

Welcome to
COS1000
Computational Thinking for Scientists
-- Introduction to lab



Lab Overview:

Learning outcome:

- (a) develop fundamental skills in Python programming
- (b) learn about the basics of data manipulation, visualisation and interpretation
- (c) learn to implement computational solutions to solve simple to real life complex problems (deterministic vs probabilistic approaches).

Week	Topics to be covered
Week 1	Installation of Anaconda (Jupyter Notebook)
Week 2	Lab 1: Introduction; Getting started with Jupyter Notebook
Week 3	Lab 2: Programming fundamentals
Week 4	Lab 3: Program flow
Week 5	Lab 4: Functions and modules
Week 6	Lab 5: Data visualization
Recess week	
Week 7	Lab 6: Logistics map
Week 8	Lab 7: ODEs & predator-prey modelling
Week 9	Lab 8: Disease spread modelling
Week 10	Lab 9: Random processes

Lab Structure:

- Live hands-on session @ S13-M-08
- Once weekly
- Timings:
 - Mondays: 9 am to 12 pm
 - Tuesdays: 9 am to 12 pm
 - Wednesdays: 9 am to 12 pm; 12 to 3 pm
 - Fridays: 12 to 3 pm
- Attendance (1 mark) and participation (1 mark)
 - Submit your work before leaving the lab; make sure you let your lab facilitator check your work before submission.

Quiz:

- Five quizzes @ LumiNUS Quiz
- MCQ/MRQ
- About two weeks to complete
- Basic concepts and coding practices

Assessment:

- Online quizzes: 5%
- Lab attendance and participation: **20%**
- Two live term tests: 40%
- Two projects: 35%

Personnel (Lab Facilitators):

Day	Name	Contact
Mon 9 – 12	Chan Si Min e0004798@u.nus.edu	Teoh Jing Yang e0148566@u.nus.edu
Tue 9 – 12	Cheng Jue Fei cjuefei@nus.edu.sg	Wu Yizhou e0540623@u.nus.edu
Wed 9 – 12	Wang Hao e0708223@u.nus.edu	Yao Yi e0564217@u.nus.edu
Wed 12 – 3	Meng Shuyang e0572880@u.nus.edu	Shen Ruizhe e0554228@u.nus.edu
Fri 12 – 3	Kaushik Jayaraman e0503722@u.nus.edu	Ms Tonishka Singh e0550656@u.nus.edu

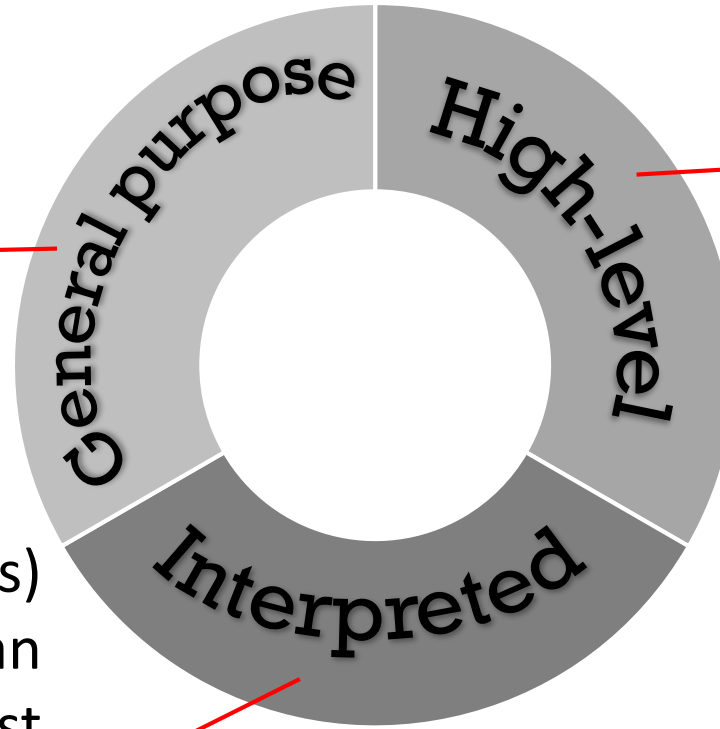
If you have any queries, please feel free to post at **Luminus Forum**.

