COMP47670 Data Science in Python
Mixed Delivery version of COMP41680

Preliminary Material

Pádraig Cunningham

Based on Slides by Derek Greene

UCD School of Computer Science

Spring 2019

Overview

- Module Details
- Why Python?
- Installing Python 3 via Anaconda
- Running Python Code
- Using Jupyter Notebooks
  - Getting Started
  - Code Cells
  - Markdown Cells

Module Outline

- **Weeks 1-3:** Crash course in Python 3
  - Working with IPython Notebooks
  - Language fundamentals
  - Data structures
  - Input/output
- **Weeks 4-12:** Practical Data Science in Python
  - Introduction to data science
  - Collecting and preparing data
  - Numerical computing and statistics in Python
  - Machine learning in Python
  - Data visualisation

Schedule

Lectures Online

Workshops: H2.38 (overflow in H2.20) SCH 3-5pm Fridays

25/01, 15/02, 08/03, 12/04 (overflow in H2.40)

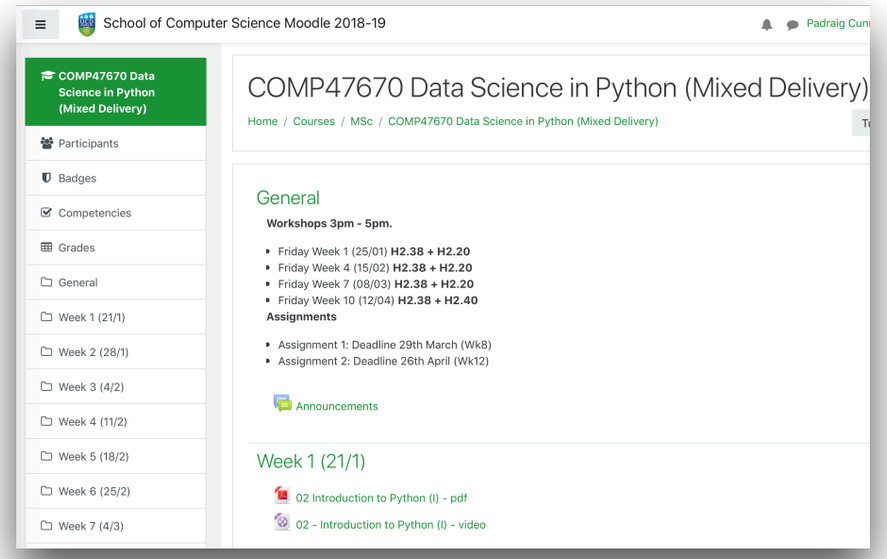
Notes, assignments, and additional material will be available on the CS Moodle page for COMP41680:

https://csmoodle.ucd.ie/moodle/course/view.php?id=702

Moodle page is currently open for registration via self-enrolment.
Password: **dsip2019**

Check that your Moodle details are all correct.

For module queries padraig.cunningham@ucd.ie



Practical Details

Module marked based on continuous assessment: 2 individual programming assignments. No end of semester exam.

50%	Assignment 1: Data Collection & Preparation
50%	Assignment 2: Data Exploration & Machine Learning

Late Submissions Policy:

- ! All assignment deadlines are hard deadlines.

  1-5 days late: 10% deduction from overall mark

  6-10 days late: 20% deduction from overall mark

  Not accepted after 10 without extenuating circumstances form or medical certificate.

Practical Details

CS grading scheme applies for this module. Pass mark is 40%.

Grade	Min	Max
A+	95	100
A	90	95
A-	85	90
B+	80	85
B	75	80
B-	70	75
C+	65	70
C	60	65
C-	55	60
D+	50	55
D	45	50
D-	40	45

Grade	Min	Max
E+	35	40
E	30	35
E-	25	30
F+	20	25
F	15	20
F-	10	15
G+	8	10
G	5	8
G-	2	5
NG	0	0

https://www.cs.ucd.ie/Grading

Plagiarism Policy

- Plagiarism is a **serious academic offence**.
- Our staff and demonstrators are proactive in looking for possible plagiarism in all submitted work.
- Suspected plagiarism is reported to the CS Plagiarism Subcommittee for investigation:
  - Usually includes an interview with student(s) involved
  - 1st offence: **usually** 0 or NG in the affected components
  - 2nd offence: referred to **University Disciplinary Committee**
- Students who enable plagiarism are equally responsible. See:

http://www.ucd.ie/registry/academicsecretariat/docs/plagiarism_po.pdf

http://www.ucd.ie/registry/academicsecretariat/docs/student_code.pdf

http://libguides.ucd.ie/academicintegrity

Why Python?

