July 2019

Lesson #02 - Dictionaries & Functions

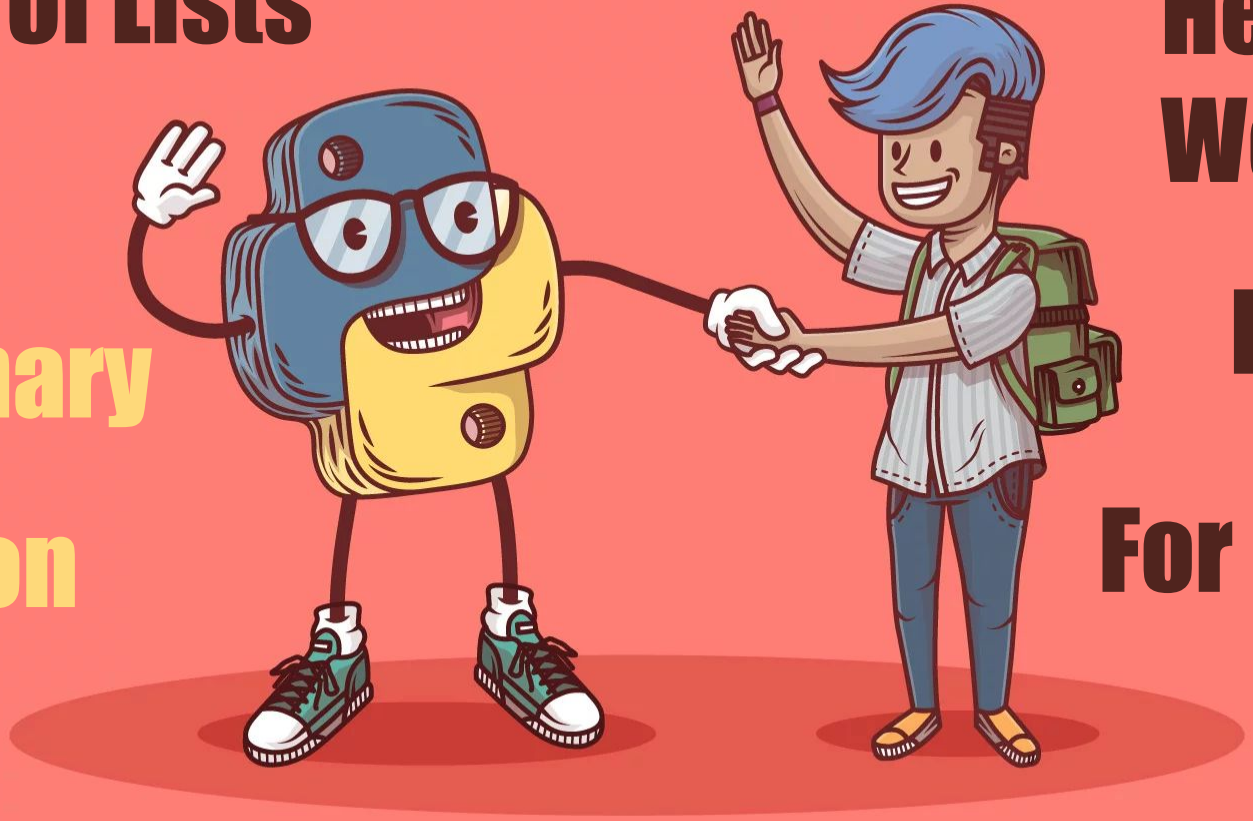
Real Python

Lists of Lists

Files

Dictionary

Function



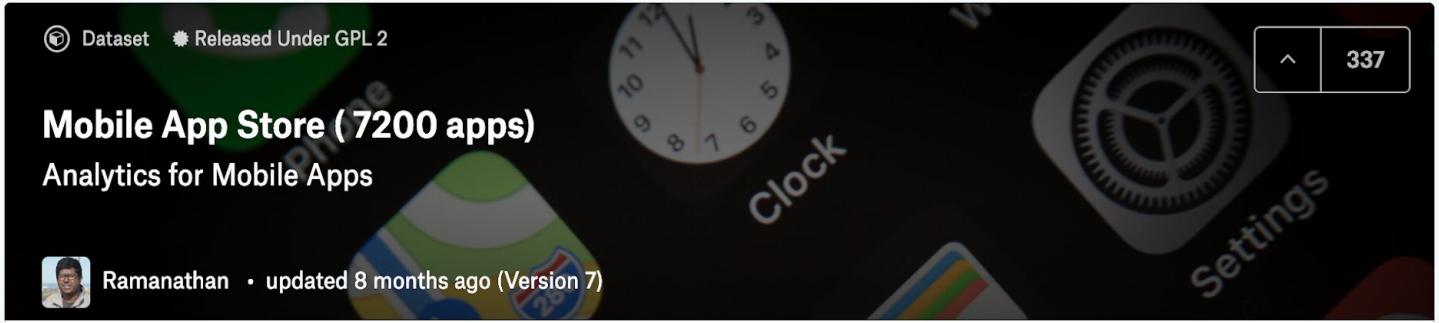
Hello World

Lists

For Loops

Conditional Statements

Real Python



Data (6 MB)