What and Why of Python

Python is a **general-purpose**, **high-level** and **interpreted** language.

Widely-applicable
not fine-tuned to a
domain specific to
a perform a task.

Instructions (source codes)
are directly executed (by an
interpreter). It is not first
converted (by a compiler)
into a sequence of machine-
level primitive operations.
Errors are found at runtime.



Instructions use more abstract
operations that have been
provided by the language
designer. Low-level languages
require you to write instructions
and data objects at the level of
the machine and also allocate
and manage memory, whereas
Python manages memory for us.

Other reasons? Easier; concepts transferable;
cross-platform; industry sought after skill...

What you need: Software Setup

- Software platform: Anaconda
- Integrated development environment (IDE): Jupyter Notebook
- Language: Python

Installing Anaconda

The following steps will help you install Anaconda:

- Go to <http://www.anaconda.com/distribution>
- Click to download software according to your operating system and system bit (latest version: Python 3.9)
- Follow the steps in <https://docs.anaconda.com/anaconda/install/index.html> to complete the installation process.

What is Anaconda?

- A popular Python distribution software
- Everything you need for Python
- Primary use is data analytics and data science
- Includes
 - Core Python language and libraries
 - Jupyter Notebook
 - Anaconda's own package manager

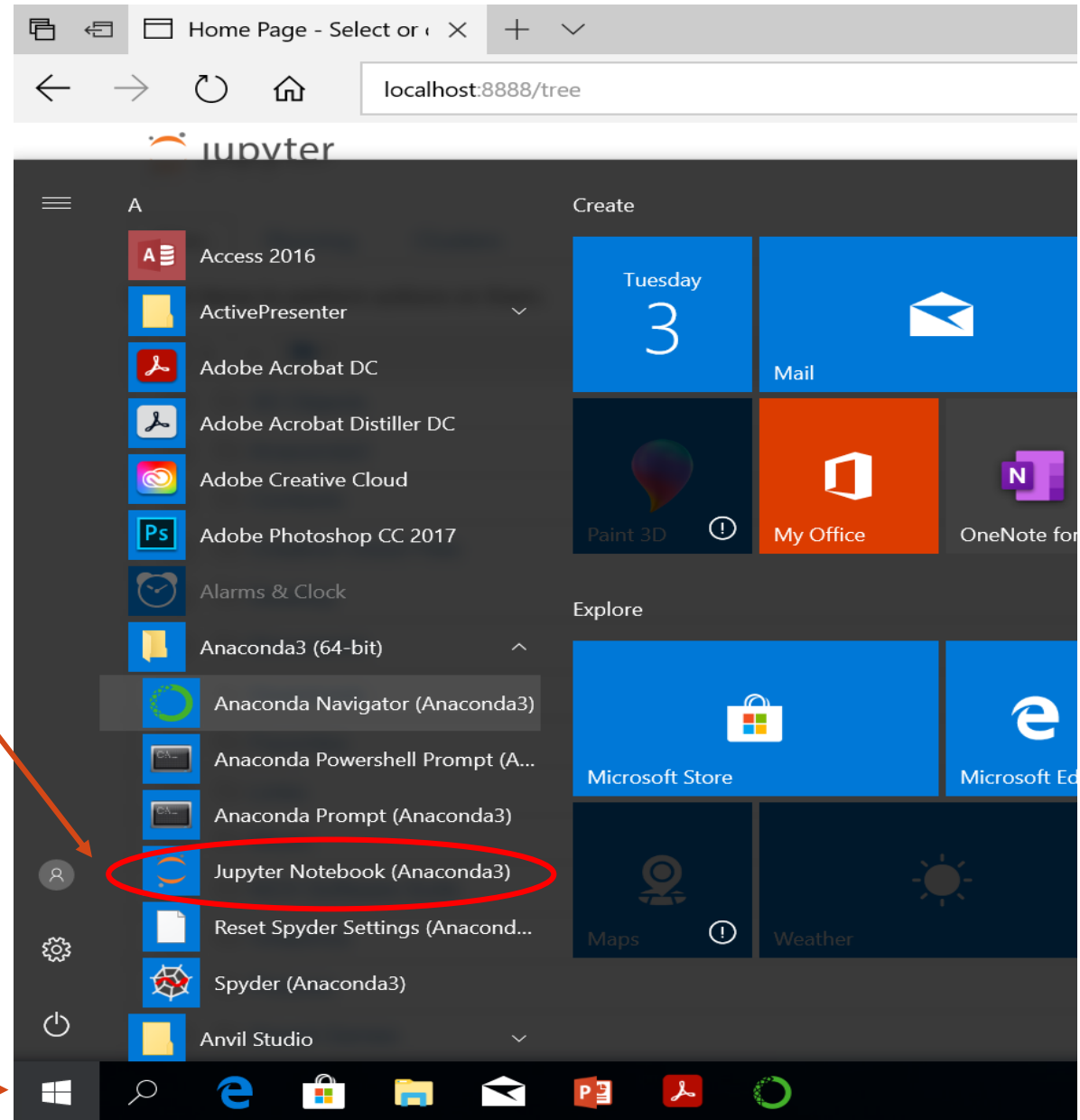
What is Jupyter Notebook?

