Python Learning Notes

Florian Legendre

April 25, 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

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 Iterable Structures (Lists, Tuples, Sets...)

2.2.1 Lists Methods

Method	Description
append(any type)	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count(Any type)	Returns the number of elements with the specified value
extend(Any iterable)	Add the elements of a list (or any iterable), to the end of the current list
index(Any type)	Returns the index of the first element with the specified value
insert(pos, any type)	Adds an element at the specified position
pop(pos)	Removes the element at the specified position
remove(any type)	Removes the item with the specified value
reverse()	Reverses the order of the list
sort(reverse=True/False, key=...)	Sorts the list (ascending by default, descending if reverse is true)

2.2.2 Lists Slices

This verbatim below comes from the moving to python wikipage [7]:

```
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)
```

This quote below comes from a stackoverflow answer [5]:

> a[start:stop:step], step is 1 by default
> stop value represents the first value that is not in the selected slice.
> A blank represents an end of the lists (can be at the beginning or the end...)
>
> a[::-1] # all items in the array, reversed
> a[1::-1] # the first two items, reversed (we start from 1 and count backwards until we reached an end of the list)
> a[:-3:-1] # the last two items, reversed (we start from some end of the list and have to reach -3 counting in the opposite direction...)
> a[-3::-1] # everything except the last two items, reversed
>
> Python is kind to the programmer if there are fewer items than you ask for. For example, if you ask for a[:-2] and a only contains one element, you get an empty list instead of an error. Sometimes you would prefer the error, so you have to be aware that this may happen.

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. *Python Module Index*. Accessed: 2020-04-24. 2020. URL: https://docs.python.org/3/library/index.html.

[4] Python Software Foundation. *The Python Standard Library*. Accessed: 2020-04-23. 2020. URL: https://docs.python.org/3/library/index.html.

[5] Greg Hewgill. *Understanding slice notation*. URL: https://stackoverflow.com/a/509295/9924401.

[6] W3Schools. *Python Reference*. URL: https://www.w3schools.com/python/python_reference.asp.

[7] Python Wiki. *Moving to Python From Other Languages*. Accessed: 2020-04-23. 2020. URL: https://wiki.python.org/moin/MovingToPythonFromOtherLanguages.

[8] Python Wiki. *Useful Modules*. Accessed: 2020-04-23. 2020. URL: https://wiki.python.org/moin/UsefulModules.