Lab 1. Python basics

LING-581-Natural Language Processing1

Instructor: Hakyung Sung

August 27, 2025

*Acknowledgment: These course slides are based on materials from CS224N: NLP with Deep Learning @ Stanford University.

Table of contents

Introduction

Python Basics

- Python is a widely used, general-purpose programming language that is easy to learn, read, and write.

Python Basics

- Python is a widely used, general-purpose programming language that is easy to learn, read, and write.
- Popular among researchers and developers for its simplicity and readability

Python Basics

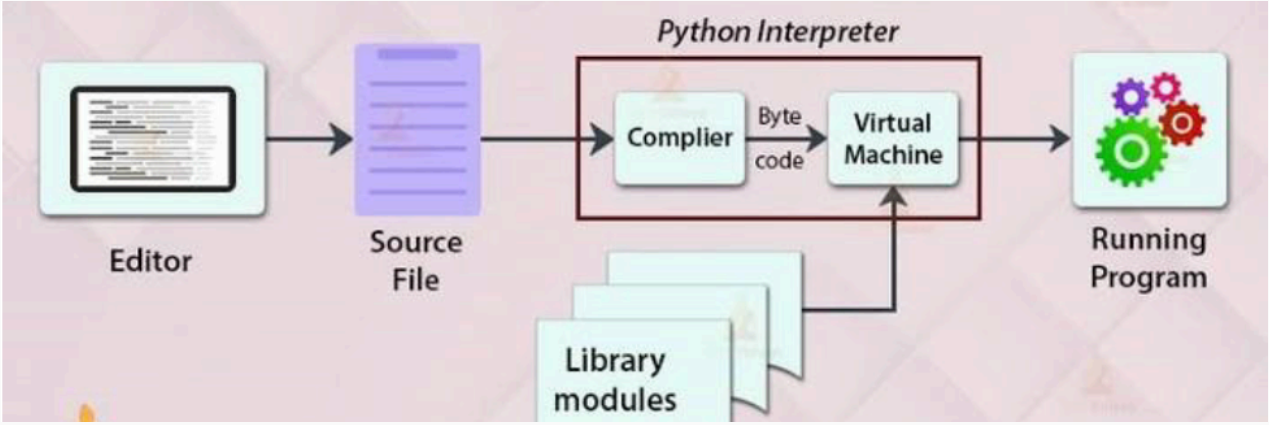
- Python is a widely used, general-purpose programming language that is easy to learn, read, and write.
- Popular among researchers and developers for its simplicity and readability
- Used by major deep learning frameworks (e.g., *PyTorch*)

Python Basics

- Python is a widely used, general-purpose programming language that is easy to learn, read, and write.
- Popular among researchers and developers for its simplicity and readability
- Used by major deep learning frameworks (e.g., *PyTorch*)
- Supported by an active open-source community and a vast ecosystem of libraries

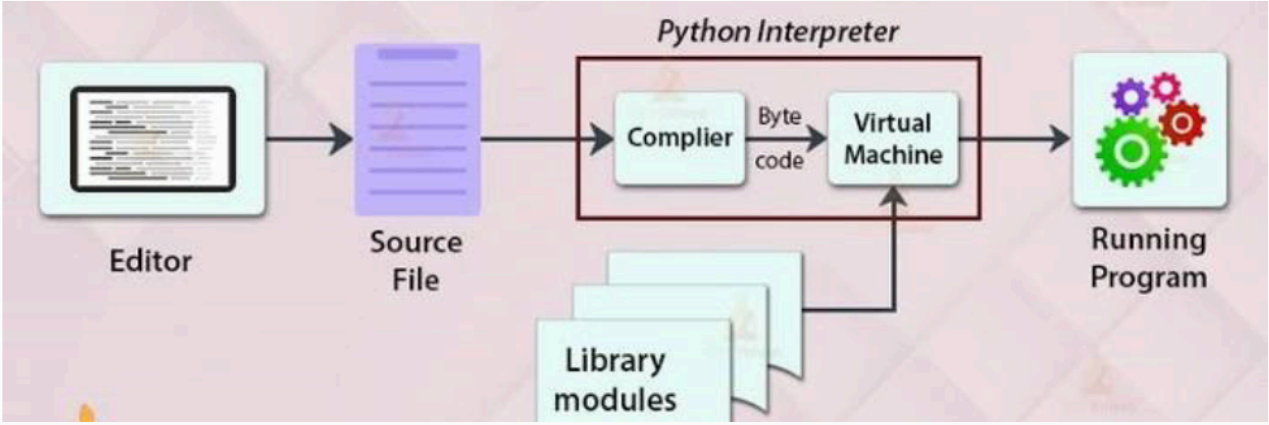
How Python Works

- Python is an **interpreted language** (cf. C, C++ which are compiled directly into machine code).



