COMP41680

Preliminary Material

Derek Greene

UCD School of Computer Science

Spring 2023

Module Outline

- **Part 1: Crash Course in Python**
  - Working with Jupyter Notebooks
  - Language fundamentals and data structures
  - Python standard library
- **Part 2: Practical Data Science in Python**
  - Introduction to data science
  - Collecting and storing data
  - Data preparation manipulation
  - Numerical computing and statistics in Python
  - Data visualisation
  - Time series analysis
  - Regression and classification

Module Schedule

Trimester 2A	Week 1	01 - Preliminary Material 02 - Introduction to Python
	Week 2	03 - Next Steps in Python
	Week 3	*Bank Holiday - No lectures and labs*
	Week 4	04 - Introduction to Data Science
	Week 5	05 - Data Formats and Collection
	Week 6	06 - Data Preparation and Manipulation
	Week 7	07 - More Advanced Pandas 08 - Data Visualisation
Trimester 2B	Week 8	09 - Numerical Computing
	Week 9	10 - Modelling and Prediction 11 - Classification and Evaluation
	Week 10	*Bank Holiday - No lectures and labs*
	Week 11	12 - Text Mining
	Week 12	13 - Time Series *Class Test*

Module Delivery

- *Lectures*: Monday 12:00 - **Room B002, CS Building**
- *Labs*: Monday 14:00 - **Room 321, Engineering Building**
- Labs start in **Week 1 (Monday 23rd January)** and will involve practical programming tasks related to data science.
- Note there will be no lectures or labs on **Monday 6th February** and **Monday 10th April** due to Bank Holidays.
- Instead, there will be additional classes on:
  - Extra Lab: **Friday 31st March**, 12:00. Room B002, CS
  - Extra Lecture: **Thursday 6th April**, 12:00. Room B106, CS
- All lecture and lab materials will be posted on the COMP41680 Brightspace page: https://brightspace.ucd.ie/d2l/home/184810

Module Assessment

Module marked based on two individual programming assignments and one class test. No end-of-semester exam.

40%	Assignment 1	Available: Mid February Due: Mid March
40%	Assignment 2	Available: Start April Due: End of Trimester
20%	Class Test	Friday 28th April, 17:00 (in person in UCD) Sample test will be provided in advance

Late Submissions Policy:

- All assignment deadlines are hard deadlines.
- 1-5 days late: 1 grade point deduction, e.g. B to B-
- 6-10 days late: 2 grade point deduction, e.g. B to C+
- Assignments will not be accepted after 10 days without Extenuating Circumstances formally approved by UCD.

Module Grading

The standard UCD CS grading scheme applies for this module. 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://csintranet.ucd.ie/CSGrading/

Plagiarism Policy

- Plagiarism is a **serious academic offence**.
- Staff & demonstrators are proactive in looking for possible plagiarism
- Suspected plagiarism is investigated by the CS Plagiarism subcommittee
  - Usually includes an interview with the student(s) involved
  - 1st offence: typically 0 or NM in the affected components
  - 2nd offence: more serious consequences (e.g. UCD Disciplinary process)
- Student who enables plagiarism is equally responsible for it. All students in a group which plagiarises are held responsible for it.
- Examples of plagiarism:
  - Copying some/all of the work of another student and submitting it as your own work
  - Copying some/all of an assignment from the Internet/book/etc without referencing it
  - Sharing individual work with another student (by email, Facebook, WhatsApp, ...)
  - Making your work available (on GitHub, website, social media, ...) before lecturer gives permission
  - Students collaborating at too detailed a level (e.g. consulting each other after implementing a segment of code)

Extenuating Circumstances

- Applications for Extenuating Circumstances
  - Refers to grave issues that occasionally arise such as:
    - Serious illness, hospitalisation, an accident.
    - Family bereavement.
    - Ongoing serious personal or emotional circumstances.
  - Extenuating Circumstances do not cover events which are foreseen (e.g. 21st party, wedding, personal travel etc).
- Minor Circumstances (absent for a few days)
  - These situations should be handled locally by making direct contact with the lecturer and or school administrator.
  - Extenuating Circumstances do **not** apply in these cases.

