



2/22/2023

Mathematics Association of Nairobi University  
isaak@students.uonbi.ac.ke  
<https://github.com/Isaakkamau>

## Anaconda

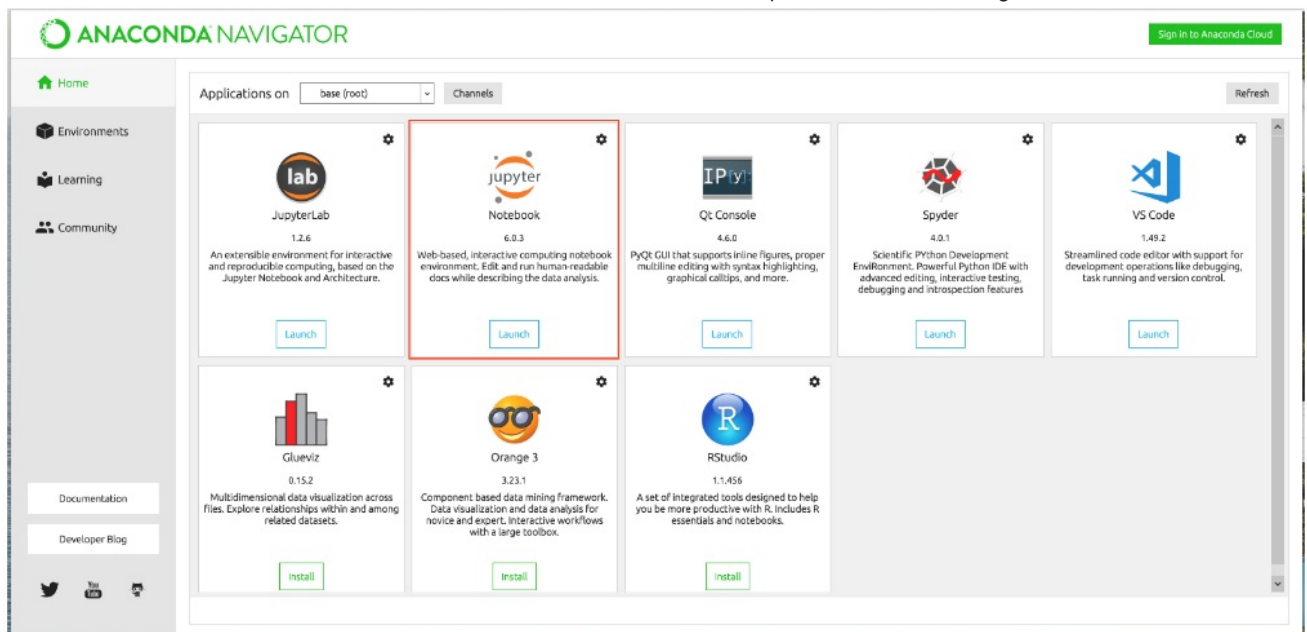
**Anaconda** is a program to manage (install, upgrade, or uninstall) packages and environments to use with Python. It's simple to install packages with Anaconda and create virtual environments to work on multiple projects conveniently.

Even if you already have Python installed, it will be beneficial to use Anaconda/Miniconda because:

1. Anaconda comes with a bunch of data science packages; you'll be all set to start working with data.
2. Using conda to manage your packages and environments will reduce future issues dealing with the various libraries you'll be using.

Anaconda is a software distribution that includes the following:

**Anaconda Navigator** - It is a graphical user interface that helps open up any installed applications, such as Jupyter notebook or VS code editor. We will learn more about the notebook in the next lesson. See a snapshot of Anaconda Navigator below:



## install packages

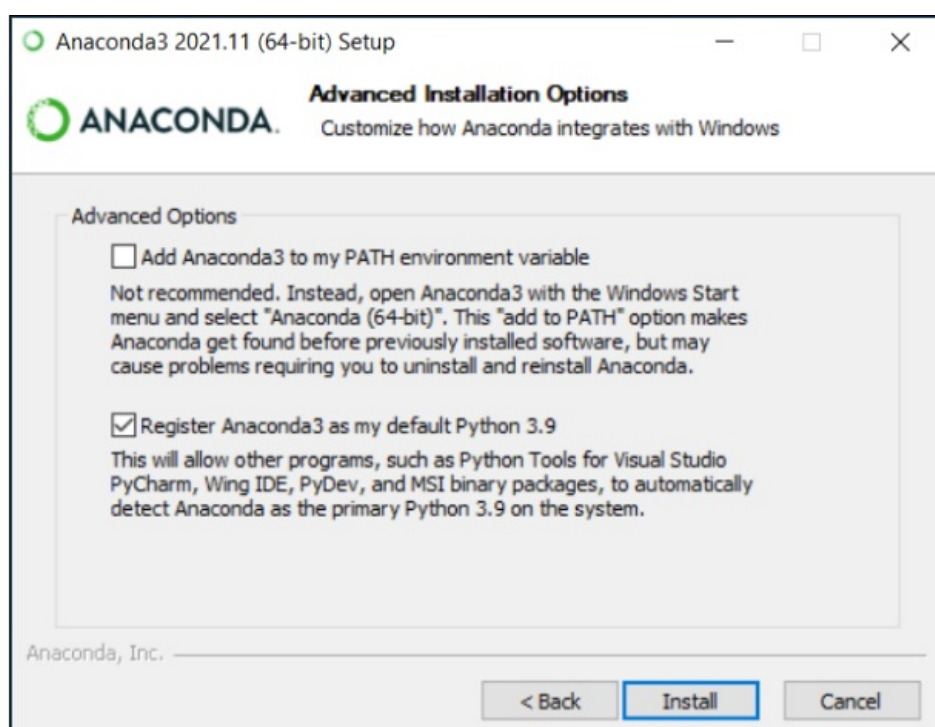
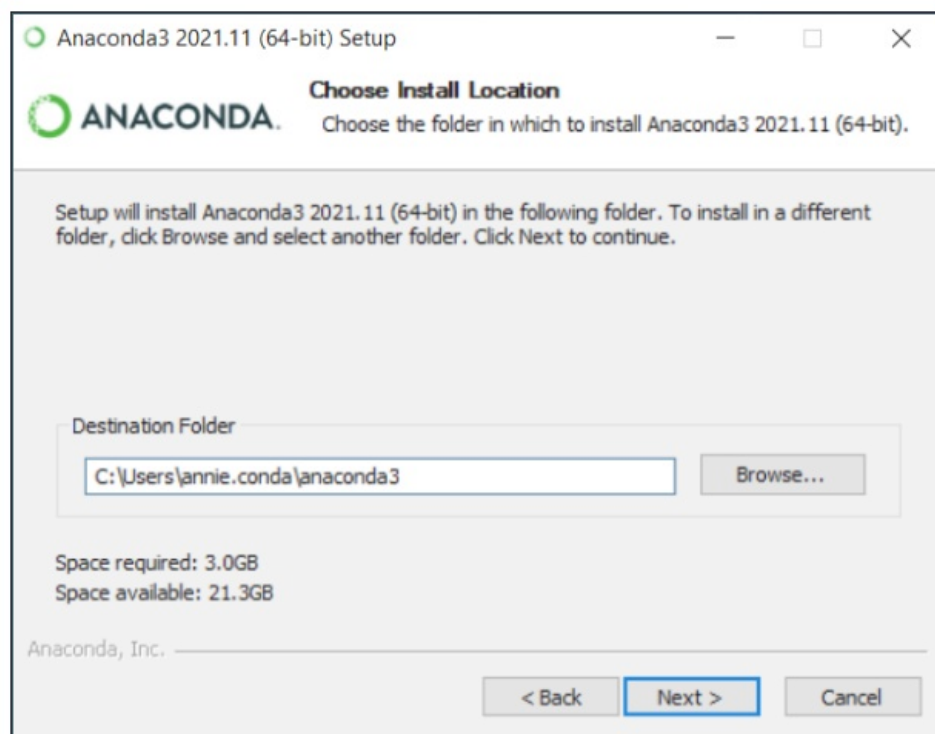
The **conda** and **pip** both are the Python package managers. Package managers are used to installing libraries and other software on your computer. **pip** is the default package manager for Python libraries, whereas **conda** focuses only on the packages that are available from the Anaconda distribution.

Pip can install any package listed on the Python Package Index (PyPI) <https://pypi.org/>.

## Installing Anaconda on Windows

For MacOS follow this link <https://docs.anaconda.com/anaconda/install/mac-os/> For Linux OS follow this link <https://docs.anaconda.com/anaconda/install/linux/>

1. Download the Anaconda installer.
2. Go to your Downloads folder and double-click the installer to launch. To prevent permission errors, do not launch the installer from the Favorites folder.  
  