- Open source, freely available online
- Clean, concise, unambiguous syntax
  - Often referred to as "executable pseudo-code"
- Supports Rapid Application Development
- Supports a variety of programming paradigms
  - Simple scripts
  - Modular & Object-oriented programming
  - Interactive notebooks
- Strong library support
  - Comprehensive built-in library provides many functions
  - Many third-party packages available, particularly for numerical computing, data analysis, and visualisation.

Installing Python

- In the module we will use **Python 3.7**
- Python 3.x is recommended for new code and fixes many of the issues and inconsistencies from Python 2.
- Be aware: Python 3.x code is not fully backwards compatible with Python 2.
- Install Python via the **Anaconda** distribution which provides a version of Python tailored for data analytics, with easy installation of many third party packages.

https://www.anaconda.com/download/

- Download and run Anaconda the graphical or terminal installer for Python 3.7 (not 2.7!) for your operating system - Windows, OSX or Linux.

Running Python Code

Several different ways to run Python code…

1. Type `python` at the terminal to start the basic Python interactive shell.

```
Last login: Fri Jan 11 10:12:38 on ttys001
MacBook-Air:~ padraigcunningham$ python
Python 3.7.1 (default, Dec 14 2018, 13:28:58)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

2. Type `jupyter notebook` at the terminal to start the Jupyter Notebooks.

```
Last login: Sat Jan 12 16:23:41 on ttys002
MacBook-Air:~ padraigcunningham$ jupyter notebook
[I 16:25:36.286 NotebookApp] JupyterLab extension loaded from /anaconda3/lib/
hon3.7/site-packages/jupyterlab
[I 16:25:36.287 NotebookApp] JupyterLab application directory is /anaconda3/
e/jupyter/lab
[I 16:25:36.288 NotebookApp] Serving notebooks from local directory: /Users/
```

3. Run full script files line by line from the terminal using:

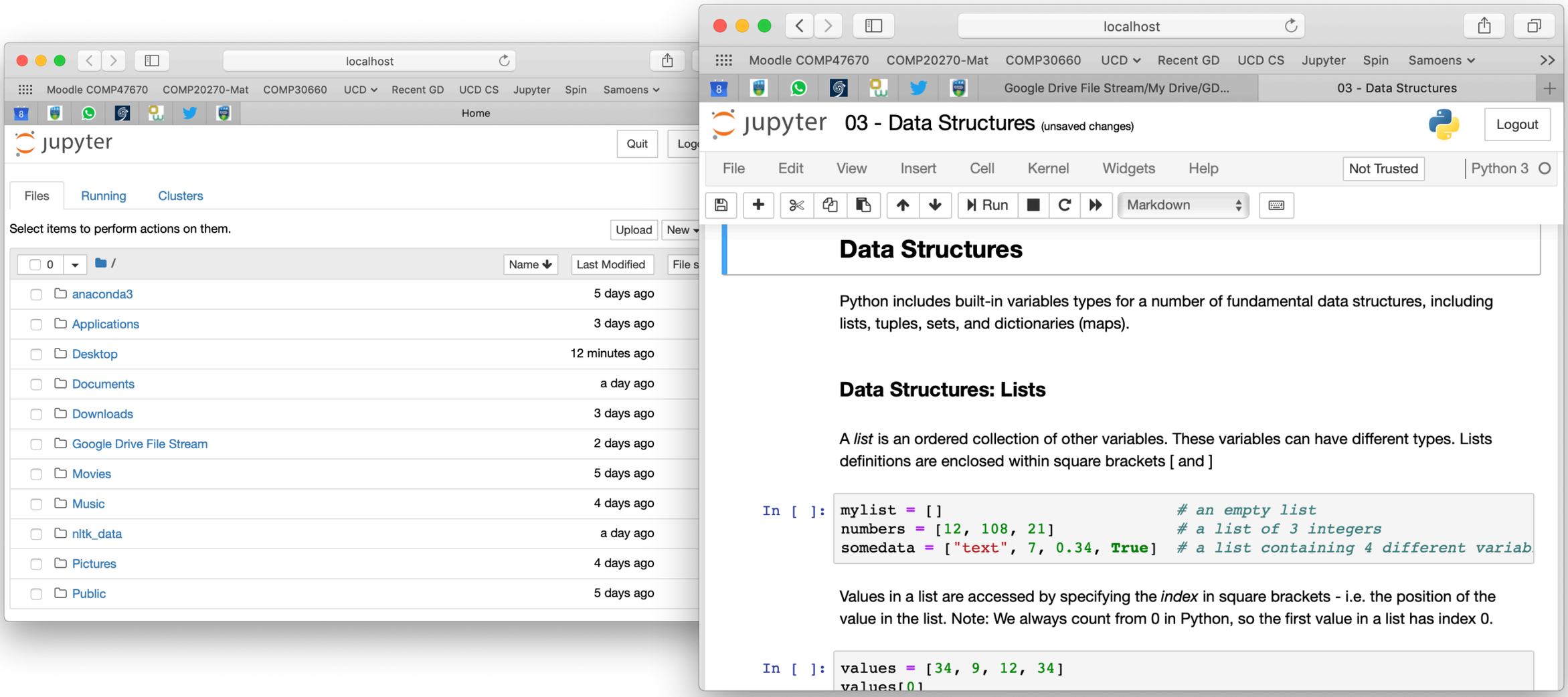
```
> python hello.py
Hello World
>
```

