Introduction to Python

DA2303 – Python Programming

What is Python?

- Python is an object-oriented scripting language that was released publicly in 1991.
- It was developed by Guido van Rossum of the National Research Institute for Mathematics and Computer Science in Amsterdam.
- Python has rapidly become one of the world's most popular programming languages.
- It's now particulrly popular for educational and scientific computing.
- It recently surpassed the programming language R as the most popular data-science programming language.

Why Python?

- It’s open source, free and widely available with a massive open-source community.
- It’s easier to learn than languages like C, C++, C# and Java, enabling novices and professional developers to get up to speed quickly.
- It’s easier to read than many other programming languages.
- It’s widely used in education.

Why Python? (cont.)

- It enhances developer productivity with extensive standard libraries and thousands of third-party open-source libraries, so programmers can write code faster and perform complex tasks with minimal code.

- There are massive numbers of free open-source Python applications.

- It supports popular programming paradigms – procedural, functional and object-oriented.

- It’s popular in artificial intelligence, which is enjoying explosive growth, in part because its special relationship with data science.

Installing Python

- There are multiple ways to install Python to your computer.
  - Download and Install Python directly from official Python website: https://www.python.org/downloads/
  - Download and Install Anaconda Individual Edition which includes Python: https://www.anaconda.com/products/individual
  - Download and Install Miniconda which includes Python: https://docs.conda.io/en/latest/miniconda.html

We'll use this installation for this module

Installing Miniconda

- Download the installer respective to your operating systems and follow the instruction given in the documentations.
  - Windows: https://docs.conda.io/projects/conda/en/latest/user-guide/install/windows.html
  - Mac OS: https://docs.conda.io/projects/conda/en/latest/user-guide/install/macos.html
  - Linux: https://docs.conda.io/projects/conda/en/latest/user-guide/install/linux.html

Anaconda Individual Edition

- It includes the latest versions of Python.

- Anaconda also includes other software packages and libraries commonly used in Python programming and data science.

- It uses conda packages where you could search Anaconda cloud-based repository to find and install additional data science and machine learning packages.

    - Traditionally Python developers used pip.

Anaconda Individual Edition (cont.)

- Conda also makes it easy to manage data environments that can be maintained and run separately without interference from each other.
    - Traditionally Python developer using virtualenv or pipenv.
- You can learn more by going to Anaconda website: https://www.anaconda.com/products/individual

Miniconda

- It is a small, bootstrap version of Anaconda that includes only conda, Python, the packages they depend on, and a small number of other useful packages.
- You can learn more by going to Miniconda website : https://docs.conda.io/en/latest/miniconda.html

Why Anaconda?

- Like the convenience of having Python and over 1,500 scientific packages automatically installed at once.
- Have the time and disk space---a few minutes and 3 GB.
- Do not want to individually install each of the packages you want to use.
- Wish to use a curated and vetted set of packages.

Why Miniconda?

- Do not mind installing each of the packages you want to use individually.
- Do not have time or disk space to install over 1,500 packages at once.
- Want fast access to Python and the conda commands and you wish to sort out the other programs later.

Why this module will use Miniconda?

- Get familiar using conda by installing required packages manually.
- Still taking advantage of conda to install packages and manage environments.

Python Packages (Libraries)

- We focus on using existing libraries to help you avoid “reinventing the wheel,” thus leveraging your program-development efforts.

- Often, rather than developing lots of original code—a costly and time-consuming process—you can simply create an object of a pre-existing library class, which takes only a single Python statement.

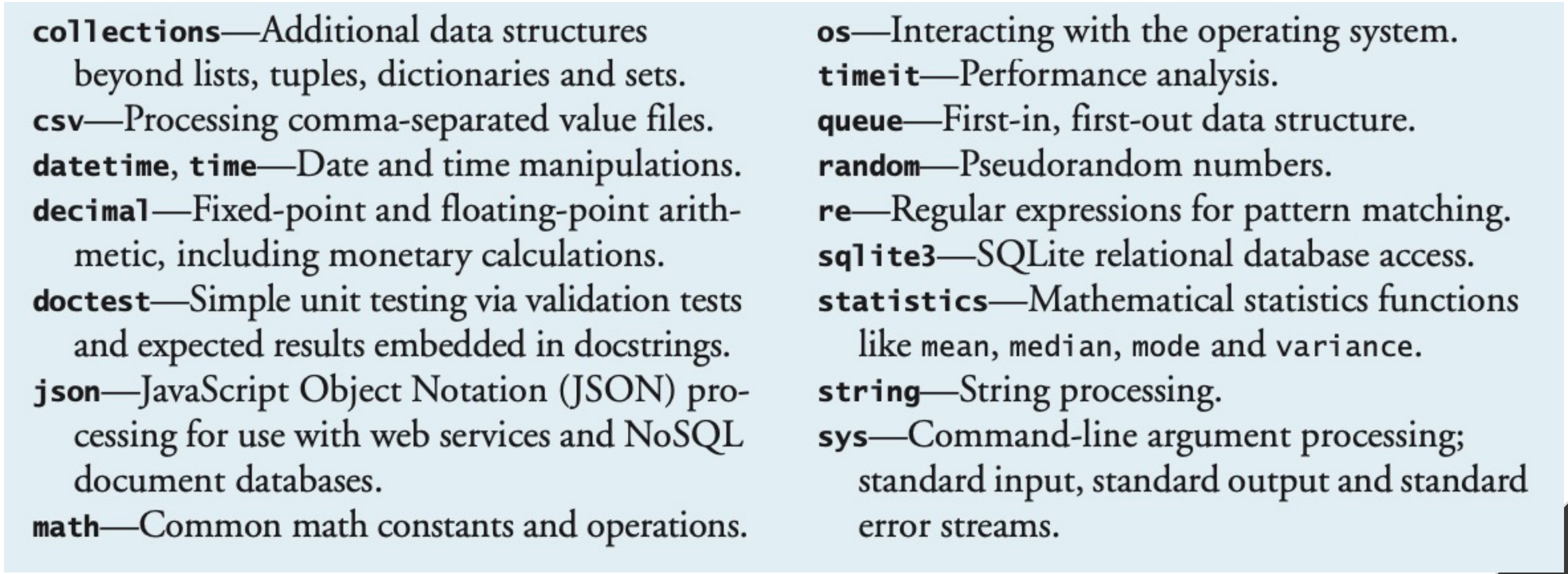
- So, libraries will help you perform significant tasks with modest amounts of code.