- An open-source integrated development environment (ide) that allows:
 - Creating documents
 - Sharing documents
 - Live codes, equations, visualisations and narrative text to coexist
- Each document is known as a notebook (.ipynb)
- Python files (.py) could be generated easily
- IDEs are a tool for developers to code in (think of canvas for artists)

Launching Jupyter Notebook - via the Anaconda application

- Press the Windows Start button, search for and launch **Jupyter Notebook (Anaconda3)**

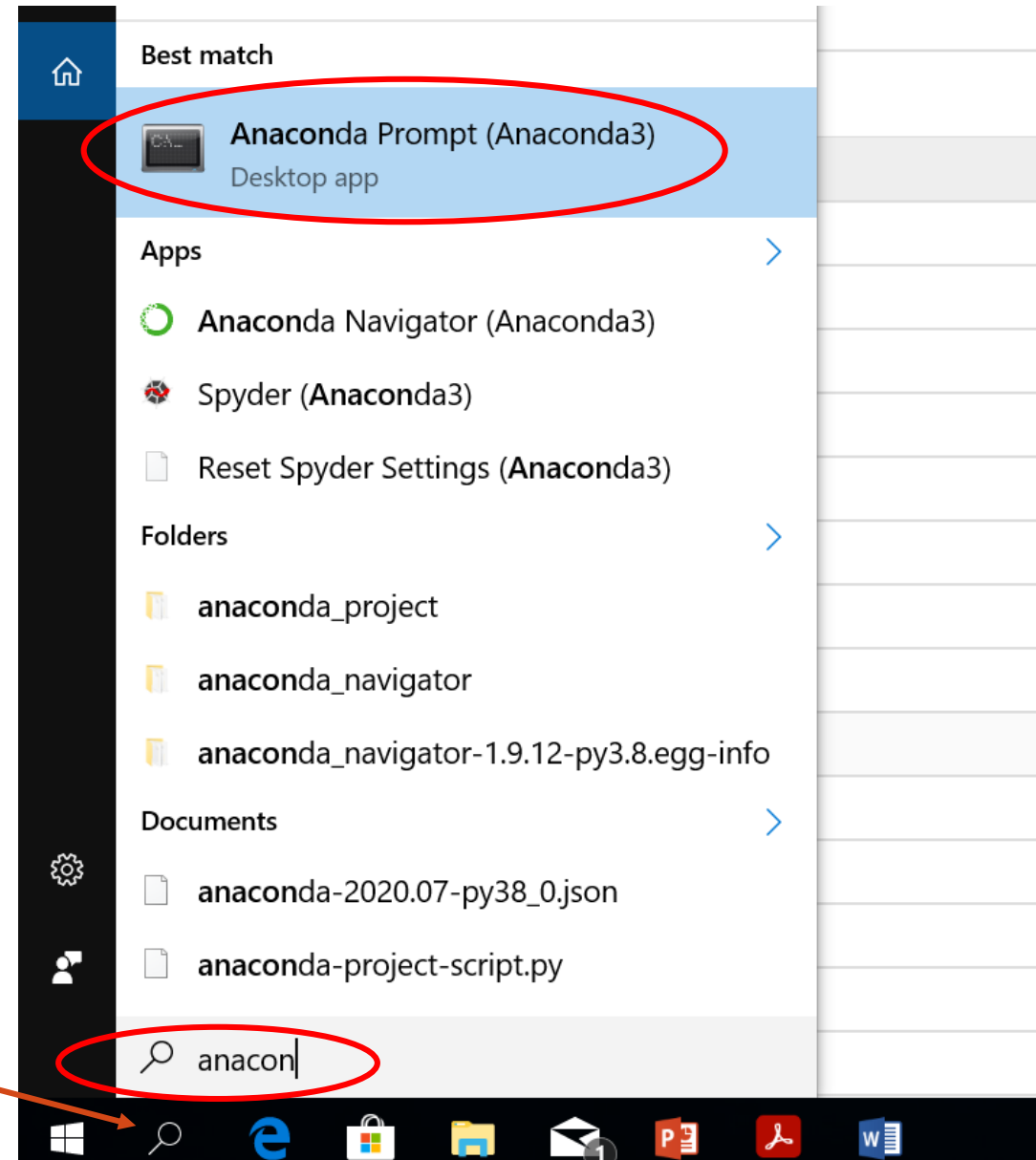
Start button →



Launching Jupyter Notebook - via the Command Prompt

- The Command Prompt (Windows) or Terminal (Linux and OS X) is a tool for users to issue commands to the computer through basic text.
- Press the Search button, type “Anacon” and click on **Anaconda Prompt (Anaconda3)**
- You can use it to
 - test Python code
 - run Jupyter Notebook