Introduction - 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 and object-oriented programming
  - Interactive web 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 3.9 or higher**
- Python 3 is recommended for new code and fixes many of the issues and inconsistencies from Python 2.
- Be aware: Python 3 code is not fully backwards compatible with Python 2.
- Install Python via the **Anaconda Individual Edition** distribution which provides a version of Python tailored for data analytics, with easy installation of many third party packages.

  https://www.anaconda.com/products/individual

- Download and run Anaconda the graphical or terminal installer for **Python 3.9 or higher** for your operating system - Windows, Mac 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.

```
> python
Python 3.9.7 (default, Sep 16 2021, 08:50:36)
[Clang 10.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 10 * 20
200
>>>
```

2. Type `ipython` at the terminal to start the enhanced IPython interactive shell.

```
> ipython
Python 3.9.7 (default, Sep 16 2021, 08:50:36)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.29.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: 10 * 20
Out[1]: 200

In [2]:
```

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

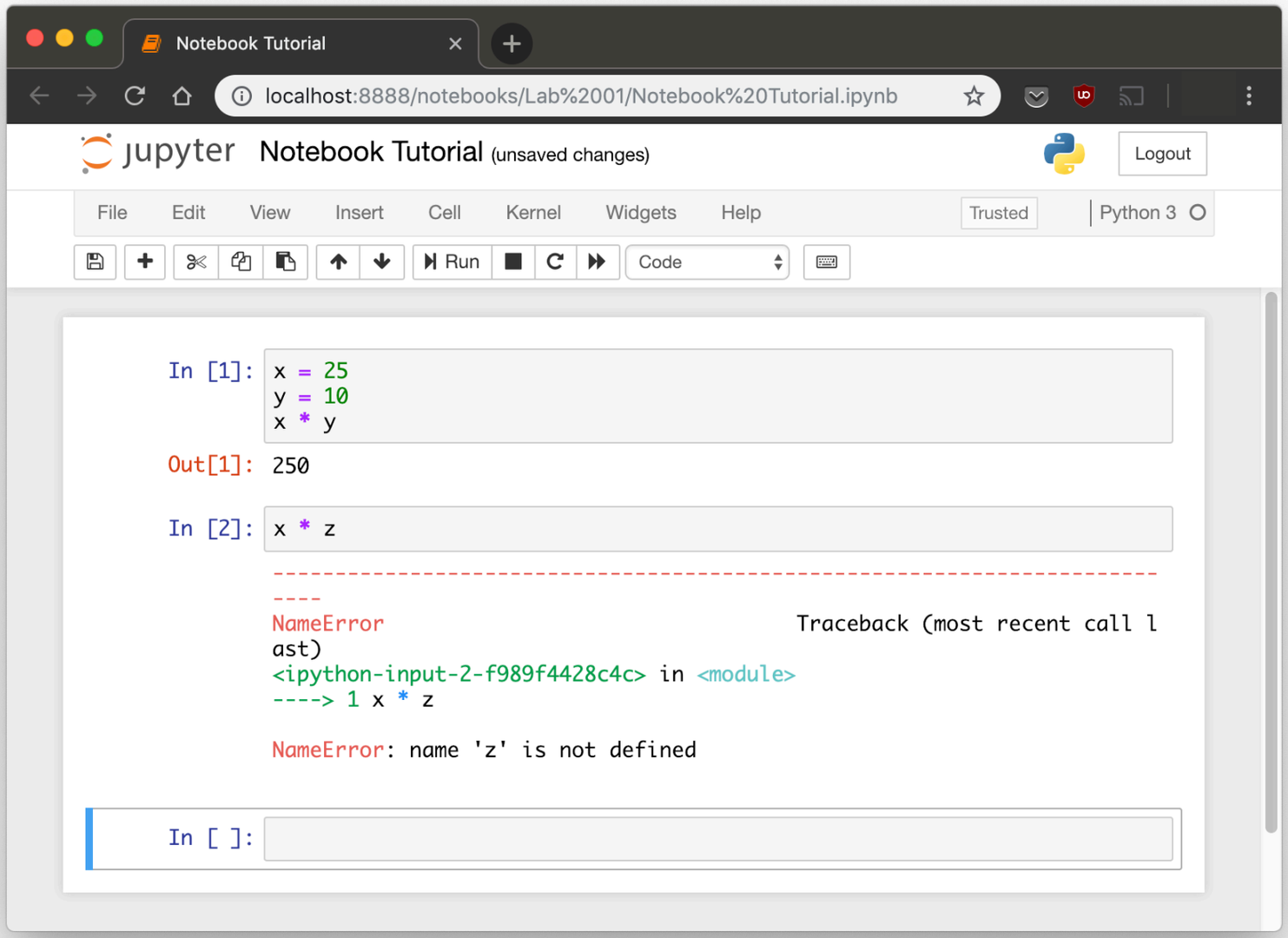
```
python <script_file.py>
```

```
> python calc.py
The answer is 200
>
```

