

Using notebooks

September 13, 2020

1 Jupyter notebooks

This is a *Jupyter notebook*. We will be computing using Jupyter notebooks (<https://jupyter.org/>) and the programming language *Python* (<https://www.python.org/>). Jupyter notebooks provide an interactive environment where you can mix text, equations, computer code and visual outputs. This is new technology that is increasingly widely used, and it is all free and open-source.

2 Using the Azure notebook service

The Azure notebook service provides a cloud-based Jupyter notebook environment at <https://notebooks.azure.com/>. You can log in with your CRSid.

2.1 Running and viewing the course material

There are 12 notebook activities for this course, plus some other helper notebooks (including this one), arranged in a library.

- Click on a notebook to view it.
- Click ‘Clone’ to get your own, runnable and editable copy of the library. If you are not logged in you will be prompted to log in. Use your CRSid.

Experiment freely with your clone of the activity notebooks - you can always return to the master version.

Once you have your own version of the notebooks, open a notebook and run it via the menu bar with **Kernel -> Restart & Run All**. Try making an edit and then run the notebook again to see your change.

2.2 Creating your own notebooks

To create your own notebooks:

1. Log in at <https://notebooks.azure.com/>.
2. Go to ‘Libraries’ (top left of the page).
3. Click ‘New Library’ to create a library (give it a suitable name, e.g. ‘IA exercises’).
4. You should now be in your new library. Click ‘Open in Jupyter’ to get a Jupyter environment.
5. At the top-right of the page, click **New -> Python 3**. You will now have a new notebook, and you’re ready to start working.

By default, the name of your new notebook will be ‘Untitled’. Click on ‘Untitled’ to give your notebook a different name, e.g. ‘Activity 02 exercises’.

You can return to <https://notebooks.azure.com/> at any time, and go to ‘Libraries’ to see your collection of notebook libraries.

It would be a good idea to periodically download your notebooks for safe keeping.

3 Editing and running notebooks

Jupyter notebooks have *text* cells and *code* cells. If you double-click on part of a notebook in a Jupyter environment (see above for creating a Jupyter environment on Azure), the cell will become editable. You will see in the menu bar whether it is a text cell (‘Markdown’) or a code cell (‘Code’). You can use the drop-down box at the top of a notebook to change the cell type. You can use **Insert** from the menu bar to insert a new cell.

The current cell can be ‘run’ using **shift-return** (the current cell is highlighted by a bar on the left-hand side of the page). When run, a text cell will be typeset, and the code in a ‘code cell’ will be executed. Any output from a code cell will appear below the code.

Often you will want to run all cells from the start of a notebook. You can do this with **Kernel -> Restart & Run All** from the notebook menu bar. In this case the cells are executed in order (first through to last).

Below is a code cell:

```
[1]: print(3 + 4)
```

7

4 Formatting text cells

Text cells are formatted using *Markdown*, and using *LaTeX* syntax for mathematics. Make extensive use of text cells to explain what your program does, and how it does it. Use mathematical typesetting to express yourself mathematically.

4.1 Markdown

You can find all the details in the [Jupyter Markdown documentation](#). Below is a brief summary.

4.1.1 Headings

Using Markdown, headings are indicated by ‘#’:

```
# Top level heading
## Second level heading
### Third level heading
```

4.1.2 Text style

The Markdown input

Opening passage

``A passage of text``

`*Some more text*`

`**Yet more text**`

appears as:

Opening passage

A passage of text

Some more text

Yet more text

4.1.3 Lists

You can create bulleted lists using:

- Option A
- Option B

to show - Option A - Option B

and enumerated lists using

1. Old approach
1. New approach

to show 1. Old approach 1. New approach

Markdown resolves the list number for you.

4.1.4 Code

Code can be typeset using:

```
```python
def f(x):
 return x*x
```
```

which produces

```
def f(x):
    return x*x
```

You can include images in Jupyter notebooks - see [Jupyter Markdown documentation](#).

4.2 LaTeX

Markdown cells support [LaTeX](#) syntax for typesetting mathematics. LaTeX is the leading tool for technical documents and presenting mathematics, and it is free.

Learning LaTeX now is a good investment for later in the Tripos. You might want to try <https://www.overleaf.com/> for an online LaTeX environment.

To typeset an inline equation, use:

The term of interest in this case is $\exp(-2x) \sin(3x^4)$.

which will appear as:

‘The term of interest in this case is $\exp(-2x) \sin(\alpha x^4)$.’

For a displayed equation, from

We wish to evaluate

$$f(x) = \beta x^3 \int_0^2 g(x) \, dx$$

when $\beta = 4$.

we get:

‘We wish to evaluate

$$f(x) = \beta x^3 \int_0^2 g(x) \, dx$$

when $\beta = 4$.’

Search online to learn the LaTeX commands for different mathematical symbols. If you see an example of mathematical typesetting in a notebook, you can also double-click it in a Jupyter environment to see the syntax. There are lots of examples at <https://notebooks.azure.com/library/1A-maths>.

5 Running Jupyter locally (optional)

See [here](#) for suggestions on running Jupyter locally.