Note, If you encounter issues during installation, temporarily disable your anti-virus software during install, then re-enable it after the installation concludes. If you installed for all users, uninstall Anaconda and re-install it for your user only.
3. Click Next.
4. Read the licensing terms and click I Agree.
5. It is recommended that you install for Just Me, which will install Anaconda Distribution to just the current user account. Only select an install for All Users if you need to install for all users' accounts on the computer (which requires Windows Administrator privileges).
6. Click Next.
7. Select a destination folder to install Anaconda and click Next. Install Anaconda to a directory path that does not contain spaces or unicode characters. For more information on destination folders, see the FAQ.





After installation, you're automatically in the default conda environment with all packages installed. You can check out your own install by entering the following command into your terminal.

## Starting the Jupyter Notebook Server

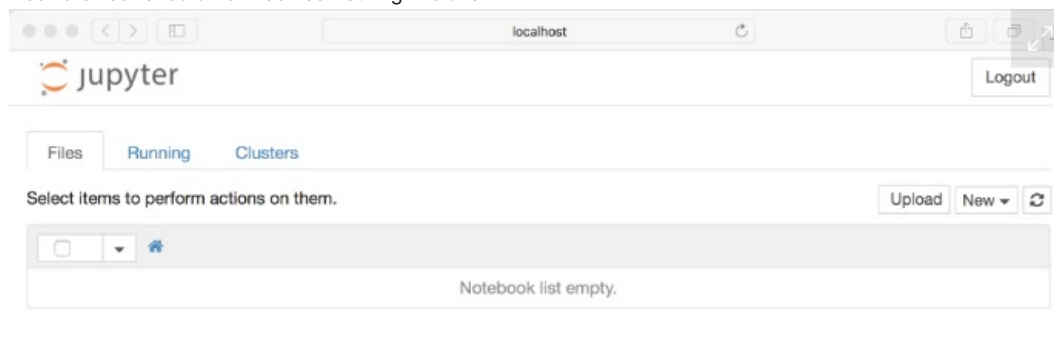
Now that you have Jupyter installed, let's learn how to use it. To get started, all you need to do is open up your terminal application and go to a folder of your choice. I recommend using something like your Documents folder to start out with and create a subfolder there called Notebooks or something else that is easy to remember.

Then just go to that location in your terminal and run the following command:

```
$ jupyter notebook
```

This will start up Jupyter and your default browser should start (or open a new tab) to the following URL: <http://localhost:8888/tree>

Your browser should now look something like this:



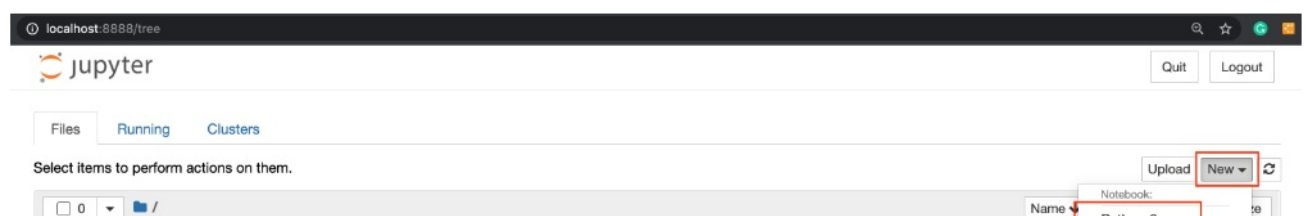
Note that right now you are not actually running a Notebook, but instead you are just running the Notebook server. Let's actually create a Notebook now!

