

# Lecture Notes: Python Lecture for Yun's Family

Sunghee Yun

January 1, 2020

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Why learn computer programming? . . . . .	2
1.2	Why Python? . . . . .	2
1.3	Background . . . . .	2
1.4	Purpose of this lecture series . . . . .	3
<b>2</b>	<b>Python Basics</b>	<b>4</b>
2.1	Python Dictionary . . . . .	4
2.1.1	Dictionary Manipulation in Python . . . . .	4

# 1 Introduction

## 1.1 Why learn computer programming?

The following quote from a book by Briggs well explains why we should learn computer programming.

Programming fosters creatively, reasoning, and problem solving. The programmer gets the opportunity to create something from nothing, use logic to turn programming constructs into a form that a computer can run, and, when things don't work quite as well as expected, use problem solving to figure out what has gone wrong. Programming is a fun, sometimes challenging (and occasionally frustrating) activity, and the skills learned from it can be useful both in school and at work ... even if your career has nothing to do with computers.

And, if nothing else, programming is a great way to spend an afternoon when the weather outside is dreary.

– Jason R. Briggs, Python for Kids

## 1.2 Why Python?

Again here I quote some sentences from the same book instead of my own explanation.

Python is an easy-to-learn programming language that has some really useful features for a beginning programmer. The code is quite easy to read when compared to other programming languages, and it has an interactive shell into which you can enter your programs and see them run. In addition to its simple language structure and an interactive shell with which to experiment. Python has some features that greatly augment the learning process and allow you to put together simple animations for creating your own games. One is the `turtle` module, inspired by Turtle graphics (used by the Logo programming language back in the 1960s) and designed for educational use. Another is the `tkinter` module, and interface for the Tk GUI toolkit, which provides a simple way to create programs with slightly more advanced graphics and animation.

– Jason R. Briggs, Python for Kids

## 1.3 Background

This lecture series started by Ghayoung's suggestion that I teach Beth how to code. I'm not sure whether I suggested it or she said it herself, but we also agreed that Ghayoung would join us, too.

I had believed that it would be a bad idea to teach something to my family members, and that's why I'd never thought of this possibility. However, I've changed my mind after a few lectures that I gave them regarding Python coding. And here's why.

First and foremost, they had almost no problem understanding the concepts I explained to them. Also their attitude was very good. They tried their best to understand and follow my lead and they actually did great in absorbing the lecture contents and doing coding practice.

Secondly, I've realized that it is more valuable than anything else in the world for me to spend my precious time and energy to convey my knowledge and experience to the people I love and care most. I've done many talks, seminars, and lectures in my life, but none of them gave me the satisfaction that I feel now when seeing them improving and growing on coding.

In short, unlike my long-lasting conviction that it's never a good idea to teach something to my family members, it's turned out that it is indeed possible and it is extremely pleasing experience to do the lecture to my loved ones. Then I realized that I've already taught many things to them including all those math tricks and insight I tried to show Beth and numerous other concepts and stories I've told her, and all those (boring) conversation that I shared with Ghayoung about various topics such as math, science, engineering, optimization, and machine learning. The fact is that I have been doing sort of teaching already, and this is just one of those examples with difference that, this time, we have official schedules and formats.

So I'm planning to keep this lecture series as long as I can or until they can learn themselves without my help.

#### **1.4 Purpose of this lecture series**

XXX

## 2 Python Basics

### 2.1 Python Dictionary

#### 2.1.1 Dictionary Manipulation in Python

##### Overview

- A dictionary is a collection of key-value pairs.
- A dictionary is a set of key:value pairs.
- All keys in a dictionary must be unique.
- In a dictionary, a key and its value are separated by a colon.
- The key, value pairs are separated with commas.
- The key & value pairs are listed between curly brackets “{ }”
- We query the dictionary using square brackets “[ ]”

##### Dictionary Manipulation

- Dictionaries are useful whenever you have to items that you wish to link together, and for example storing results for quick lookup.
- Create an empty dictionary

```
months = {}
```

- Create a dictionary with some pairs (Note: Each key must be unique)

```
months = { 1 : "January",
           2 : "February",
           3 : "March",
           4 : "April",
           5 : "May",
           6 : "June",
           7 : "July",
           8 : "August",
           9 : "September",
          10 : "October",
          11 : "November",
          12 : "December" }
```

- months “1–12” are keys and “January-December” are the values
- Print all keys

```
print("The dictionary contains the following keys: ", list(months.keys()))
```

Output:

```
The dictionary contains the following keys:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

- Accessing

To get a value out of a dictionary, you must supply its key, you cannot provide the value and get the key

```
whichMonth = months[1]
print(whichMonth)
```

Output:

January

- To delete an element from a dictionary, use `dict.pop`

```
print(months.pop(5))
print(months.keys())
```

Output:

May  
[1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12]

- To add a new element to a dictionary, assign a value to a new key

```
months[5] = "MAY"
print(months.keys())
```

Output:

[1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 5]

- To update an element of a dictionary, assign a new value to its key

```
months[1] = "Jan"
print(months)
```

Output:

{1: 'Jan', 2: 'February', 3: 'March', 4: 'April', 5... }

- Sorting

```
sortedkeys = sorted(months.keys())
print(sortedkeys)
```

Output:

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

- Dictionaries and Loops

- Iterating over keys

```
for key in months:
    print(key, months[key])
```

Output:

```
1 Jan
2 February
3 March
4 April
5 MAY
6 June
7 July
8 August
9 September
10 October
11 November
12 December
```

– Iterating over (key, value) pairs

```
for key, value in months.items():
    print(key, value)

print "The entries in the dictionary are:"
for item in months.keys():
    print("months[ ", item, " ] = ", months[item])
```

- Combining List and Dictionary

Example of a list of dictionaries

```
customers = [
    {"uid":1,"name":"John"},
    {"uid":2,"name":"Smith"},
    {"uid":3,"name":"Andersson"},
]
print(customers)
```

Output:

```
[{'uid': 1, 'name': 'John'}, {'uid': 2, 'name': 'Smith'}, {'uid': 3, 'name': 'Andersson'}]
```

- Print the uid and name of each customer

```
for x in customer:
    print(x["uid"], x["name"])
```

Output:

```
1 John
2 Smith
3 Andersson
```

- Modify an entry

This will change the name of customer 2 from Smith to Charlie

```
customers[2]["name"]="charlie"  
print(customers)
```

Output:

```
[{'uid': 1, 'name': 'John'}, {'uid': 2, 'name': 'Smith'}, {'uid': 3, 'name': 'charlie'}]
```

- Add a new field to each entry

```
for x in customers:  
    x["password"]="123456" # any initial value  
  
print(customers)
```

Output:

```
[{'password': '123456', 'uid': 1, 'name': 'John'}, {'password': '123456', 'uid': 2,  
    'name': 'Smith'}, {'password': '123456', 'uid': 3, 'name': 'Andersson'}]
```

- Delete a field

```
del customers[1]  
print(customers)
```

Output:

```
[{'uid': 1, 'name': 'John'}, {'uid': 3, 'name': 'Andersson'}]
```

- Delete all fields

This will delete id field of each entry.

```
for x in customers:  
    del x["id"]
```

Output:

```
[{'name': 'John'}, {'name': 'Smith'}, {'name': 'Andersson'}]
```

## References

- [1] Jason R. Briggs. *Python for Kids*. No Starch Press, Incorporated.