Week 5 - Python Basics

Hongzheng Chen

Dec 13, 2019

1 Installation and Configuration

Unix-like systems basically pre-install Python distribution, which can be easily called by `python` or `python3`. If your Python version is 2.x, please update to 3.x, since Python organization will no longer support 2.x when it comes to 2020.

PyCharm is the widely used IDE for Python, but I am **not** recommend you to install, since VS Code has had initial support for Python and its extensions, which performs well and gets much better and better in recent versions. If you use Windows and Linux dual system (including those using WSL), what I recommend is to install Anaconda, an extremely powerful platform for Python development. The build-in conda and pip management tools make Python packages in order, and many packages (like pytorch for deep learning) can only be installed in this way. Detailed usage of Anaconda will be covered in the next few seminars, so you only need to download and install it.

If you use a singleton Unix-like system (e.g. MacOS, Linux whole system), everything is easy. No need for Anaconda. Once pip is installed, you get rid of tedious configuration steps for different packages.

2 Resources

There are massive Python tutorials on the Chinese Internet, and you can easily find them on Zhihu or Baidu, but most of them are not systematic enough. Thus, several recommended introductory materials are listed here:

- *Learning Python: Powerful Object-Oriented Programming* (Chinese: 《Python 学习手册》): The slide's examples are based on this book, and the electronic version can be found on net disk. This book is very long but extremely detailed. It is highly recommended to get through all its chapters and compare Python with C/C++. If you are familiar with C++, these contents can be learned quickly. But if you are still struggling with C/C++, learn *C++ Primer* first.
- *Structure and Interpretation of Computer Programs* (SICP, Python version): This book is firstly written in Scheme but latter translated into Python and becomes the introductory course of University of Berkeley, i.e. UCB 61A. Prof. Xinyu Feng in Nanjing University introduces it to China, his course can be found on NJU 22000130.
- *Python Crash Course: A Hands-On, Project-Based Introduction to Programming* (Chinese: 《Python 编程：从入门到实践》): It gives several application projects in the last

few chapters. This book can be found on net disk.

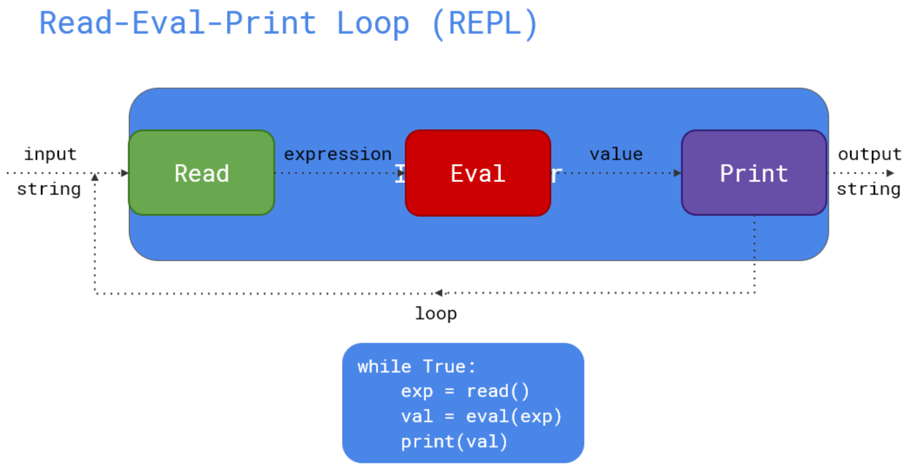
- Official Python 3.6 documentation: using it as a manual is the best way.
- 实验楼 - *Python3 简明教程*: You can use its interactive environment to learn Python, which is very down-to-earth. But maybe it is time-consuming if you have basic knowledge of programming languages.
- 廖雪峰 - *Python 教程*: concise but may miss some details.

You should read through PEP 8 (*Style Guide for Python code*) and PEP 20, and remember them in mind when you coding.

Moreover, Python will be used through the following seminars, your courses in sophomore and junior, and maybe your whole career life. Therefore, learn it carefully and get as deep as you can.

3 Project

Use Python to implement a Python interpreter. The flow of interpreter is shown in Fig. 1. Detailed descriptions will be posted later.



```
while True:
    exp = read()
    val = eval(exp)
    print(val)
```

图 1: A basic interpreter flow: Read-Eval-Print Loop (REPL)

Some reference materials:

- Byterun: A 500-line-code Python-based Python interpreter, which is based on Stack machine and may be very slow. Its description can be found in this Chinese blog.
- PyPy: A Python-implemented Just-in-time (JIT) Python compiler. It is much faster than the official CPython implementation in some specific cases.
- NJU SICP 20 - Interpreters: This slide gives the basic knowledge of interpreters.