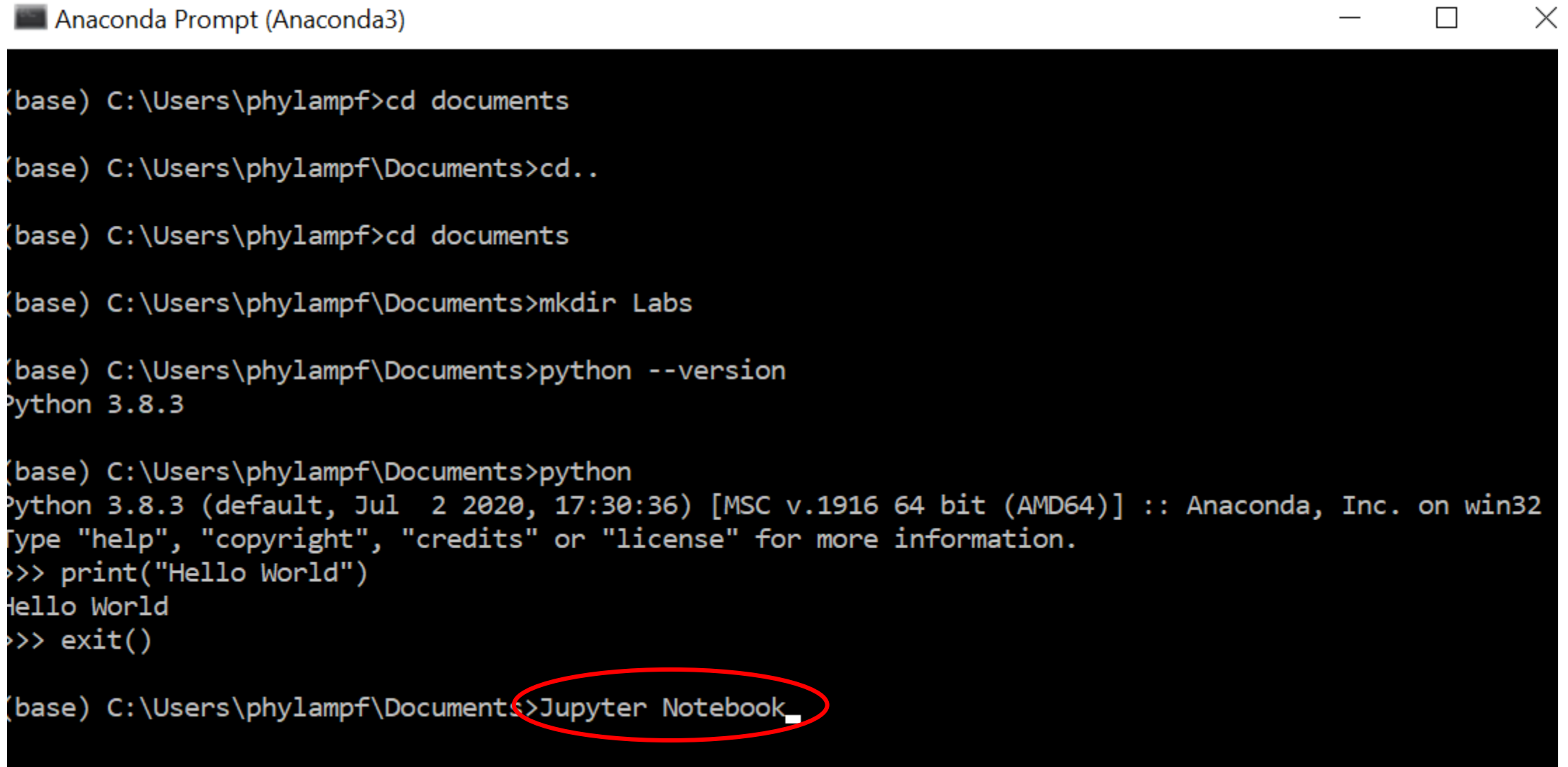
Search button



Using the Command Prompt/Terminal

- Some useful commands in the command prompt or terminal session
 - `cd` **change directory**
 - `cd documents` **navigate to the “documents” folder**
 - `cd ..` **go backward**
 - `mkdir Labs` **create a new folder called “Labs”**
 - `cls` **clears screen**
 - `python --version` **checks for the Python version**
 - `python` **activate Python shell**
 - `exit()` **exit from Python shell**
 - `Jupyter Notebook` **launch Jupyter Book**
 - `ctrl c (x2)` **exit from Python shell**
 - `exit` **exit from Anaconda Prompt**