Data Sources

- AppleStore.csv 7198 x 17
- appleStore_descriptio... 7197 x 4

About this file

Apple Store

Columns

- # user_rating
- # user_rating_ver
- A ver
- A cont_rating
- A prime_genre
- # sup_devices.num
- # ipadSc_urls.num
- # lang.num

	id	track_name	size_bytes	price	user_rating_ver	ver	cont_rating	prime_genre
0	284882215	Facebook	USD	3.5	3.5	95.0	4+	Social Networking
1	389801252	Instagram	USD	4.5	4.0	10.23	12+	Photo & Video
2	529479190	Clash of Clans	USD	4.5	4.5	9.24.12	9+	Games
3	420009108	Temple Run	USD	4.5	4.0	1.6.2	9+	Games
4	284035177	Pandora - Music & Radio	USD	4.0	4.5	8.4.1	12+	Music

AppleStore.csv

The content rating of an app (also known as the maturity rating) represents the age required to use that app

What is the best data structure to store this kind of data?

Content rating	Number of apps
4+	4,433
9+	987
12+	1,155
17+	622

```
# Two lists
content_ratings = ['4+', '9+', '12+', '17+']
numbers = [4433, 987, 1155, 622]

# A list of lists
content_rating_numbers = [['4+', '9+', '12+', '17+'],
                          [4433, 987, 1155, 622]]
```

Content rating	Number of apps
4+	4,433
9+	987
12+	1,155
17+	622

```
numbers = [4433, 987, 1155, 622]
           0     1     2     3
```

Index numbers

```
numbers = [4433, 987, 1155, 622]
           0     1     2     3
           ↓     ↓     ↓     ↓
         '4+'  '9+'  '12+' '17+'
```

Dictionary

key:value

```
content_ratings = {'4+': 4433, '9+': 987, '12+': 1155, '17+': 622}
print(content_ratings['4+'])
print(content_ratings['12+'])
```

Output

```
4433
1155
```

```
d_1 = {'key_1': 'value_1',
       'key_2': 1,
       'key_3': 1.832,
       'key_4': False,
       'key_5': [1,2,3],
       'key_6': {'inner_key' : 10}
       }

print(d_1)
print(d_1['key_1'])
print(d_1['key_6'])
```

key:value
Value can be a generic type

Output

```
{'key_5': [1, 2, 3], 'key_3': 1.832, 'key_4':
False, 'key_6': {'inner_key': 10}, 'key_2': 1,
'key_1': 'value_1'}
value_1
{'inner_key': 10}
```

```
content_ratings = {'4+': 4433, '9+': 987, '12+': 1155, '17+': 622}
```


```
'12+' in content_ratings
```

Output

True

```
content_ratings = {'4+': 4433, '9+': 987, '12+': 1155, '17+': 622}

content_ratings['4+'] = 0
content_ratings['9+'] += 13
content_ratings['12+'] -= 1155
content_ratings['17+'] = '622'
print(content_ratings)
```

This is the same as:

```
content_ratings['9+'] = content_ratings['9+'] + 13
```

Output

```
{'4+': 0, '9+': 1000, '12+': 0, '17+': '622'}
```

```
content_ratings = {}
ratings = ['4+', '4+', '4+', '9+', '9+', '12+', '17+']

for c_rating in ratings:
    if c_rating in content_ratings:
        content_ratings[c_rating] += 1
    else:
        content_ratings[c_rating] = 1

content_ratings
```

Dictionaries can be used to count values

Output

```
{'12+': 1, '17+': 1, '4+': 3, '9+': 2}
```

```
content_ratings = {'4+': 4433, '9+': 987, '12+': 1155, '17+': 622}
total_number_of_apps = 7197
c_ratings_proportions = {}

for key in content_ratings:
    proportion = content_ratings[key] / total_number_of_apps
    c_ratings_proportions[key] = proportion

print(c_ratings_proportions)
print(content_ratings)
```