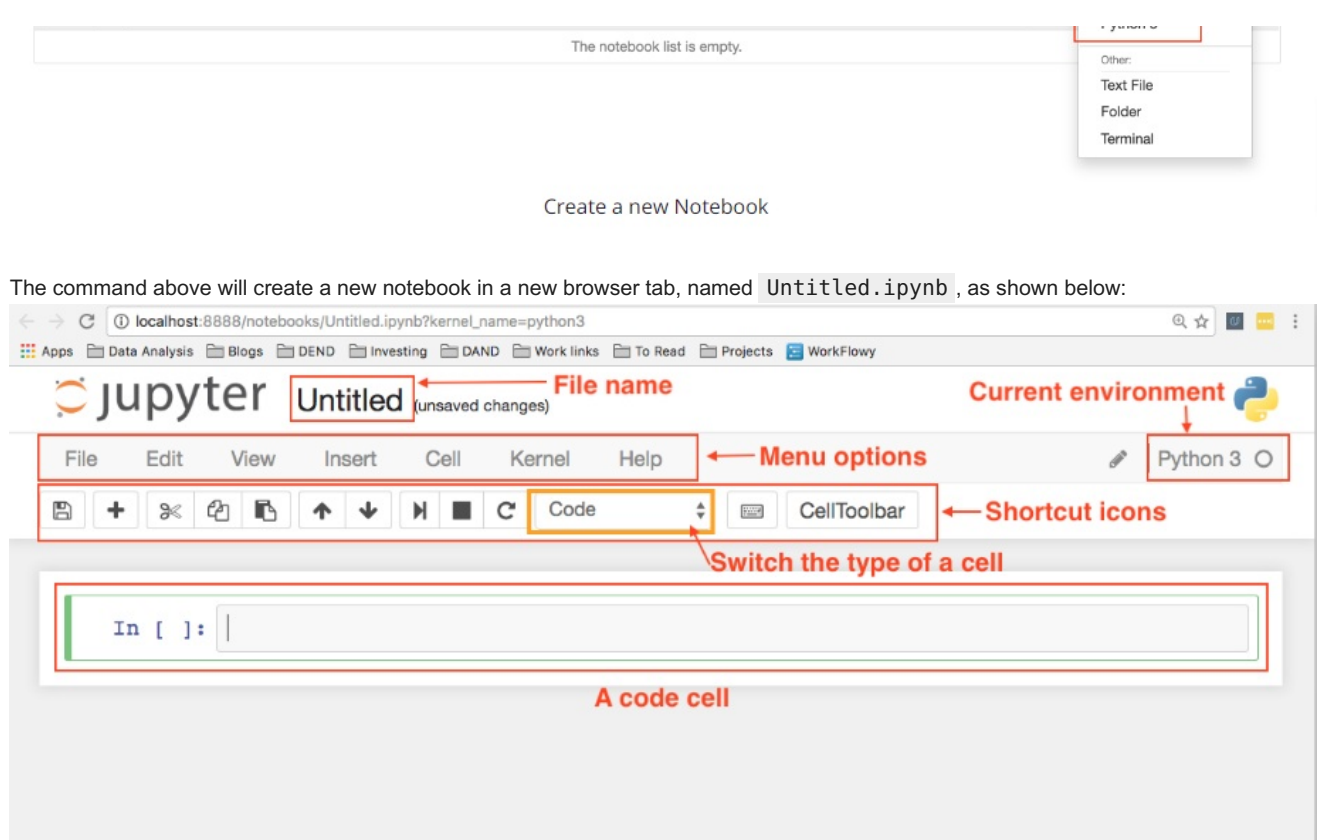
## creating a Notebook

Now that you know how to start a Notebook server, you should probably learn how to create an actual Notebook document.

All you need to do is click on the New button (upper right), and it will open up a list of choices. On my machine, I happen to have Python 2 and Python 3 installed, so I can create a Notebook that uses either of these. For simplicity's sake, let's choose Python 3.

Your web page should now look like this:

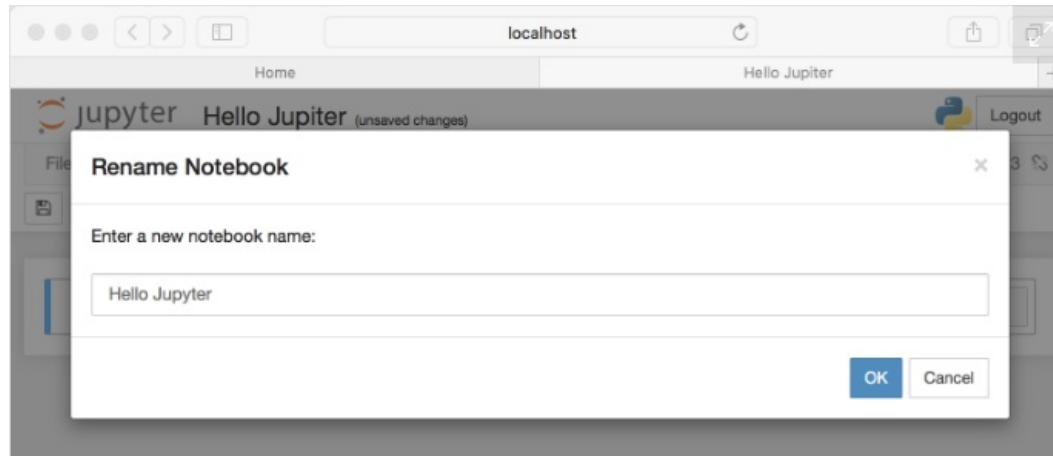




## Naming

You will notice that at the top of the page is the word `Untitled`. This is the title for the page and the name of your Notebook. Since that isn't a very descriptive name, let's change it!

Just move your mouse over the word `Untitled` and click on the text. You should now see an in-browser dialog titled `Rename Notebook`. Let's rename this one to `Hello Jupyter`:



```
In [ ]:
```

## Running Cells

A Notebook's cell defaults to using code whenever you first create one, and that cell uses the kernel that you chose when you started your Notebook.

In this case, you started yours with Python 3 as your kernel, so that means you can write Python code in your code cells. Since your initial Notebook has only one empty cell in it, the Notebook can't really do anything.

Thus, to verify that everything is working as it should, you can add some Python code to the cell and try running its contents.

Let's try adding the following code to that cell:

```
In [3]: print("Hello Jupyter")
```

Hello Jupyter

Running a cell means that you will execute the cell's contents. To execute a cell, you can just select the cell and click the Run button that is in the row of buttons along the top. It's towards the middle. If you prefer using your keyboard, you can just press `Shift+Enter`.

The Menus The Jupyter Notebook has several menus that you can use to interact with your Notebook. The menu runs along the top of the Notebook just like menus do in other applications. Here is a list of the current menus:

- File
- Edit
- View
- Insert
- Cell
- Kernel
- Widgets
- Help

Let's go over the menus one by one. This article won't go into detail for every single option in every menu, but it will focus on the items that are unique to the Notebook application.

The first menu is the File menu. In it, you can create a new Notebook or open a preexisting one. This is also where you would go to rename a Notebook. I think the most interesting menu item is the Save and Checkpoint option. This allows you to create checkpoints that you can roll back to if you need to.

Next is the Edit menu. Here you can cut, copy, and paste cells. This is also where you would go if you wanted to delete, split, or merge a cell. You can reorder cells here too.

Note that some of the items in this menu are greyed out. The reason for this is that they do not apply to the currently selected cell. For example, a code cell cannot have an image inserted into it, but a Markdown cell can. If you see a greyed out menu item, try changing the cell's type and see if the item becomes available to use.

The View menu is useful for toggling the visibility of the header and toolbar. You can also toggle Line Numbers within cells on or off. This is also where you would go if you want to mess about with the cell's toolbar.

The Insert menu is just for inserting cells above or below the currently selected cell.

The Cell menu allows you to run one cell, a group of cells, or all the cells. You can also go here to change a cell's type, although I personally find the toolbar to be more intuitive for that.

The other handy feature in this menu is the ability to clear a cell's output. If you are planning to share your Notebook with others, you will probably want to clear the output first so that the next person can run the cells themselves.

The Kernel cell is for working with the kernel that is running in the background. Here you can restart the kernel, reconnect to it, shut it down, or even change which kernel your Notebook is using.

You probably won't be working with the Kernel all that often, but there are times when you are debugging a Notebook that you will find you need to restart the Kernel. When that happens, this is where you would go.

The Widgets menu is for saving and clearing widget state. Widgets are basically JavaScript widgets that you can add to your cells to make dynamic content using Python (or another Kernel).

Finally you have the Help menu, which is where you go to learn about the Notebook's keyboard shortcuts, a user interface tour, and lots of reference material.

## Cell Types

There are technically four cell types: `Code` , `Markdown` , `Raw NBConvert` , and `Heading` .