Launching Jupyter Notebook (cont'd)



```
Anaconda Prompt (Anaconda3)

(base) C:\Users\phylampf>cd documents

(base) C:\Users\phylampf\Documents>cd..

(base) C:\Users\phylampf>cd documents

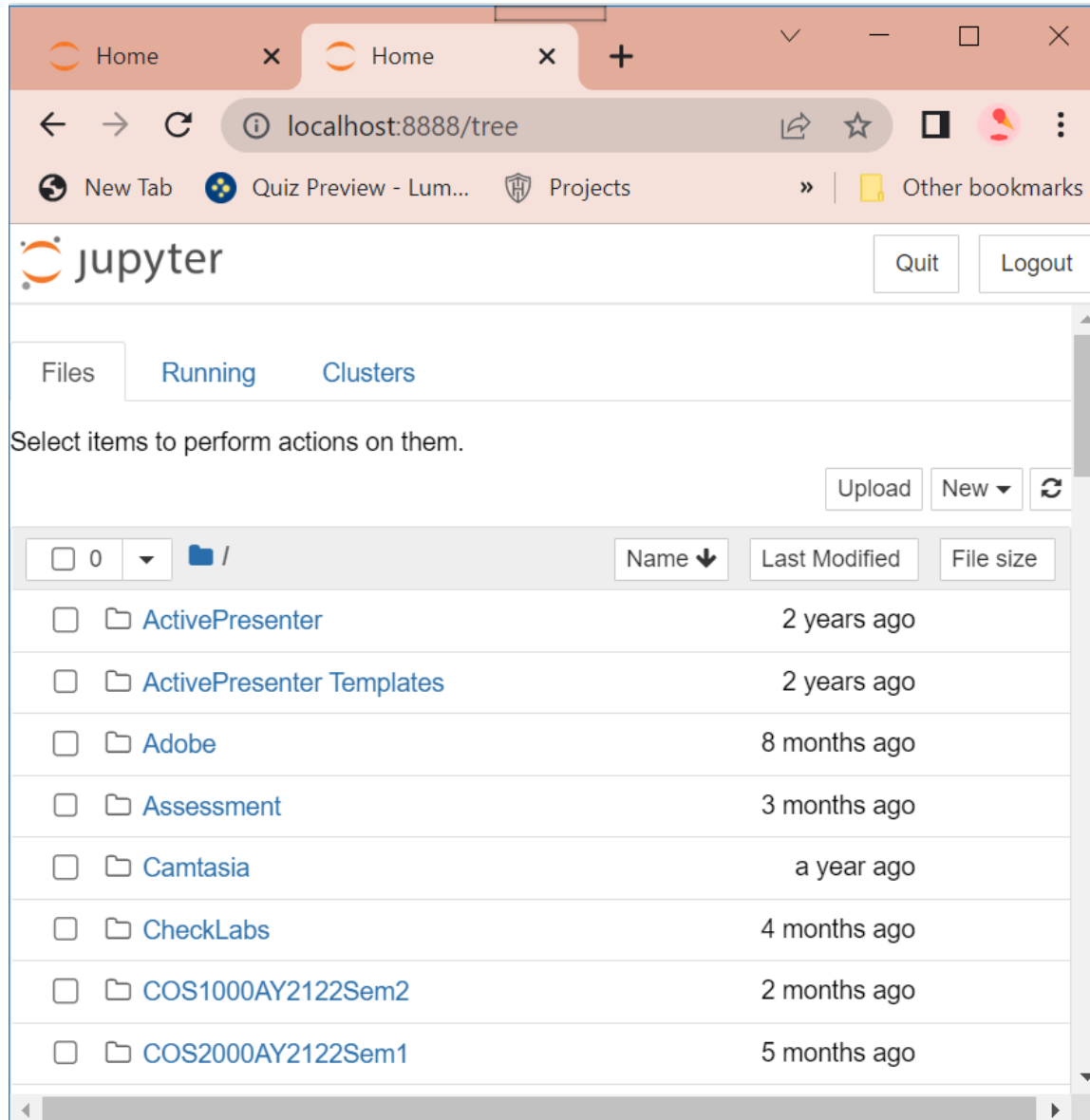
(base) C:\Users\phylampf\Documents>mkdir Labs

(base) C:\Users\phylampf\Documents>python --version
Python 3.8.3

(base) C:\Users\phylampf\Documents>python
Python 3.8.3 (default, Jul 2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello World")
Hello World
>>> exit()

(base) C:\Users\phylampf\Documents>Jupyter Notebook_
```

Launching Jupyter Notebook (cont'd)



- Your default browser will open at the Documents directory
- It is also the place to store all your folders and notebooks