4. Run Python code in web-based interactive notebooks…

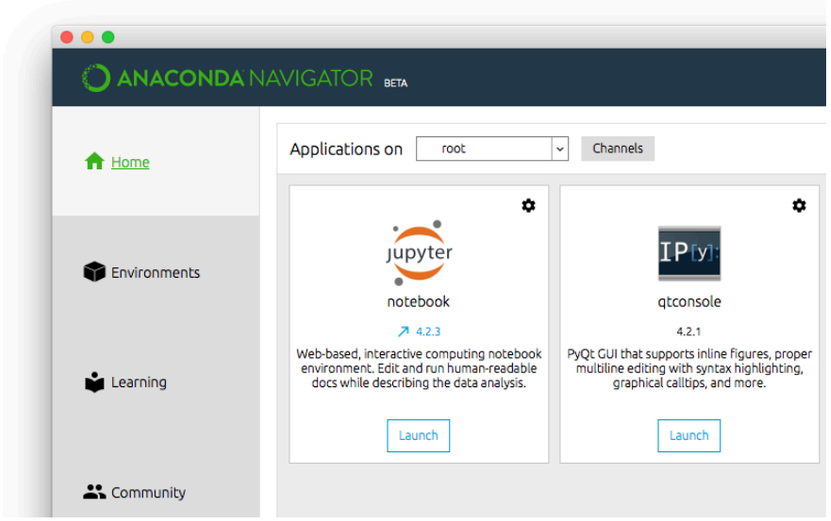
Jupyter Notebooks

- Rather than using an editor or development environment, use an interactive browser-based environment to learn programming.
- Online notebooks are now increasingly used in both academic and industrial settings (e.g. data science teams).



Jupyter Notebooks

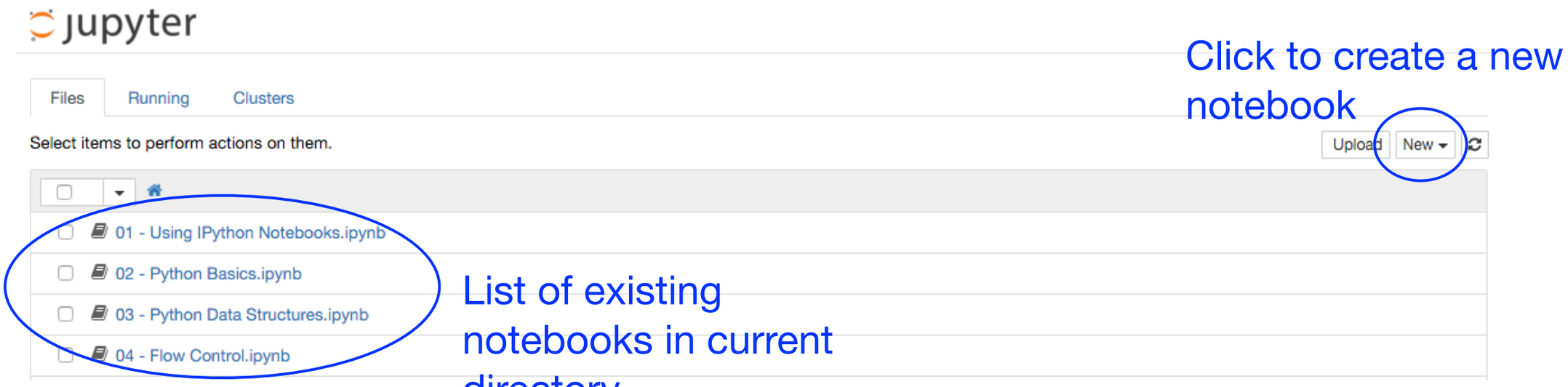
- **Jupyter Notebooks** (also called IPython Notebooks) provide a web environment for interactive data science and scientific computing: https://jupyter.org
- These notebooks communicate with a Python engine to run code and display the results.
- We will use Jupyter Notebooks for many of the labs and assignments in COMP41680.
- 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.



