



Florian Legendre

April 23, 2020

“Code is more often read than written.”

— Guido van Rossum

“It doesn’t matter how good your software is, because if the documentation is not good enough, people will not use it.”

— Daniele Procida

“Code tells you how; Comments tell you why.”

— Jeff Atwood (aka Coding Horror)

Contents

I	The Basics	3
1	Code Documentation	4
1.1	Comments	4
2	Data Types and Structures	5
2.1	Mutable/Not Mutable Data Types	5
2.2	Lists slices	5
2.3	Global Variables	6
3	Control Structures	7
3.1	Python's Switch/Case	7
4	Conventions	8
4.1	Indentation	8

Part I

The Basics

Chapter 1

Code Documentation

1.1 Comments

Only one way to comment in python: `#`

`""" """` has another use that I have yet to figure out.

Important note:

- C-x C-; can toggle commenting/decommenting several lines in Emacs! It seems to work for any programming language!

Chapter 2

Data Types and Structures

2.1 Mutable/Not Mutable Data Types

1. list = mutable
2. tuple = not mutable
3. string = not mutable

2.2 Lists slices

This verbatim below comes from the moving to python wikipage [4]:

Python indexes and slices for a six-element list.

Indexes enumerate the elements, slices enumerate the spaces between the elements.

```
Index from rear:  -6  -5  -4  -3  -2  -1
Index from front:   0   1   2   3   4   5
                  +---+---+---+---+---+---+
                  | a | b | c | d | e | f |
                  +---+---+---+---+---+---+
Slice from front:  :   1   2   3   4   5   :
Slice from rear:   :  -5  -4  -3  -2  -1   :
```

```
a=[0,1,2,3,4,5]    a[1:]==[1,2,3,4,5]
len(a)==6          a[:5]==[0,1,2,3,4]
a[0]==0            a[:-2]==[0,1,2,3]
a[5]==5            a[1:2]==[1]
a[-1]==5           a[1:-1]==[1,2,3,4]
a[-2]==4
```

```
b=a[:]
b==[0,1,2,3,4,5] (shallow copy of a)
```

2.3 Global Variables

In order to use global variables do as indicated here:

1. Declare variable
2. In local scope, statement `global <var>`
3. Use variable...

Chapter 3

Control Structures

3.1 Python's Switch/Case

There's no switch/case control structures in Python so to speak. But dictionaries can be used to fulfill that job as in the example below:

```
def hangman_states(case):  
    switcher = {  
        1: hangman_state1,  
        2: hangman_state2,  
        3: hangman_state3,  
        4: hangman_state4,  
        5: hangman_state5,  
        6: hangman_state6,  
        7: hangman_state7,  
        8: hangman_state8  
    }  
  
    return switcher.get(case, "State doesn't exist")()
```

Chapter 4

Conventions

4.1 Indentation

Indentation = holywar between tabs, four-spaces and weirdos. Not sure what I should use.

Bibliography

- [1] Python Software Foundation. *Pep 8 - Style Guide for Python Code*. Accessed: 2020-04-23. 2020. URL: <https://www.python.org/dev/peps/pep-0008/>.
- [2] Python Software Foundation. *Python Documentation Homepage*. Accessed: 2020-04-23. 2020. URL: <https://www.python.org/doc/>.
- [3] Python Software Foundation. *The Python Standard Library*. Accessed: 2020-04-23. 2020. URL: <https://docs.python.org/3/library/index.html>.
- [4] Python Wiki. *Moving to Python From Other Languages*. Accessed: 2020-04-23. 2020. URL: <https://wiki.python.org/moin/MovingToPythonFromOtherLanguages>.