Launching Jupyter Notebook (cont'd)

```
Select Anaconda Prompt (Anaconda3) - Jupyter Notebook

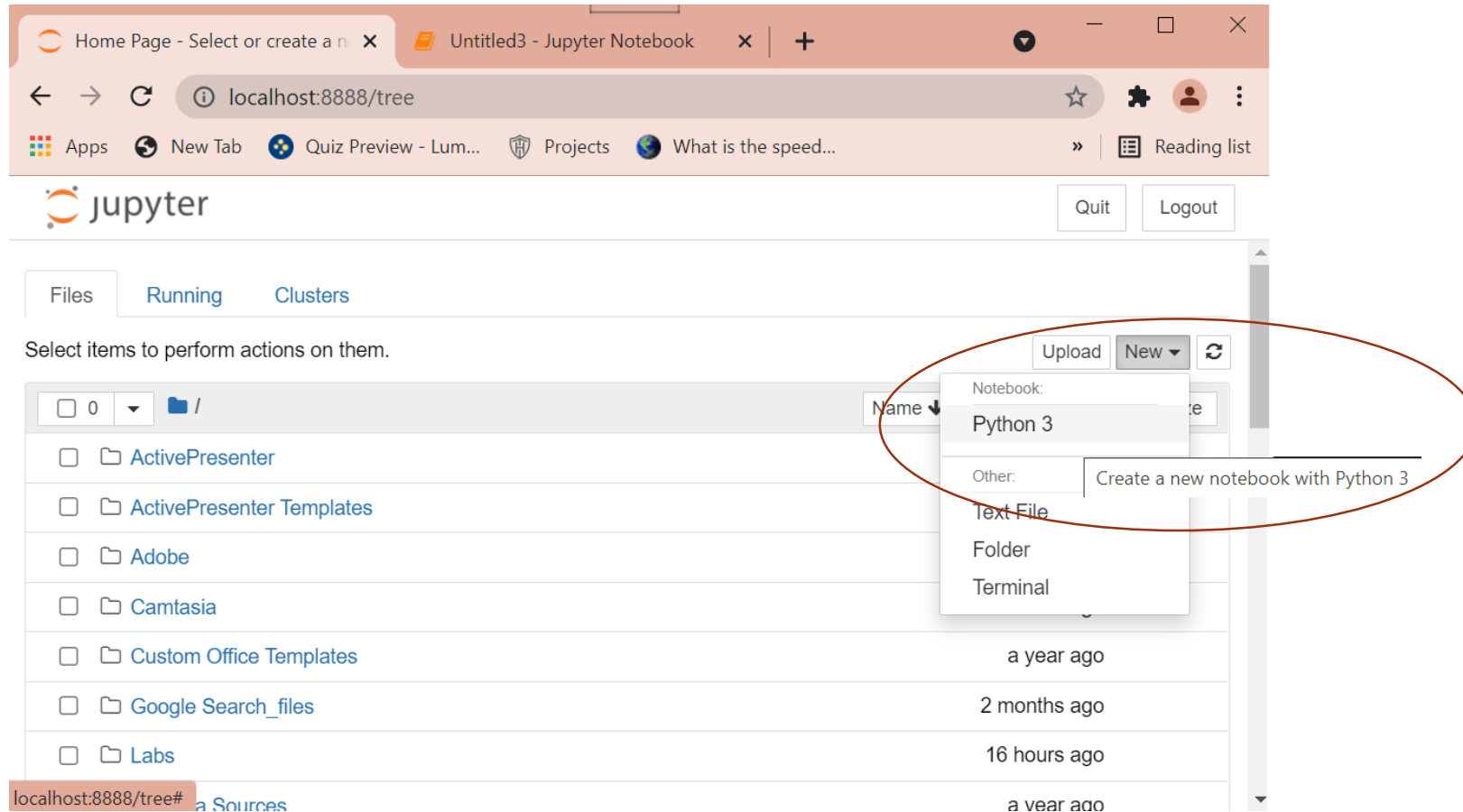
(base) C:\Users\phylampf>cd documents

(base) C:\Users\phylampf\Documents>Jupyter Notebook
[I 12:19:57.937 NotebookApp] JupyterLab extension loaded from C:\Users\phylampf\Anaconda3\lib\site-packages\jupyterlab
[I 12:19:57.939 NotebookApp] JupyterLab application directory is C:\Users\phylampf\Anaconda3\share\jupyter\lab
[I 12:19:57.942 NotebookApp] Serving notebooks from local directory: C:\Users\phylampf\Documents
[I 12:19:57.942 NotebookApp] The Jupyter Notebook is running at:
[I 12:19:57.942 NotebookApp] http://localhost:8888/?token=a4fb7195343b64a25418e58b7f319a0c3c8815c7c9298cc8
[I 12:19:57.942 NotebookApp] or http://127.0.0.1:8888/?token=a4fb7195343b64a25418e58b7f319a0c3c8815c7c9298cc8
[I 12:19:57.942 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 12:19:58.024 NotebookApp]

To access the notebook, open this file in a browser:
    file:///C:/Users/phylampf/AppData/Roaming/jupyter/runtime/nbserver-68444-open.html
Or copy and paste one of these URLs:
    http://localhost:8888/?token=a4fb7195343b64a25418e58b7f319a0c3c8815c7c9298cc8
    or http://127.0.0.1:8888/?token=a4fb7195343b64a25418e58b7f319a0c3c8815c7c9298cc8
```

You could also copy the url of localhost and paste it into any browser of your choice.