- You’ll use a broad range of Python standard libraries, data-science libraries and other third-party libraries.

Python Standard Library

- The Python Standard Library provides rich capabilities for
  - text/binary data processing,
  - mathematics,
  - functional-style programming,
  - file/directory access,
  - data persistence,
  - data compression/archiving,
  - cryptography,
  - operating-system services,
  - concurrent programming,
  - JSON/XML/other Internet data formats,
  - multimedia,
  - GUI,
  - and more.

Example: Python Standard Library



`collections`—Additional data structures beyond lists, tuples, dictionaries and sets.

`csv`—Processing comma-separated value files.

`datetime`, `time`—Date and time manipulations.

`decimal`—Fixed-point and floating-point arithmetic, including monetary calculations.

`doctest`—Simple unit testing via validation tests and expected results embedded in docstrings.

`json`—JavaScript Object Notation (JSON) processing for use with web services and NoSQL document databases.

`math`—Common math constants and operations.

`os`—Interacting with the operating system.

`timeit`—Performance analysis.

`queue`—First-in, first-out data structure.

`random`—Pseudorandom numbers.

`re`—Regular expressions for pattern matching.

`sqlite3`—SQLite relational database access.

`statistics`—Mathematical statistics functions like `mean`, `median`, `mode` and `variance`.

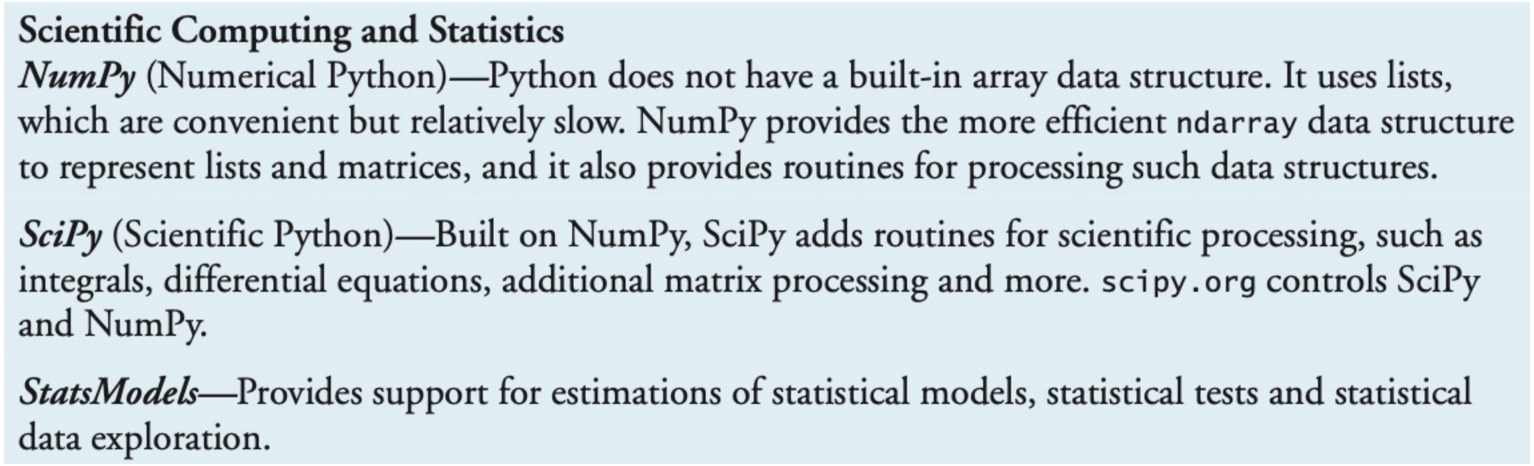
`string`—String processing.

`sys`—Command-line argument processing; standard input, standard output and standard error streams.

Data-Science Libraries

- Python has an enormous and rapidly growing community of open-source developers in many fields.
- One of the biggest reasons for Python’s popularity is the extraordinary range of open-source libraries developed by the open-source community.
- In this module, we will explore Pandas and Numpy.