Creating a new empty dictionary

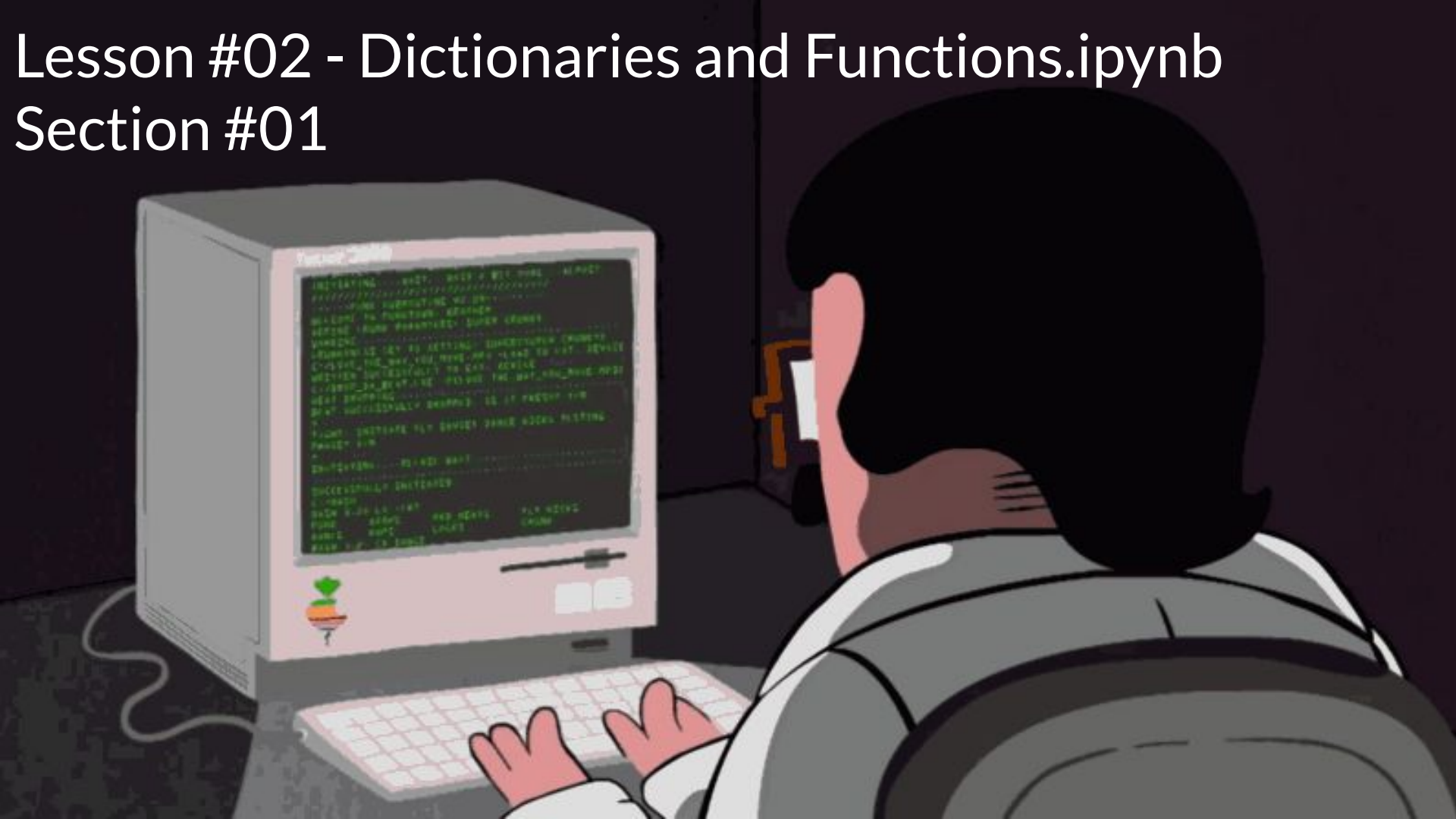
Populating the dictionary within the loop

Output

```
{'9+': 0.13714047519799916, '12+': 0.16048353480616923,
'4+': 0.6159510907322495, '17+': 0.08642489926358204}

{'9+': 987, '12+': 1155, '17+': 622, '4+': 4433}
```

Lesson #02 - Dictionaries and Functions.ipynb
Section #01



```
def square(a_number):
    squared_number = a_number * a_number
    return squared_number
```

header

body

the `return` statement

Indentation: four spaces to the right

```
def sum_and_difference(a, b):
    a_sum = a + b
    difference = a - b
    return a_sum, difference
```

We need to separate the variable names with a comma

```
sum_diff = sum_and_difference(15, 5)
sum_diff
```

Output

```
(20, 10)
```

```
a_list = [1, 'a', 10.5]
a_tuple = (1, 'a', 10.5)

print(a_tuple)
type(a_tuple)
```