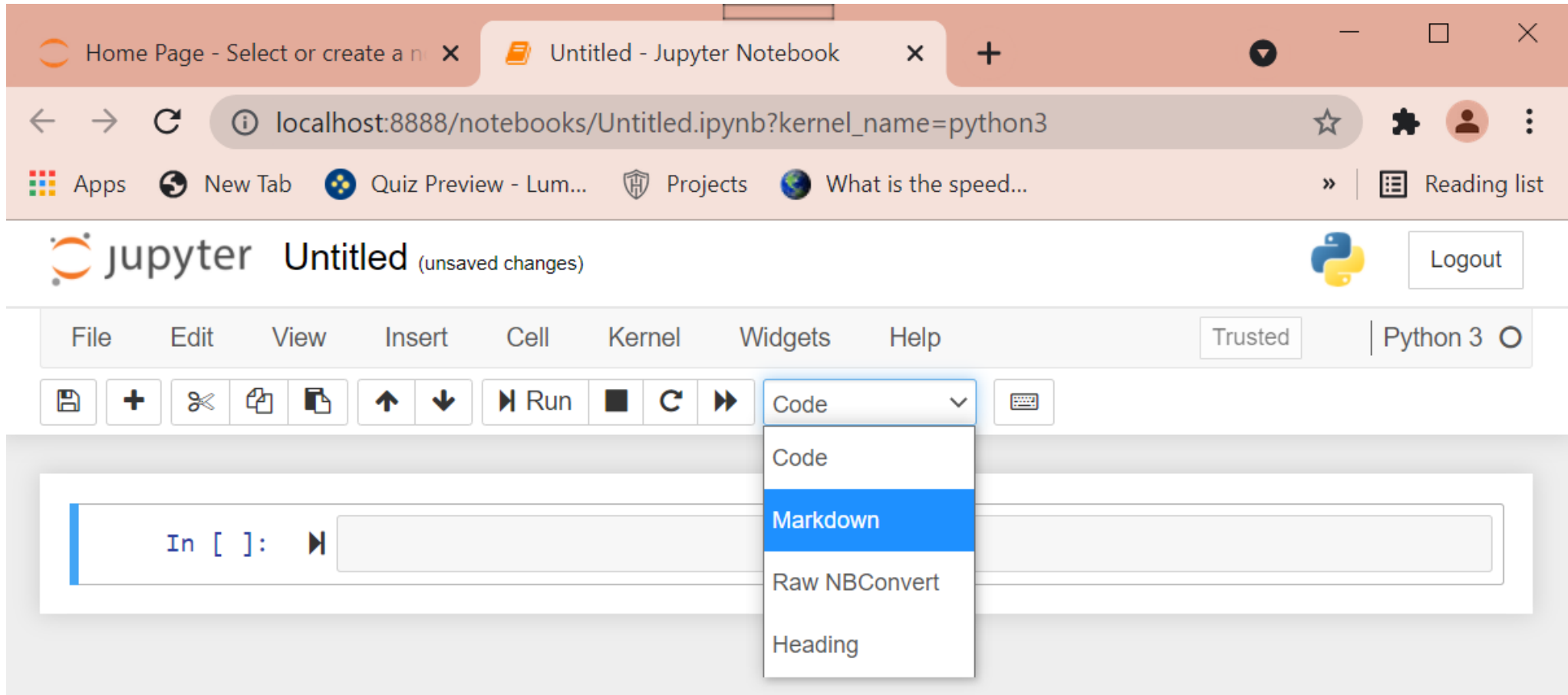
Creating a Python notebook



Click New → Python 3 to create a new Python 3 notebook



The notebook environment



Now we are ready to code!