CSI 2103 - Data Structures: Python / Jupyter Guide

Instructor: Seong Jae Hwang (seongjae@yonsei.ac.kr)

Installing Python3 and Jupyter Notebook/Lab

Our goal is to eventually run this `assignment1.ipynb` on your own Jupyter Notebook or Jupyter Lab. Focus on this section if you need to install Python and/or Jupyter Notebook/Lab based on `assignment1.pdf`.

What you will install:

- Python3 (any stable latest version is fine)
- Jupyter Notebook (open-source IDE for running .ipynb files) or
- Jupyter Lab (More "IDE-like" version of Jupyter Notebook. I recommend this one.)

To enter the command lines (if needed)

For macOS and Linux: use Terminal

For Windows: use Command Prompt

1. Traditional Way (pip):

- Pro: command-line based, light-weight, supports virtual environment
- Cons: may need to install additional packages, needs terminal or command prompt

1. Install Python3 manually from the official website: https://www.python.org/
   - When downloading, any latest stable version (3.x) for your OS should be fine.
   - Windows: during installation steps, look for "Modify Setup" and check "pip" (python package management system)
   - Check your python3 installation: `python3 --version`
   - Check your pip installation: `pip --version`
2. Install Jupyter Notebook or Jupyter Lab (recommended) from the official website: https://jupyter.org/install
   - Install Jupyter Lab: `pip install jupyterlab`
   - Install Jupyter Notebook: `pip install notebook`
3. Run
   - Run Jupyter Lab: `jupyter lab` (or `jupyter-lab`)
   - Run Jupyter Notebook: `jupyter notebook` (or `jupyter-notebook`)

2. Another Way (conda):

- Pro: GUI-based, easy to install on Windows, comes with some packages, supports virtual environment (starts in base)
- Cons: GUI-based, heavy, difficult to install some packages (won't be the case for us)

1. Install conda (python package manager): https://www.anaconda.com/products/individual
    - Official installation guide: https://docs.anaconda.com/anaconda/install/
    - conda is always `Python 3`
2. Open Anaconda-Navigator
3. In "Home", install JupyterLab or Notebook if needed
4. Launch JupyterLab or Notebook

Useful Resources and Links:

These can also be found on our course learnus under "Useful Resources and Links"

Here, you will find links to various useful resources.

Python and Setup (Official Links)

Note: official links are not always the easiest to follow

- Python: Official website for installation and etc.
- Python Doc: Official documentation for Python. Useful for Python related syntax help.
- Anaconda: Official website for installation
- Jupyter: Official website
- Jupyter Installation: Official installation guide
- Jupyter Lab Documentation: Official Jupyter Lab guide
- Anaconda Navigator User Guide: Official guide for Anaconda Navigator
- Anaconda Navigator Package Management: How to manage packages in Anaconda Navigator
- Anaconda Package Management: How to manage packages in Anaconda (command-line)

Other Online Resources and Tutorials

Note: some of these are by third-party public individuals

- Python (Anaconda) on Windows: install Python through Conda on Windows
- Jupyter Notebook Windows Installation: install Jupyter Notebook on Windows
- Jupyter Notebook/Lab Tutorial: useful tutorial for Jupyter Notebook/Lab
- Stanford Python Numpy Tutorial: useful tutorial for Python Numpy library (which we use a lot)
- Stanford Jupyter Notebook Tutorial File (**Recommend**): an extensive Jupyter Notebook tutorial .ipynb file. Download by **right-click and save link as** in pdf or learnus wiki, and open in Notebook/Lab
- w3school Python Tutorial: thorough Python tutorial where you can try directly in your browser

---

Getting Started: Running Jupyter

By now, you should be able to run Jupyter Notebook/Lab and load `assignment1.ipynb` . Load in Jupyter Lab/Notebook.

The best thing about Jupyter Notebook/Lab is how we run and analyze our code, one `cell` at a time. Each `.ipynb` file is a series of cells (i.e., a "block" of code) which can be rendered into the text you are reading now, or be treated as Python code.

1. Select the cell by clicking anywhere in the cell. If the cell does not have Python codes (i.e., `code` cell), you are looking at a `Markdown` cell (like this one you are reading) which you need to double click the cell.
2. Run the cell
    - `Run` Tab at the top menu -> `Run Selected Cell`
    - $\blacktriangleright$ button at the top of this notebook tab
    - Short-cut: `shift+enter`

This cell is a `Markdown` cell for documentation: indicated with `Markdown` in the drop-down menu at the top of this tab. This also needs to be run to render the text.

The cell below is a `Code` cell for running code: indicated with `Code` .

In [17]:
```
# Sample code (try it with your code)
print('Hello, world.')
2 + 3
result = 3 + 5
```

```
Hello, world.
```

Important: Notebook keeps track of all the variables run thus far as global variables. This is very useful since we can essentially treat the entire notebook as a series of code which we are familiar with, except we can run groups of lines and see the corresponding results.

Below, if we have assigned a variable, it can be used again:

In [18]:
```
print(result)
```

```
8
```

If you want to start fresh from scatch (delete all variables, etc.), click $\blacktriangleright\blacktriangleright$ for `restart the kernel, then re-run thew whole notebook`

In [19]:
```
# Only the last evaluated line will be automatically printed

# will not appear
1+1

# will appear
2+2
```

Out[19]:
```
4
```

See all variables and functions

- Enable debugger (top-right corner of this tab, left of `Python 3` . You can also do `Run` -> `Restart kernel and debug`
- Open the debugger tab in the right sidebar (`View -> Show Right Sidebar`)

Short-cuts

Here are a few commonly used short-cuts. (`ESC - m` means press `ESC` and press `m` , no need to hold `ESC`):

- `Esc – m` : switch cell to Markdown cell
- `Esc – y` : switch cell to code cell
- `Esc – a` : insert cell above
- `Esc – b` : insert cell below
- `Esc – d – d` : delete cell
- `Shift + Enter` : run the cell and move to the next cell
- `arrow keys` : move between cells
- `Command/Control – s` : save

To run multiple cells

See the "Run" tab at the top

- `Run All Above Selected Cell` : If you want to run all the cells just before a specific cell you want to start working on.
- `Run Selected Cell and All Below` : If you want to run all the cells from the one you are working on.