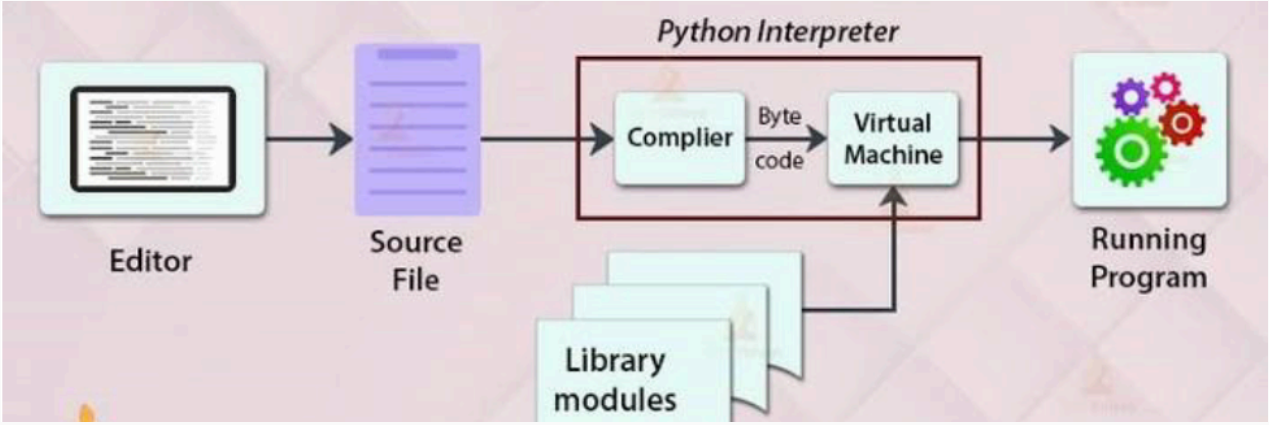
How Python Works

- Python is an **interpreted language** (cf. C, C++ which are compiled directly into machine code).
- Your code (.py) is first converted into bytecode (.pyc).



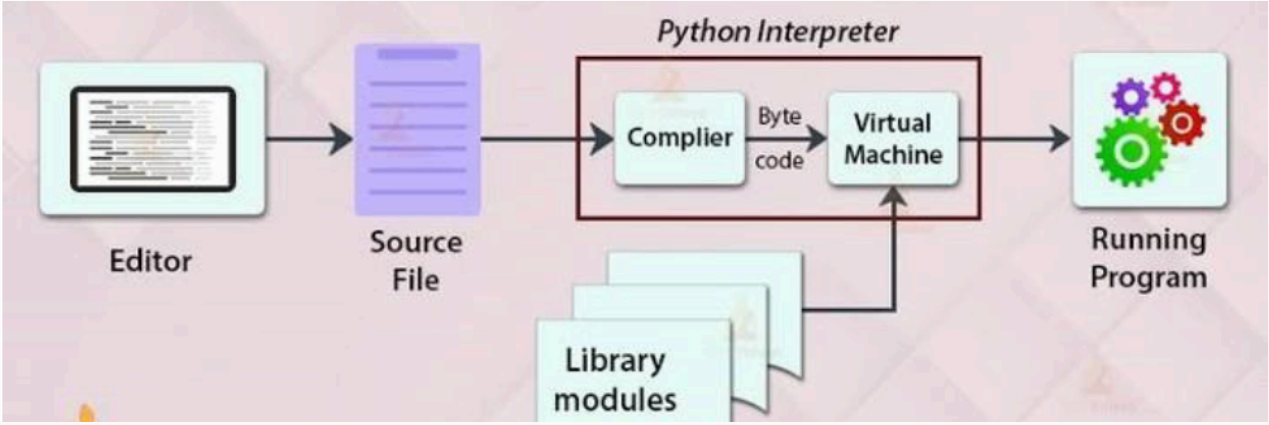
How Python Works

- Python is an **interpreted language** (cf. C, C++ which are compiled directly into machine code).
- Your code (.py) is first converted into bytecode (.pyc).
- The bytecode is executed by the Python Virtual Machine (VM).



How Python Works

- Python is an **interpreted language** (cf. C, C++ which are compiled directly into machine code).
- Your code (.py) is first converted into bytecode (.pyc).
- The bytecode is executed by the Python Virtual Machine (VM).
- Most implementations (e.g., CPython) are written in C and translate into machine code.



Python is Strongly Typed

- Python keeps track of the type of each variable.

Python is Strongly Typed

- Python keeps track of the type of each variable.
- It does not automatically convert between types unless explicitly told to.

Python is Strongly Typed

- Python keeps track of the type of each variable.
- It does not automatically convert between types unless explicitly told to.
- The interpreter respects types and raises errors for incompatible operations.