Examples: Data-Science Libraries



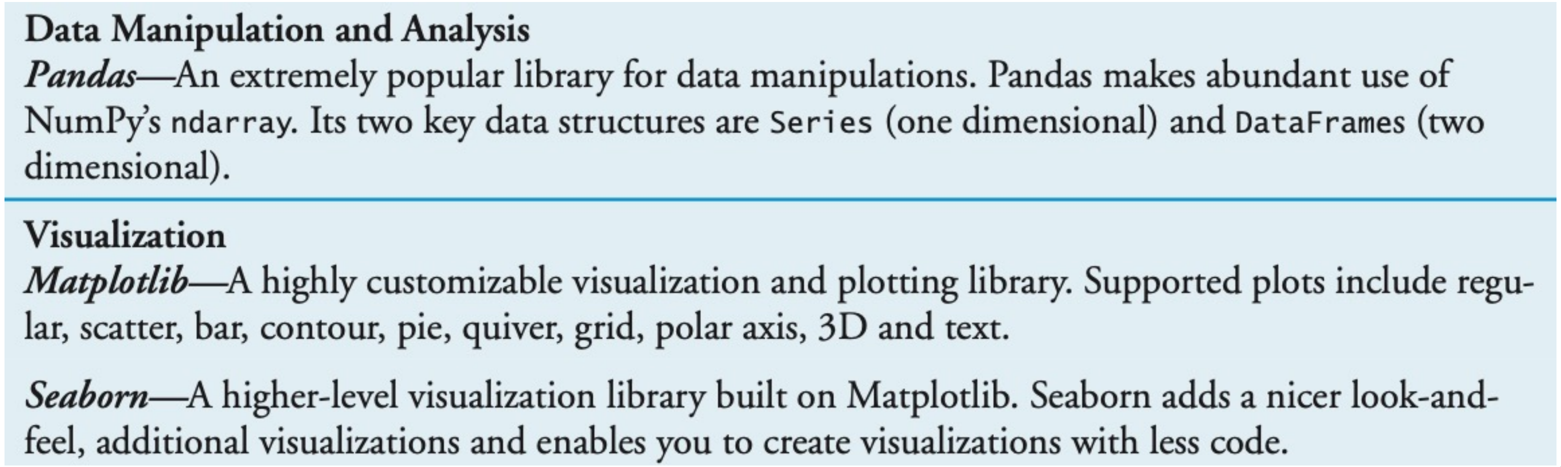
Scientific Computing and Statistics

NumPy (Numerical Python)—Python does not have a built-in array data structure. It uses lists, which are convenient but relatively slow. NumPy provides the more efficient `ndarray` data structure to represent lists and matrices, and it also provides routines for processing such data structures.

SciPy (Scientific Python)—Built on NumPy, SciPy adds routines for scientific processing, such as integrals, differential equations, additional matrix processing and more. `scipy.org` controls SciPy and NumPy.

StatsModels—Provides support for estimations of statistical models, statistical tests and statistical data exploration.

Examples: Data-Science Libraries (cont.)



Data Manipulation and Analysis

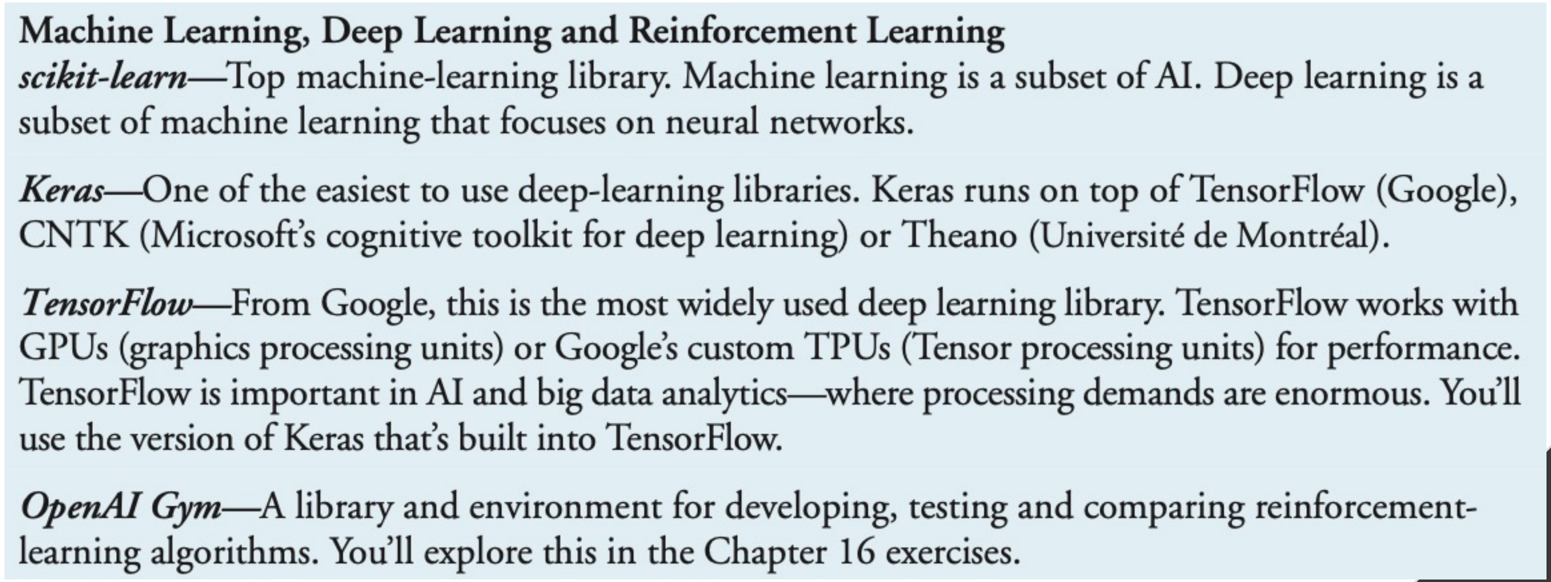
Pandas—An extremely popular library for data manipulations. Pandas makes abundant use of NumPy's `ndarray`. Its two key data structures are `Series` (one dimensional) and `DataFrames` (two dimensional).