Output

```
(1, 'a', 10.5)
tuple
```

```
a_list = [1, 'a', 10.5]
a_tuple = (1, 'a', 10.5)

print(a_tuple[0])
print(a_list[0])
print(a_tuple[-1])
print(a_list[-1])
```

Output

```
1
1
10.5
10.5
```

```
a_list = [1, 'a', 10.5]
a_list[0] = 99
a_list
```

Output

```
[99, 'a', 10.5]
```

```
a_tuple = (1, 'a', 10.5)
a_tuple[0] = 99
a_tuple
```

Output

```
TypeError: 'tuple' object does not support item assignment
```

Mutable	Immutable
Lists	*Tuples*
Dictionaries	*Integers*
	Floats
	Strings
	Booleans

```
def sum_and_difference(a, b):
    a_sum = a + b
    difference = a - b
    return a_sum, difference

a_sum, a_diff = sum_and_difference(15, 5)
print(a_sum)
print(a_diff)
```

```
Output
20
10
```

```
def print_constant():
    x = 3.14
    print(x)

print_constant()
```

Output

```
3.14
```

```
def print_constant():
    x = 3.14
    print(x)

j = print_constant()
print(j)
type(j)
```

Output

```
3.14
None
NoneType
```

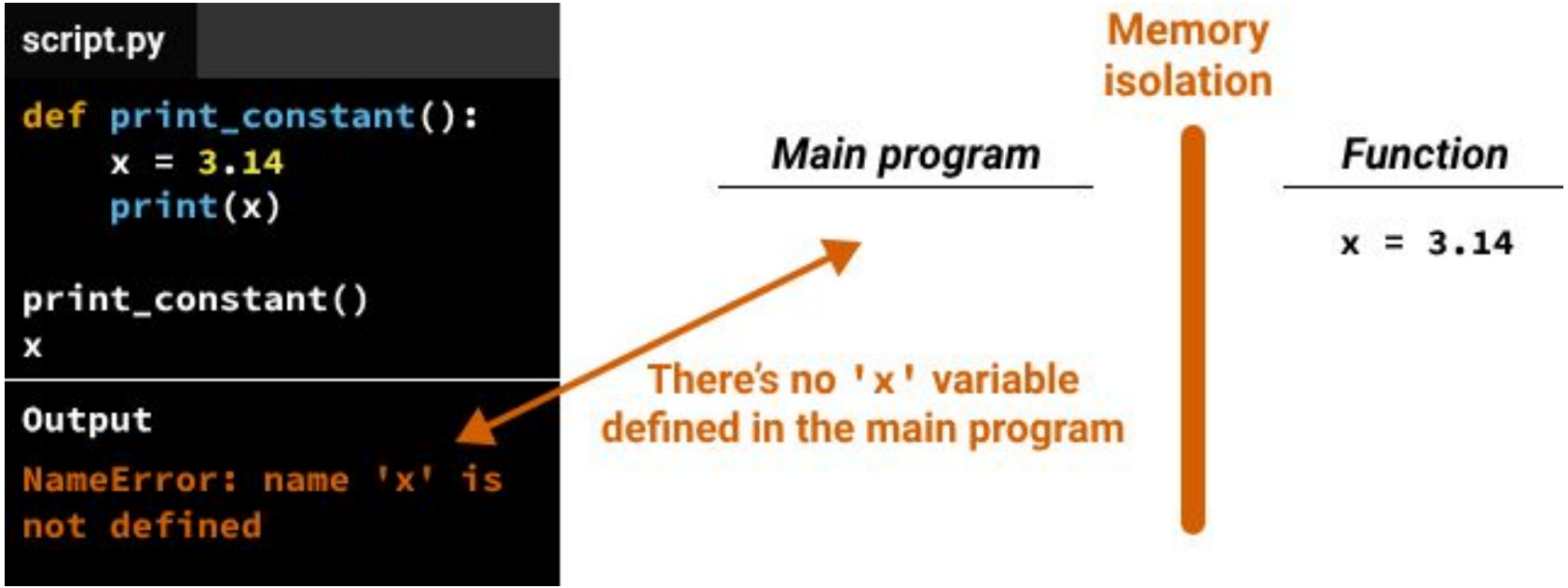
```
x = 10

def print_constant():
    x = 3.14
    print(x)

print_constant()
x
```

Output

```
3.14
10
```


script.py
def print_constant():
x = 3.14
print(x)
print_constant()
x
Output
NameError: name 'x' is not defined
Memory isolation
Main program
Function
x = 3.14
There's no 'x' variable defined in the main program

Lesson #02 - Dictionaries and Functions.ipynb
Section #02