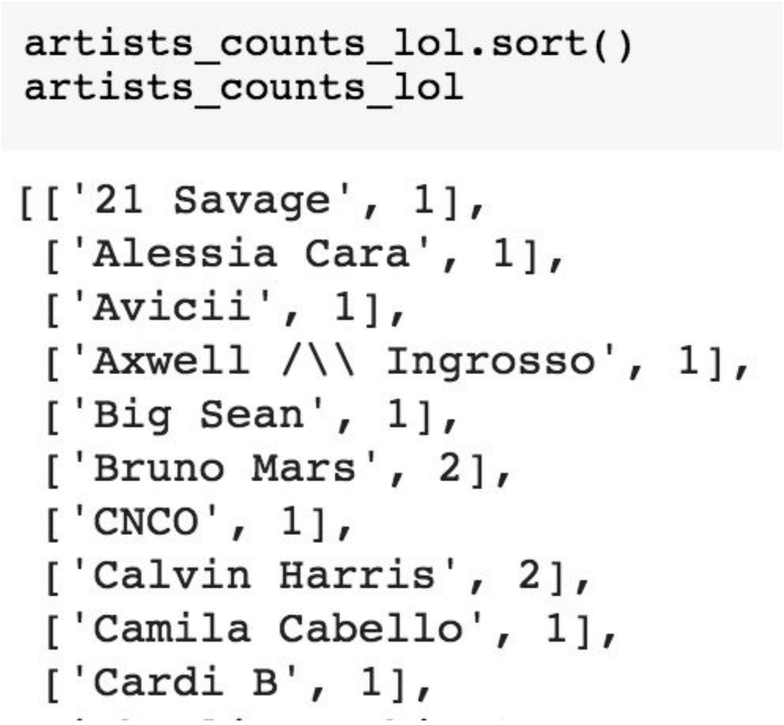


```
count_artists_list = [[key, item] for key, item in count_artists.items()]
count_artists_list
```

```
[['The Chainsmokers', 1],
 ['Ed Sheeran', 1],
 ['Kendrick Lamar', 1],
 ['Luis Fonsi', 2]]
```

Sorting a list of lists

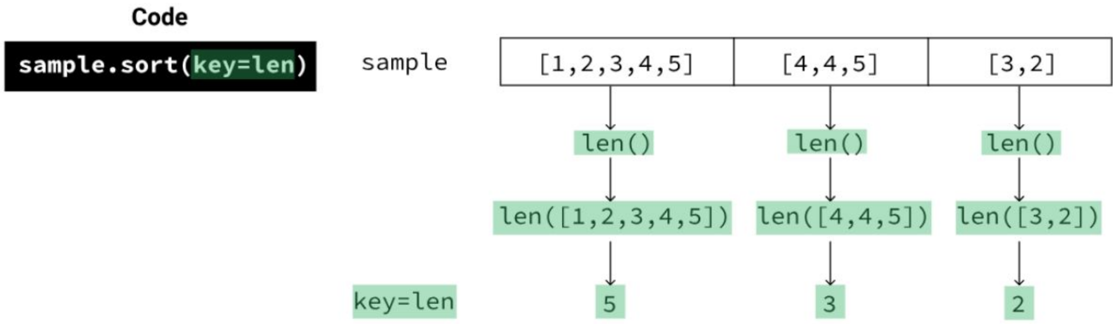
Alphabetical order



```
artists_counts_lol.sort()
artists_counts_lol
```

```
[['21 Savage', 1],
 ['Alessia Cara', 1],
 ['Avicii', 1],
 ['Axwell /\\ Ingrosso', 1],
 ['Big Sean', 1],
 ['Bruno Mars', 2],
 ['CNCO', 1],
 ['Calvin Harris', 2],
 ['Camila Cabello', 1],
 ['Cardi B', 1],
```

Customizing sort()