Visualization

Matplotlib—A highly customizable visualization and plotting library. Supported plots include regular, scatter, bar, contour, pie, quiver, grid, polar axis, 3D and text.

Seaborn—A higher-level visualization library built on Matplotlib. Seaborn adds a nicer look-and-feel, additional visualizations and enables you to create visualizations with less code.

Examples: Data-Science Libraries (cont.)



Machine Learning, Deep Learning and Reinforcement Learning

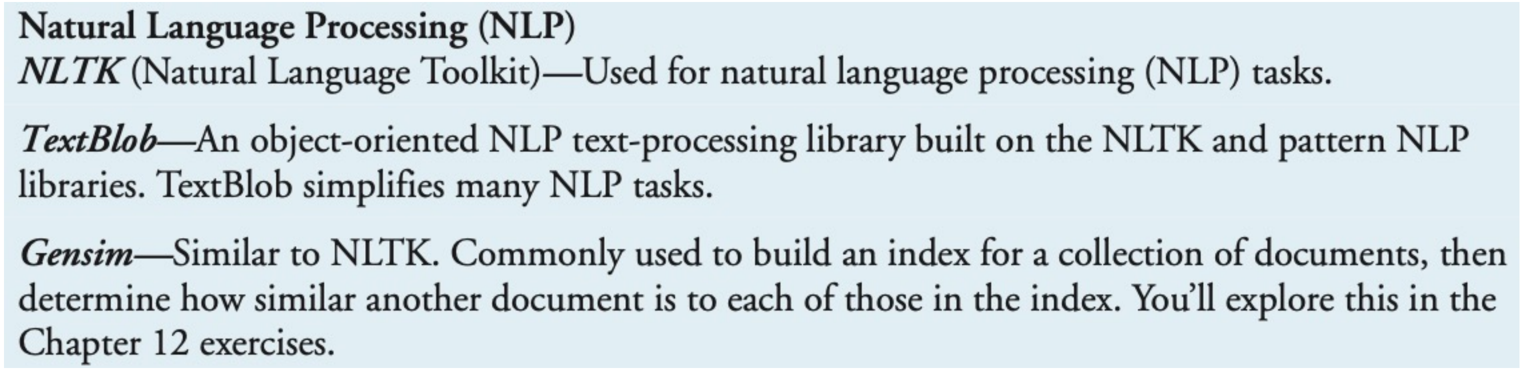
scikit-learn—Top machine-learning library. Machine learning is a subset of AI. Deep learning is a subset of machine learning that focuses on neural networks.

Keras—One of the easiest to use deep-learning libraries. Keras runs on top of TensorFlow (Google), CNTK (Microsoft's cognitive toolkit for deep learning) or Theano (Université de Montréal).

TensorFlow—From Google, this is the most widely used deep learning library. TensorFlow works with GPUs (graphics processing units) or Google's custom TPUs (Tensor processing units) for performance. TensorFlow is important in AI and big data analytics—where processing demands are enormous. You'll use the version of Keras that's built into TensorFlow.

OpenAI Gym—A library and environment for developing, testing and comparing reinforcement-learning algorithms. You'll explore this in the Chapter 16 exercises.

Examples: Data-Science Libraries (cont.)