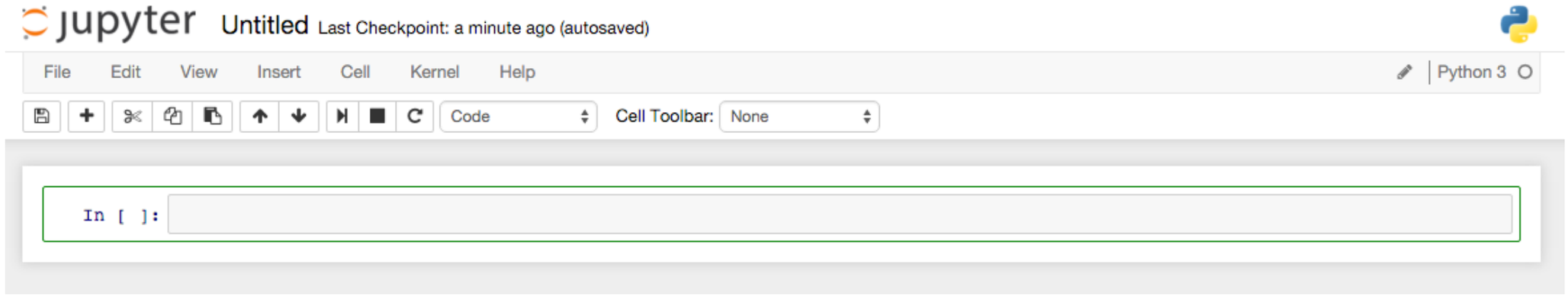
- This should load the Jupyter Notebook dashboard in your browser. Later you can also manually go to http://localhost:8888

Notebook Dashboard

- The Jupyter 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.