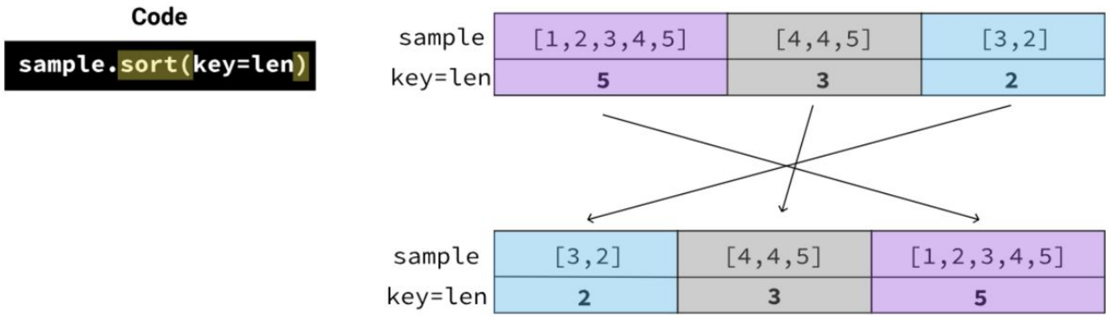


Code

```
sample.sort(key=len)
```

sample	[1,2,3,4,5]	[4,4,5]	[3,2]
	len()	len()	len()
	len([1,2,3,4,5])	len([4,4,5])	len([3,2])
key=len	5	3	2

After calculating the length for each value, each value will be sorted:



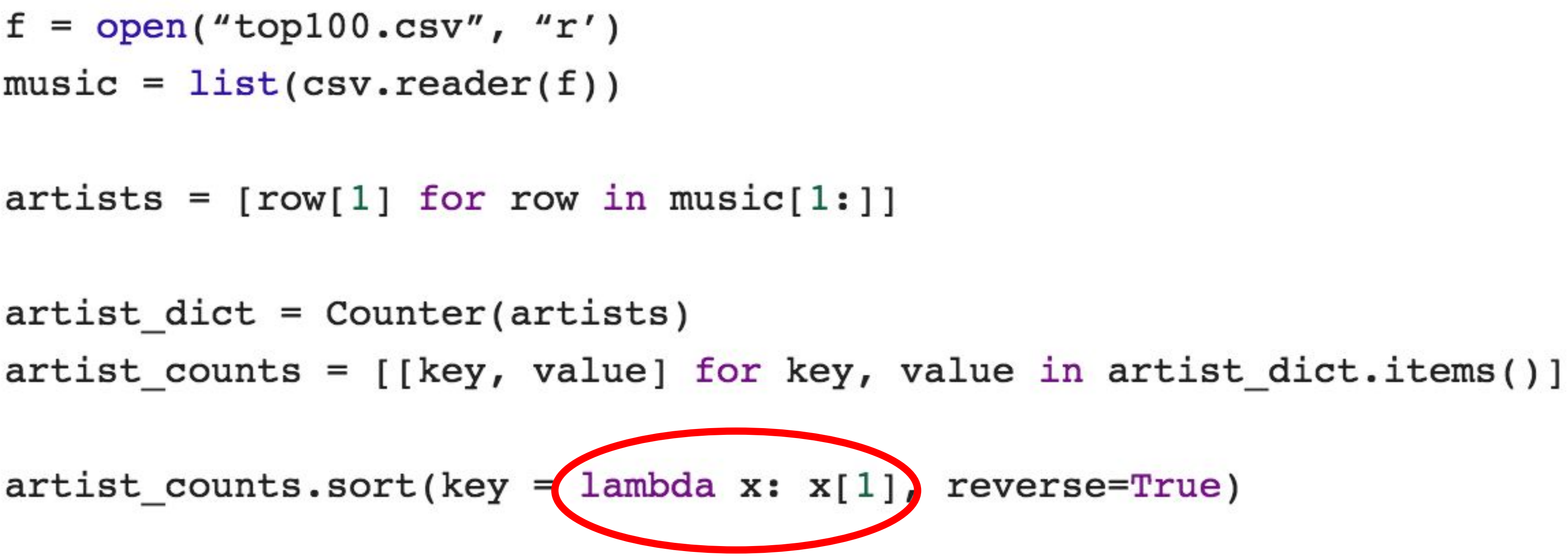
Code

```
sample.sort(key=len)
```

sample	[1,2,3,4,5]	[4,4,5]	[3,2]
key=len	5	3	2

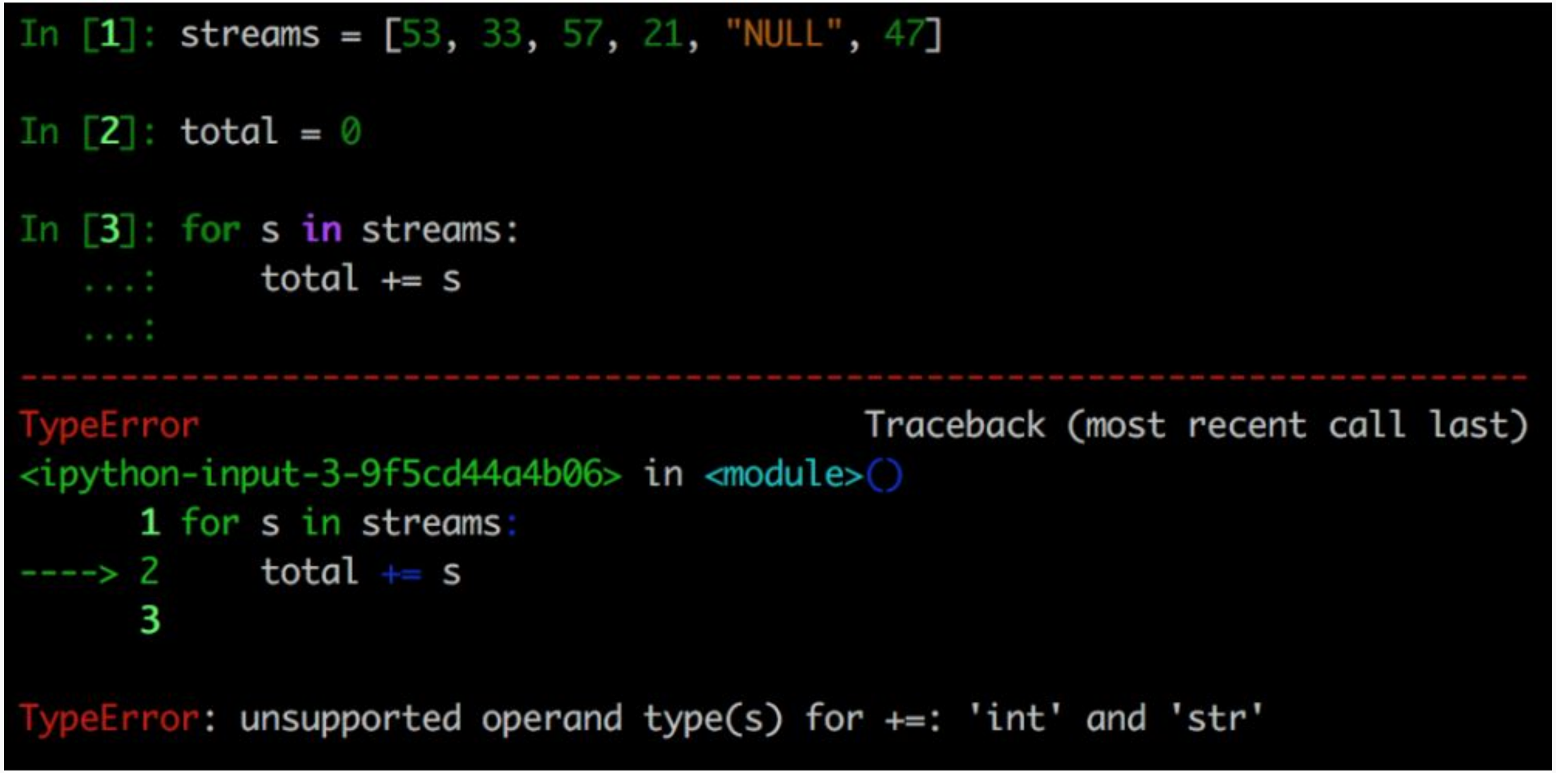
sample	[3,2]	[4,4,5]	[1,2,3,4,5]
key=len	2	3	5