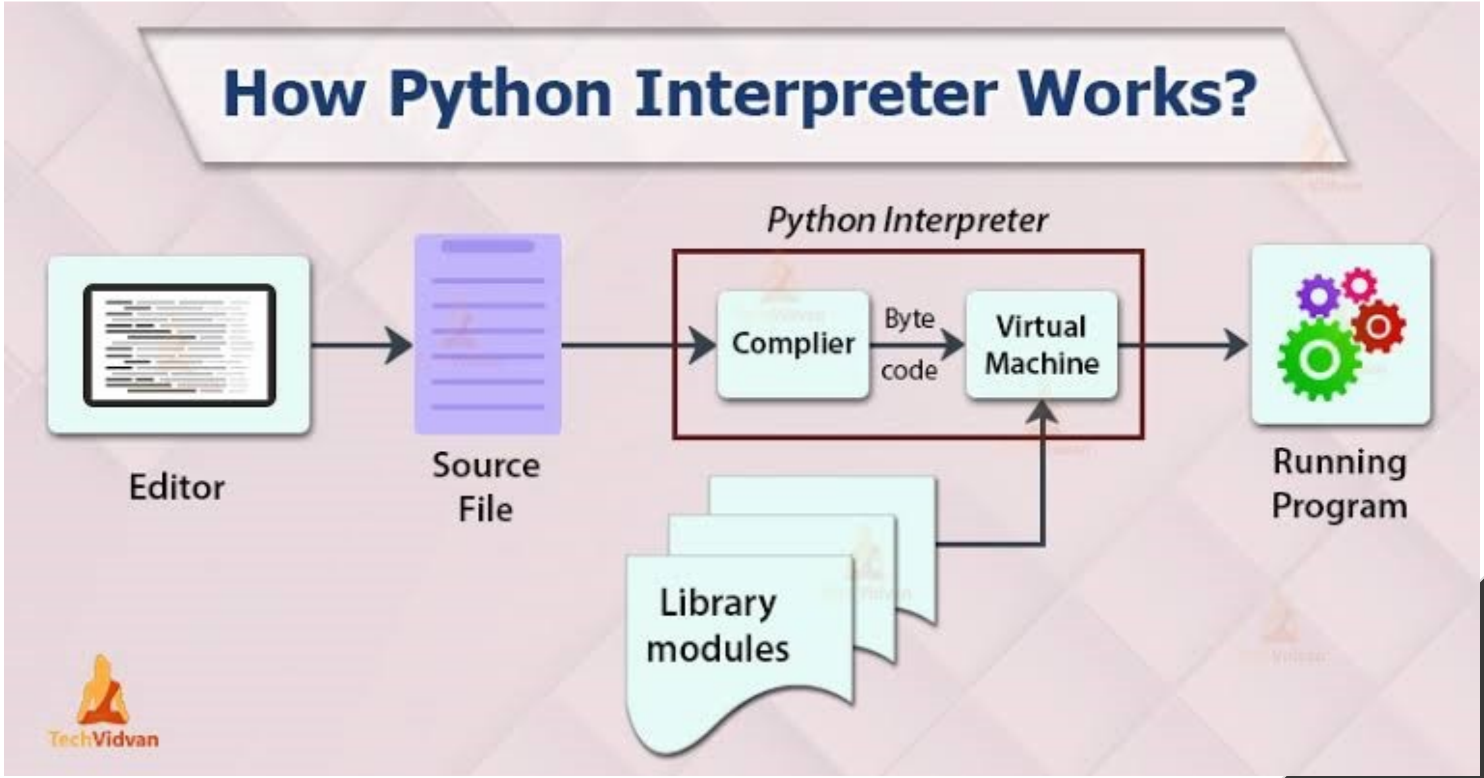
Natural Language Processing (NLP)

NLTK (Natural Language Toolkit)—Used for natural language processing (NLP) tasks.

TextBlob—An object-oriented NLP text-processing library built on the NLTK and pattern NLP libraries. TextBlob simplifies many NLP tasks.

Gensim—Similar to NLTK. Commonly used to build an index for a collection of documents, then determine how similar another document is to each of those in the index. You'll explore this in the Chapter 12 exercises.

How Python works.



Writing Python codes

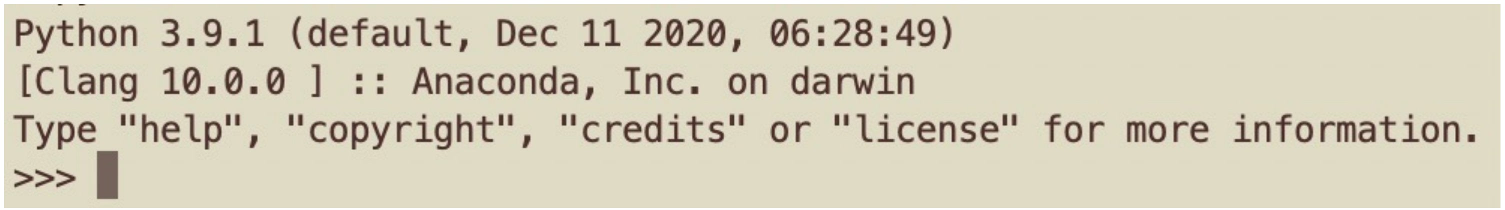
- There are two ways to write Python codes.
  - In interactive mode: you’ll enter small bits of Python code called snippets and immediately see their results.
  - In script mode: you’ll execute code loaded from a file that has the .py extension (short for Python). Such files are called scripts or programs, and they’re generally longer than the code snippets you’ll do in interactive mode.

Interactive Mode

- First, open a command-line window on your system:
    - On macOS, open a Terminal from Applications folder’s Utilities subfolder.
    - On Windows, open Anaconda Prompt from the start menu.
    - On Linux, open your system’s Terminal or shell (this varies by Linux distribution).

Interactive Mode (cont.)