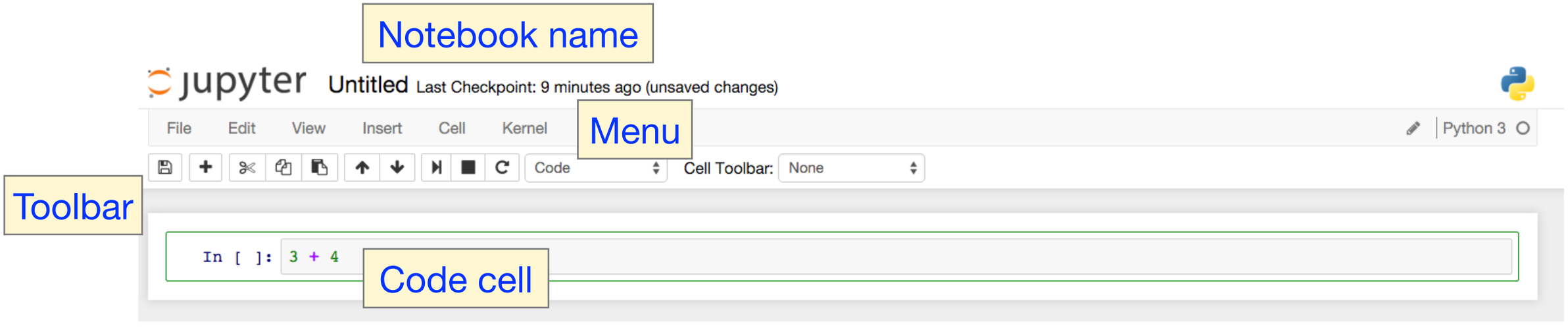


- To start writing code, create **New → Python 3 Notebook**

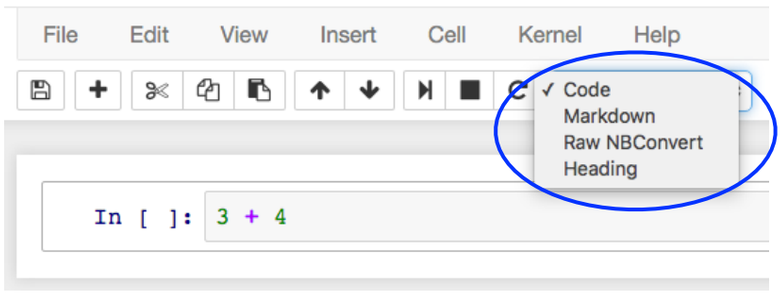


Notebook Interface

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



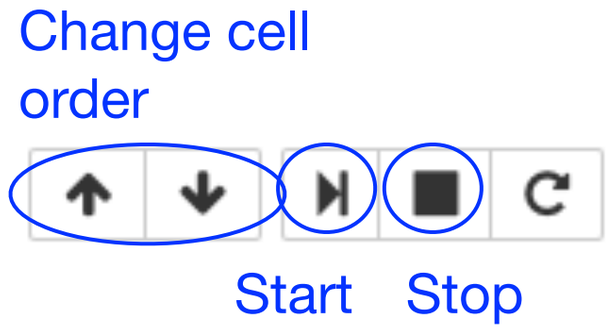
- Jupyter notebooks have two fundamental types of cells:
  1. **Markdown cells:** Contain text content for explaining a notebook.
  2. **Code cells:** Allow you to type and run Python code.



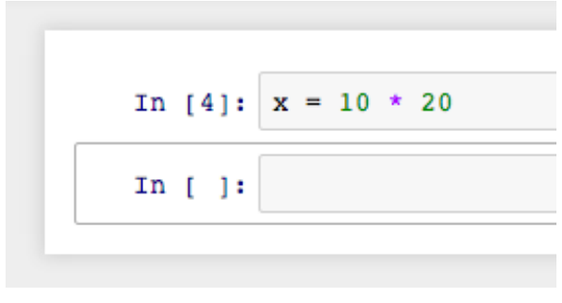
Every new cell starts off being a code cell. But this can be changed by using the drop-down on the toolbar

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.
- Jupyter 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>
```

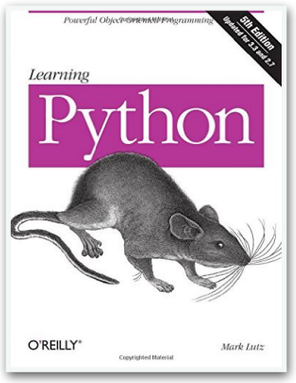
Example HTML use

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

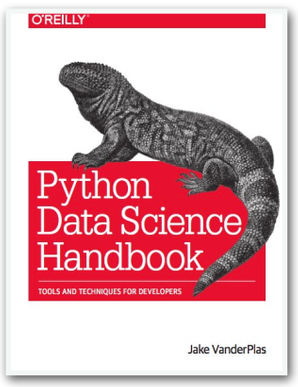
Formula: $x = \frac{y}{z}$

Book Resources

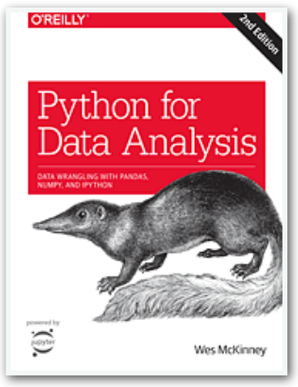
No single textbook for this module. A range of good Python books are available. Make sure the book covers Python 3.



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
- **Jupyter Notebooks**
  - Official Jupyter documentation
    https://jupyter.org/documentation
- **Markdown**
  - Github guide to Markdown
    https://docs.github.com/en/github/writing-on-github/getting-started-with-writing-and-formatting-on-github
  - Original Markdown syntax specification
    http://daringfireball.net/projects/markdown/syntax/