```
python <script_file.py>
```

4. Use web-based interactive notebooks…

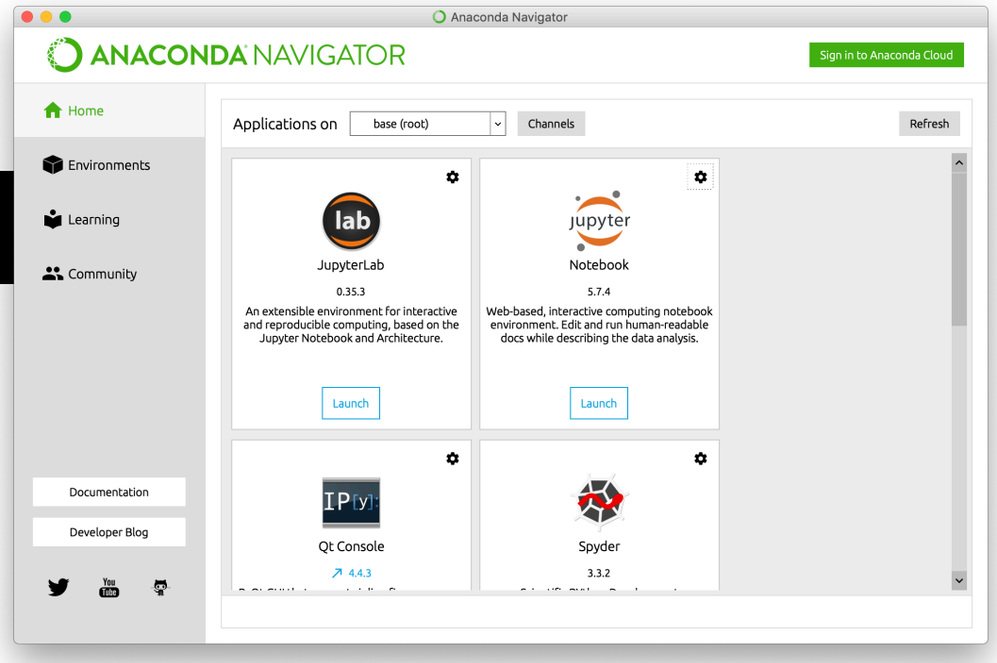
Jupyter Notebooks

- **Basic idea:** Rather than using an editor or development environment, use an interactive browser-based environment to learn programming.
- Online notebooks are now increasingly used in commercial settings (e.g. data science teams).



Jupyter & IPython Notebooks

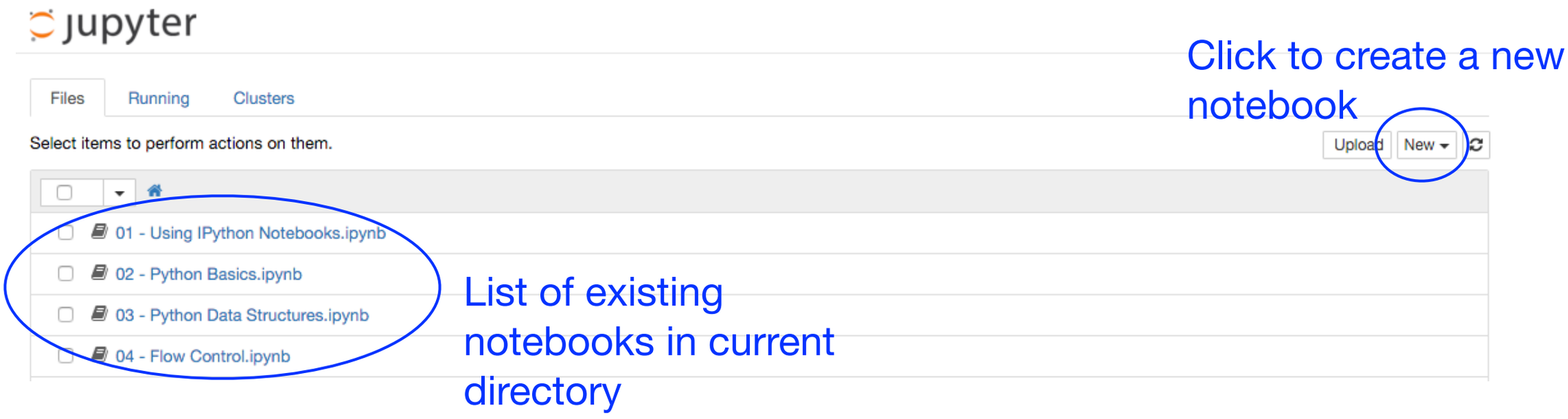
- **Jupyter project:** a web application for interactive data science and scientific computing.
- **IPython Notebooks:** an engine for running Python code under the Jupyter system.
- We will use IPython Notebooks for many of the labs and assignments in COMP47670.
- To start the Notebook server, either:
  1. In the terminal, type `jupyter notebook`
  2. Or click the Anaconda Navigator icon, then choose **jupyter-notebook** from the list of apps.



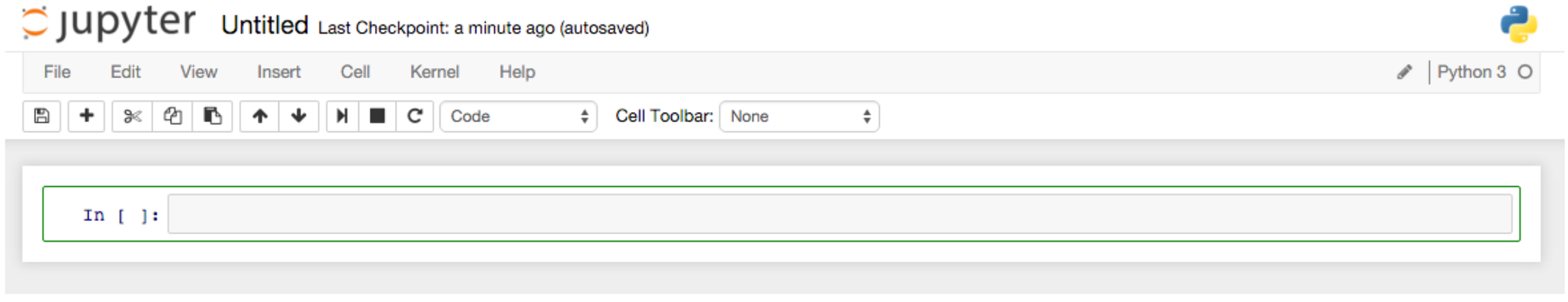
- This should load the IPython Notebook dashboard in your browser. Later you can also manually go to http://localhost:8888