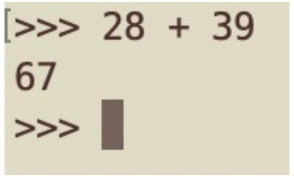
- In the command-line window, type python3, then press Enter (or Return).
- You'll see text like the following, this varies by platform and by python version:



```
Python 3.9.1 (default, Dec 11 2020, 06:28:49)
[Clang 10.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

Interactive Mode (cont.)

- The text “>>>” is a prompt, indicating that python is waiting for your input.
- You can begin enter snippets. Example, you can evaluate expressions:



```
>>> 28 + 39
67
>>> 
```

- After you type 28 + 39 and press Enter, python reads the snippet, evaluates it and prints its result after that.
- Then python displays the text “>>>” to show that it’s waiting for the next snippet.

Exiting Interactive Mode

- To leave interactive mode, you can:
    - Type the exit() command at the current prompt and press Enter to exit immediately.
    - Press the key sequence <Ctrl> + d.

Script Mode

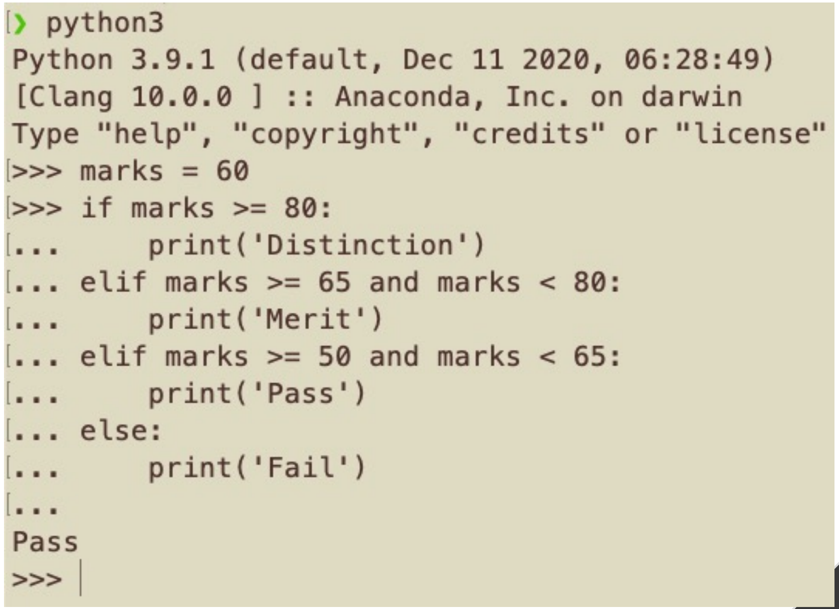
- Consider the following code.

```python
marks = 60

if marks >= 80:
    print('Distinction')
elif marks >= 65 and marks < 80:
    print('Merit')
elif marks >= 50 and marks < 65:
    print('Pass')
else:
    print('Fail')
```

Script Mode (cont.)

- It is still possible to run this in interactive mode.
- The problem writing this in interactive mode is that what if you want to test with a different marks.
- You have to write the code again.
- It would be much faster to write the code in a text editor and save it as a Python file (.py) file.
    - Then execute the code in script mode.



```
❯ python3
Python 3.9.1 (default, Dec 11 2020, 06:28:49)
[Clang 10.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license"
>>> marks = 60
>>> if marks >= 80:
...     print('Distinction')
... elif marks >= 65 and marks < 80:
...     print('Merit')
... elif marks >= 50 and marks < 65:
...     print('Pass')
... else:
...     print('Fail')
...
Pass
>>>
```

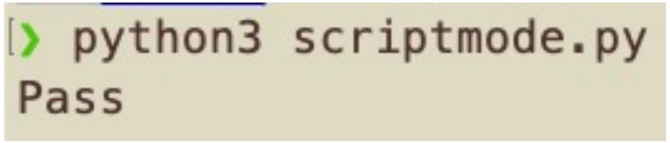
Script Mode (cont.)

- Lets say this code is save as a file as scriptmode.py

```python
marks = 60

if marks >= 80:
    print('Distinction')
elif marks >= 65 and marks < 80:
    print('Merit')
elif marks >= 50 and marks < 65:
    print('Pass')
else:
    print('Fail')
```

- Then we could execute this file in script mode.