Tools to Run Python

IDEs and Notebooks

- IDE: Visual Studio Code (VSCode)

Tools to Run Python

IDEs and Notebooks

- IDE: Visual Studio Code (VSCode)
  - IDE = Integrated Development Environment

Tools to Run Python

IDEs and Notebooks

- IDE: Visual Studio Code (VSCode)
  - IDE = Integrated Development Environment
  - Lightweight, powerful editor

Tools to Run Python

IDEs and Notebooks

- IDE: Visual Studio Code (VSCode)
  - IDE = Integrated Development Environment
  - Lightweight, powerful editor
  - Integrated terminal, linting, debugging, version control

Tools to Run Python

IDEs and Notebooks

- IDE: Visual Studio Code (VSCode)
  - IDE = Integrated Development Environment
  - Lightweight, powerful editor
  - Integrated terminal, linting, debugging, version control
- Google Colab

Tools to Run Python

IDEs and Notebooks

- **IDE: Visual Studio Code (VSCode)**
  - IDE = Integrated Development Environment
  - Lightweight, powerful editor
  - Integrated terminal, linting, debugging, version control
- **Google Colab**
  - Browser-based, no installation

Tools to Run Python

IDEs and Notebooks

- **IDE: Visual Studio Code (VSCode)**
  - IDE = Integrated Development Environment
  - Lightweight, powerful editor
  - Integrated terminal, linting, debugging, version control
- **Google Colab**
  - Browser-based, no installation
  - Built on Jupyter notebooks

Tools to Run Python

IDEs and Notebooks

- **IDE: Visual Studio Code (VSCode)**
  - IDE = Integrated Development Environment
  - Lightweight, powerful editor
  - Integrated terminal, linting, debugging, version control
- **Google Colab**
  - Browser-based, no installation
  - Built on Jupyter notebooks
  - Pre-loaded libraries, GPU support, Google Drive integration

Tools to Run Python

IDEs and Notebooks

- **IDE: Visual Studio Code (VSCode)**
  - IDE = Integrated Development Environment
  - Lightweight, powerful editor
  - Integrated terminal, linting, debugging, version control
- **Google Colab**
  - Browser-based, no installation
  - Built on Jupyter notebooks
  - Pre-loaded libraries, GPU support, Google Drive integration
- For lab exercises, I'm planning to share Google Colab files.

Tools to Run Python

IDEs and Notebooks

- **IDE: Visual Studio Code (VSCode)**
  - IDE = Integrated Development Environment
  - Lightweight, powerful editor
  - Integrated terminal, linting, debugging, version control
- **Google Colab**
  - Browser-based, no installation
  - Built on Jupyter notebooks
  - Pre-loaded libraries, GPU support, Google Drive integration
- For lab exercises, I'm planning to share Google Colab files.
  - You will need a Google account to copy the files.

Tools to Run Python

IDEs and Notebooks

- **IDE: Visual Studio Code (VSCode)**
  - IDE = Integrated Development Environment
  - Lightweight, powerful editor
  - Integrated terminal, linting, debugging, version control
- **Google Colab**
  - Browser-based, no installation
  - Built on Jupyter notebooks
  - Pre-loaded libraries, GPU support, Google Drive integration
- For lab exercises, I'm planning to share Google Colab files.
  - You will need a Google account to copy the files.
  - Please submit your work as an `.ipynb` file so the grader can check both your code and its executed output.

Installation

Install Python 3

- Download installer: `https://www.python.org/downloads/`
- Windows: run `.exe`, check “Add Python to PATH”, click Install
- macOS: open `.pkg`, follow prompts
- Verify:
  - `python3 --version`
  - `python3 -v`, `python3 -vv`

Install Visual Studio Code

- Download: `https://code.visualstudio.com/`
- Windows: run `.exe`, follow defaults
- macOS: drag `VSCode.app` to /Applications
- Launch and install Python extension (Ctrl+Shift+X → Python)
- Verify in terminal: `python3 --version`

Running Code

Three Ways to Run Code

Example: Python as a Calculator

- `>>> 1 + 5 * 2 - 3 → 8`
- `>>> (2 + 3) * 4 → 20`
- `>>> 1.0 / 3.0 → 0.333...`

Execution Methods

- Interactive Terminal
  - `$ python3`
- Script File
  - `python3 calculator.py`
- IDE or Notebook
  - VSCode Python file
  - Colab notebook cell

Environment Management

Why Manage Environments?

Problem	Solution
Multiple Python versions	Create isolated environments
Many dependencies	Manage within environments
Conflicting packages	Keep project-specific envs
Version conflicts	Isolate environments per project
Hard to reproduce	Share via virtual env configs

Solution 1: venv

- Built-in tool for creating virtual environments.
- Create: `python -m venv myenv`
- Activate: `source myenv/bin/activate`
- Deactivate: `deactivate`
- Includes: interpreter, libraries, scripts, isolated from global install.

Solution 2: Anaconda / Miniconda

- Manages both Python and non-Python dependencies
- Create env: `conda create -n myenv python=3.10`
- Activate: `conda activate myenv`
- Deactivate: `conda deactivate`
- Export env: `conda env export > environment.yml`

Installing Packages

- Using conda:
  - `conda install -n myenv package_name`
  - Specify version: `=1.2.3`
- Using pip in conda env:
  - `pip install package_name`
  - Tip: prefer conda, use pip only when necessary

Tutorials

Tutorials

For the remainder of the class, students will work on the tutorials (either individually or with a peer next to you).

Please go through the four tutorials step-by-step using the provided Colab code.

- Values, variables, functions, methods
- Strings, lists, conditional statements, loops
- Tuples, dictionaries, functions, classes, files

All the necessary information is in the tutorials. At the end of class, please submit your `.ipynb` file with your name (e.g., *Lab1_HakyungSung.ipynb*).