Notebook Dashboard

- The IPython dashboard provides a mini filesystem interface for creating and accessing notebooks.
- Note: The dashboard shows notebooks in the directory where you launched the notebook server.

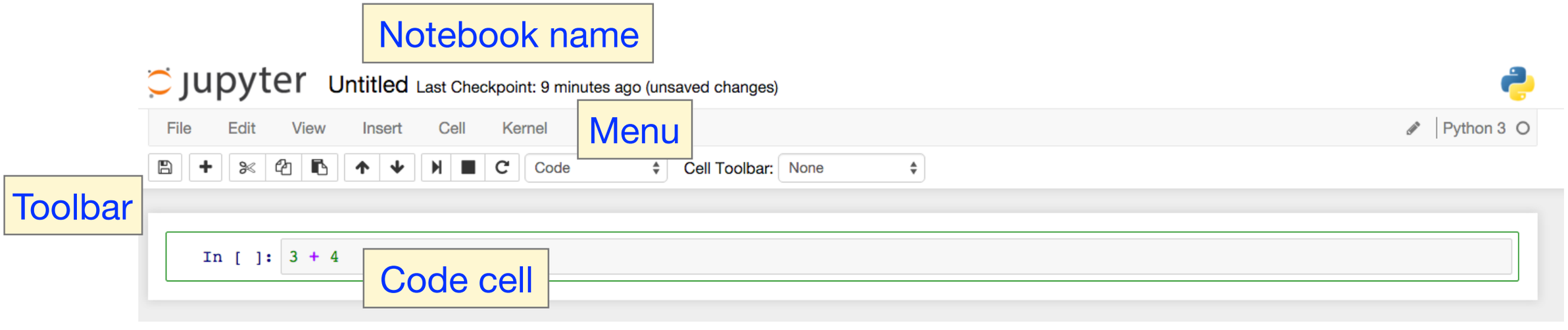


- To start writing code, create **New → Python 3 Notebook**

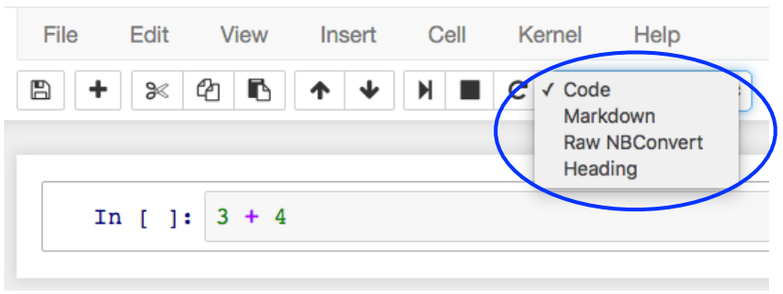


Notebook Interface

- When you create a new notebook, you will be presented with the notebook name, a menu bar, a toolbar and an empty code cell.



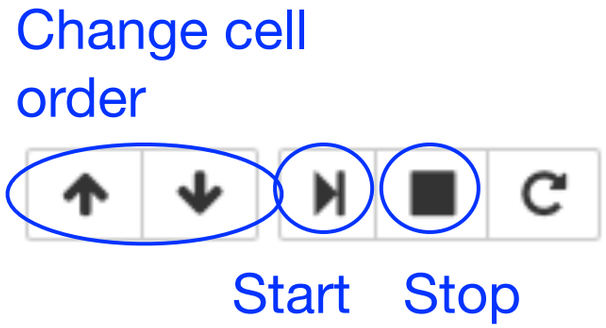
- IPython notebooks have two fundamental types of cells:
  1. **Markdown cells:** Contain text content for explaining a notebook.
  2. **Code cells:** Allow you to type and run Python code.



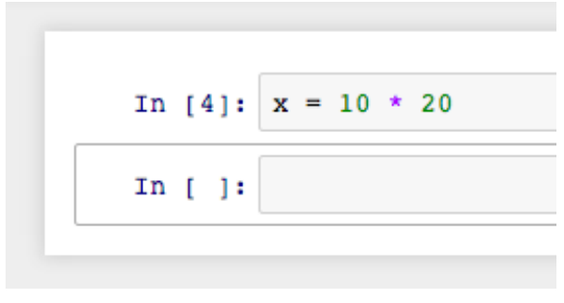
Every new cell starts off being a code cell. But this can be changed by using the drop-down on the toolbar

Code Cells

- In a code cell, you can enter one or more lines of Python code. Run the code by hitting Shift-Enter or by pressing the **Play** button in the toolbar.



- You can modify and re-run code cells multiple times in any order.
- When a code cell is executed, the code it contains is sent to the **kernel** associated with the notebook - i.e. the Python instance running in the background.
- The results returned from this computation are displayed as the cell's output. Note that some code will not have an output.



- Restarting the kernel associated with a notebook clears all previous history (e.g. variable values).

Markdown Cells

- It can be helpful to provide explanatory text in notebooks.
- **Markdown** is a lightweight type of markup language with plain text formatting syntax which can be rendered as HTML.
- IPython supports a set of common Markdown commands. HTML tags and LaTeX formulae can also be included.
- When a Markdown cell is executed, the Markdown code is converted into the corresponding formatted rich text.