```
> python3 scriptmode.py
Pass
```

Visual Studio Code (VS Code)

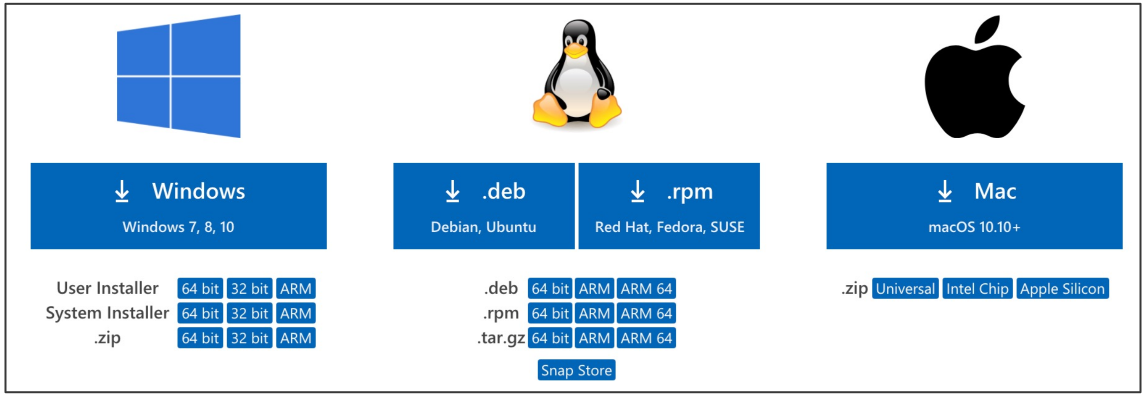
- Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux.

- It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

- We are going to use this code editor to write our Python scripts.

Installing VS Code

- Download the respective installer for your operating systems.
    - https://code.visualstudio.com/#alt-downloads



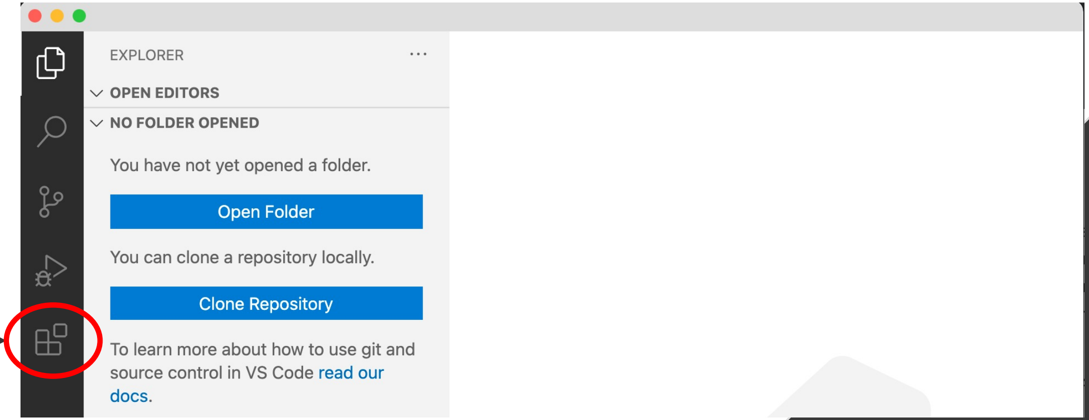
Installing VS Code (cont.)

- The following are the instructions on how to install VS Code in your respective operating systems.
  - Windows: https://code.visualstudio.com/docs/setup/windows
  - Mac: https://code.visualstudio.com/docs/setup/mac
  - Linux: https://code.visualstudio.com/docs/setup/linux

Python in VS Code

- There is no built-in support for Python development in VS Code.
- To make VS Code an excellent Python editor, extensions for VS Code needs to be install into VS Code.

Click this icon to start installing extensions for your VS Code



You have not yet opened a folder.

You can clone a repository locally.

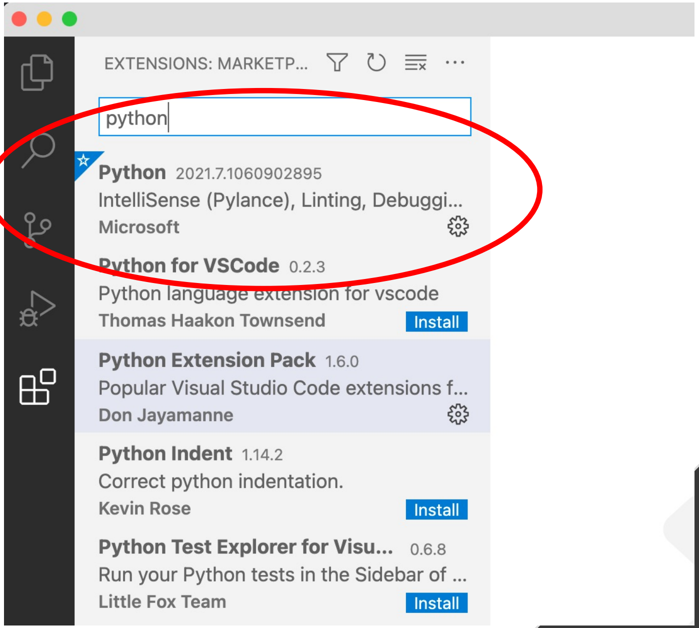
To learn more about how to use git and source control in VS Code read our docs.

Go to this link to read whats included in the extension
https://code.visualstudio.com/docs/languages/python

Python in VS Code (cont.)

- Search "python" extensions by typing in the text field available.

Select this Python extension provided by Microsoft



Python 2021.7.1060902895
IntelliSense (Pylance), Linting, Debuggi...
Microsoft

Python for VSCode 0.2.3
Python language extension for vscode
Thomas Haakon Townsend
Install

Python Extension Pack 1.6.0
Popular Visual Studio Code extensions f...
Don Jayamanne

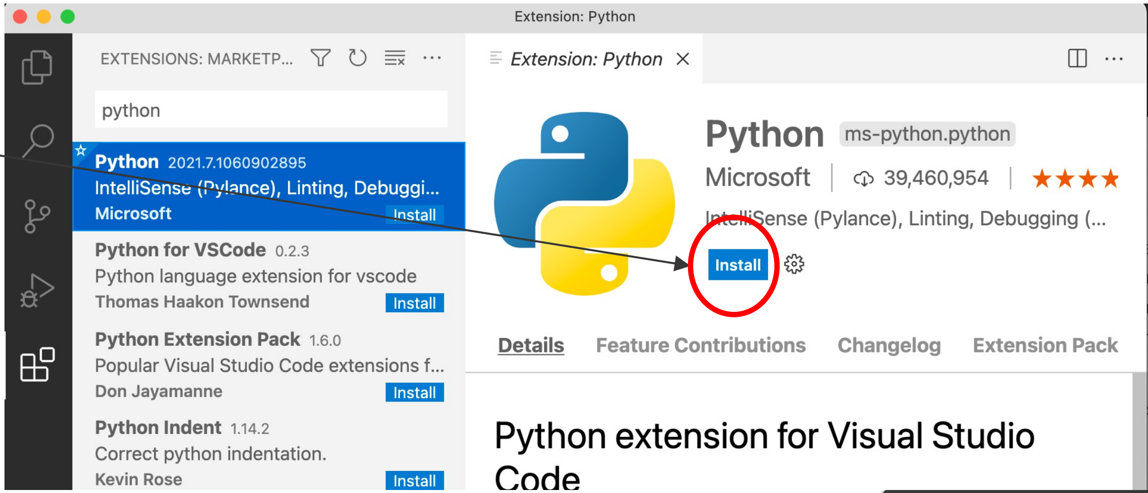
Python Indent 1.14.2
Correct python indentation.
Kevin Rose
Install

Python Test Explorer for Visu... 0.6.8
Run your Python tests in the Sidebar of ...
Little Fox Team