Creating a anonymous function



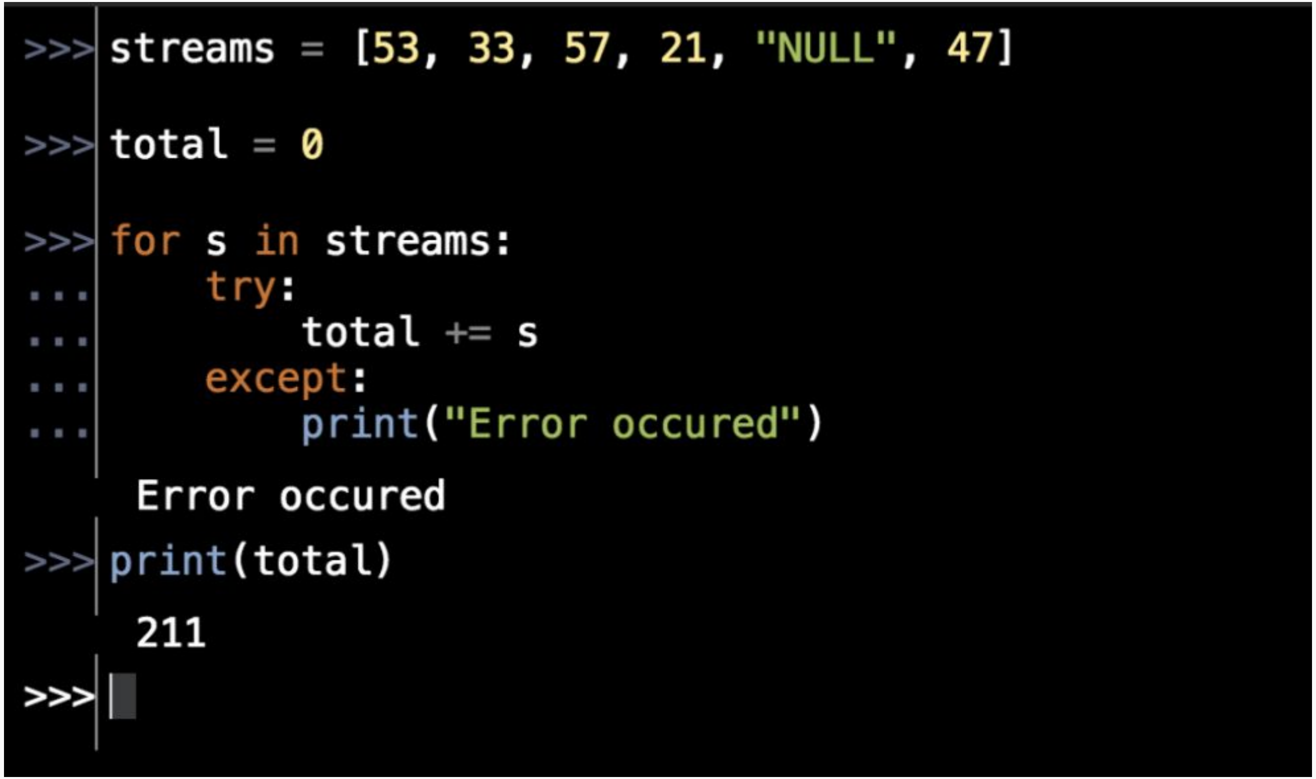
```
f = open(“top100.csv”, “r’)
music = list(csv.reader(f))

artists = [row[1] for row in music[1:]]

artist_dict = Counter(artists)
artist_counts = [[key, value] for key, value in artist_dict.items()]

artist_counts.sort(key = lambda x: x[1], reverse=True)
```

How to deal with errors



```
In [1]: streams = [53, 33, 57, 21, "NULL", 47]

In [2]: total = 0

In [3]: for s in streams:
   ...:     total += s
   ...:
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-3-9f5cd44a4b06> in <module>()
      1 for s in streams:
----> 2     total += s
      3

TypeError: unsupported operand type(s) for +=: 'int' and 'str'
```

How to deal with errors



```
>>> streams = [53, 33, 57, 21, "NULL", 47]

>>> total = 0

>>> for s in streams:
...     try:
...         total += s
...     except:
...         print("Error occured")
 Error occured
>>> print(total)
 211
>>> 
```

```
console.log("response.data", response.data);
this.setState({
  ajaxReturn: response.data
});
}.bind(this));
},

addContactAjaxCall: function() {
  console.log("Add new contact:", this.state.name, this.state.
  const contact = {
    name: this.state.name,
    email: this.state.email
  };
  ajaxHelpers.addContact(contact)
},

nameInput: function(e){
  this.setState({
    name: e.target.value
  })
},
emailInput: function(e){
  this.setState({
    email: e.target.value
  })
},
newNameInput: function(e){
  this.setState({
    newName: e.target.value
  })
},
newEmailInput: function(e){
```

```
import React from "react";

const FindContact = React.createClass({

  render: function(){
    return(
      <div className="find-contact">
        <label>Name:</label>
        <input
          type="text"
          className="find-name"
          placeholder="Name"
          onChange={this.props.getSearchName}
        />
      <button
        type="button"
        onClick={this.props.searchAjaxCall}>
        Get Contact
      </button>


    </div>
    )
  }
});

export default FindContact;
```