```
This is normal text.
```

This is normal text.

```
*This is italics*.
```

This is italics.

```
And **this is bold**.
```

And **this is bold**.

```
# Heading 1
## Heading 2
### Heading 3
```

Heading 1
Heading 2
Heading 3

```
Example <font color='red'>HTML use</font>
```

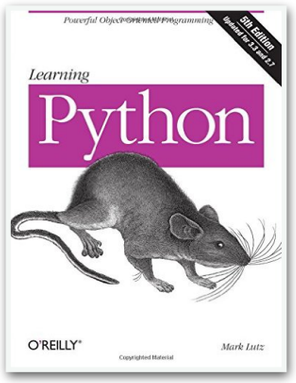
Example HTML use

```
Formula: $x=\frac{y}{z}$
```

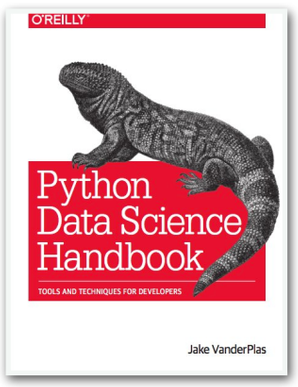
Formula: $x = \frac{y}{z}$

Book Resources

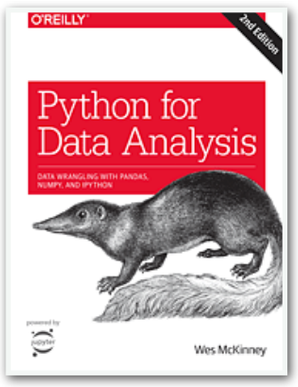
No single textbook for this module. A range of good Python books are available. Make sure the book covers Python 3.x.



Learning Python, 5th Edition
Mark Lutz
http://shop.oreilly.com/product/0636920028154.do



Python Data Science Handbook
Jake VanderPlas
http://shop.oreilly.com/product/0636920034919.do



Python for Data Analysis, 2nd Edition
William McKinney
http://shop.oreilly.com/product/0636920050896.do

Online Resources

- **Python**
  - Official Python 3 documentation
    https://docs.python.org/3/
  - SciPy lectures notes
    http://www.scipy-lectures.org
- **IPython Notebooks**
  - Official documentation
    http://ipython.readthedocs.org/en/stable/overview.html
- **Markdown**
  - Github guide to Markdown
    https://help.github.com/articles/markdown-basics
  - Original Markdown syntax specification
    http://daringfireball.net/projects/markdown/syntax/