Install

Python in VS Code (cont.)

- Click the install button to install the extension to VS Code.

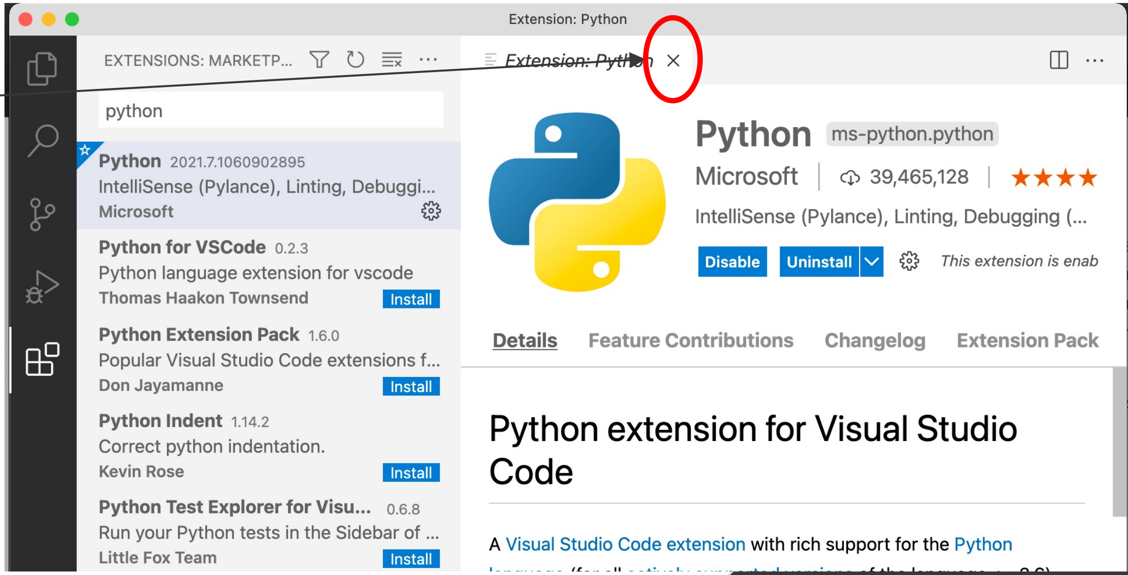
Click the install button



Python in VS Code (cont.)

- After it is done installing. Close the Extension Tab and you can now start developing Python scripts.

Close this Tab



python

Python 2021.7.1060902895
IntelliSense (Pylance), Linting, Debuggi...
Microsoft

Python for VSCode 0.2.3
Python language extension for vscode
Thomas Haakon Townsend

Python Extension Pack 1.6.0
Popular Visual Studio Code extensions f...
Don Jayamanne

Python Indent 1.14.2
Correct python indentation.
Kevin Rose

Python Test Explorer for Visu... 0.6.8
Run your Python tests in the Sidebar of ...
Little Fox Team

Python ms-python.python
Microsoft | 39,465,128 | ★★★★
IntelliSense (Pylance), Linting, Debugging (...
This extension is enab

Python extension for Visual Studio Code

A Visual Studio Code extension with rich support for the Python

Get Familiar with VS Code

- To get familiar with VS CodeIt is recommend viewing the videos:
    - https://code.visualstudio.com/docs/getstarted/introvideos
- To increase productivity in writing codes using VS Code, it is recommended to read the documentations below:
    - https://code.visualstudio.com/docs/getstarted/tips-and-tricks
- Also get familiar with the keyboard shortcuts:
    - https://code.visualstudio.com/shortcuts/keyboard-shortcuts-windows.pdf

Get Familiar with VS Code (cont.)

- Also get familiar with the keyboard shortcuts:
    - Windows: https://code.visualstudio.com/shortcuts/keyboard-shortcuts-windows.pdf
    - Mac: https://code.visualstudio.com/shortcuts/keyboard-shortcuts-macos.pdf
    - Linux: https://code.visualstudio.com/shortcuts/keyboard-shortcuts-linux.pdf

Table of contents

01

Hardware and software

Thanks

Do you have any question?

jailani.rahman@pb.edu.bn
(+673) 223 4466 ext 241
Unit 6.01, Ong Sum Ping Campus,
Politeknik Brunei.