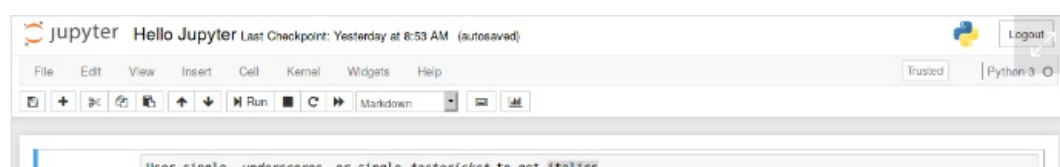
The Heading cell type is no longer supported and will display a dialog that says as much. Instead, you are supposed to use Markdown for your Headings.

The Raw NBConvert cell type is only intended for special use cases when using the nbconvert command line tool. Basically it allows you to control the formatting in a very specific way when converting from a Notebook to another format.

## Styling Your Text

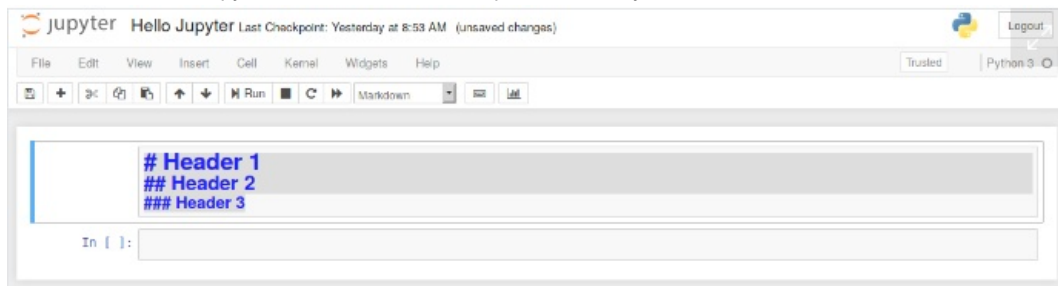
Jupyter Notebook supports Markdown, which is a markup language that is a superset of HTML. This tutorial will cover some of the basics of what you can do with Markdown.

Set a new cell to Markdown and then add the following text to the cell:



## Headers

Creating headers in Markdown is also quite simple. You just have to use the humble pound sign. The more pound signs you use, the smaller the header. Jupyter Notebook even kind of previews it for you:



## Emphasis

Use the following code to emphasize text: Bold text: `__string__` or `**string**` Italic text: `_string_` or `*string*`

## Mathematical symbols

Surround mathematical symbols with a dollar sign (`$`), *for example*: ``5x + 3y``

```
$$  
y = \frac{a}{b+c}  
$$
```

$$y = \frac{a}{b+c}$$

## Indenting

Use the greater than sign (`>`) followed by a space, for example: `> Text that will be indented when the Markdown is rendered.`

## Numbered lists

To create a numbered list, enter 1. followed by a space, for example:

1. Numbered item
2. Numbered item

## Colored note boxes

Use one of the following

tags to display text in a colored box.

```
<div class="alert alert-block alert-info">  
<b>Tip:</b> Use blue boxes (alert-info) for tips and notes.  
If it's a note, you don't have to include the word "Note".  
</div>
```

The color of the box is determined by the alert type that you specify: `Blue boxes (alert-info)`

**Tip:** Use blue boxes (alert-info) for tips and notes. If it's a note, you don't have to include the word "Note".

`Yellow boxes (alert-warning)`

```
<div class="alert alert-block alert-warning">  
<b>Example:</b> Use yellow boxes for examples that are not  
inside code cells, or use for mathematical formulas if needed.  
</div>
```

**Example:** Use yellow boxes for examples that are not inside code cells, or use for mathematical formulas if needed.

#### Green boxes (alert-success)

```
<div class="alert alert-block alert-success">
<b>Up to you:</b> Use green boxes sparingly, and only for some specific
purpose that the other boxes can't cover. For example, if you have a lot
of related content to link to, maybe you decide to use green boxes for
related links from each section of a notebook.
</div>
```

**Up to you:** Use green boxes sparingly, and only for some specific purpose that the other boxes can't cover. For example, if you have a lot of related content to link to, maybe you decide to use green boxes for related links from each section of a notebook.

#### Red boxes (alert-danger)

```
<div class="alert alert-block alert-danger">
<b>Just don't:</b> In general, avoid the red boxes. These should only be
used for actions that might cause data loss or another major issue.
</div>
```

#### Red boxes (alert-danger)

**Just don't:** In general, avoid the red boxes. These should only be used for actions that might cause data loss or another major issue.

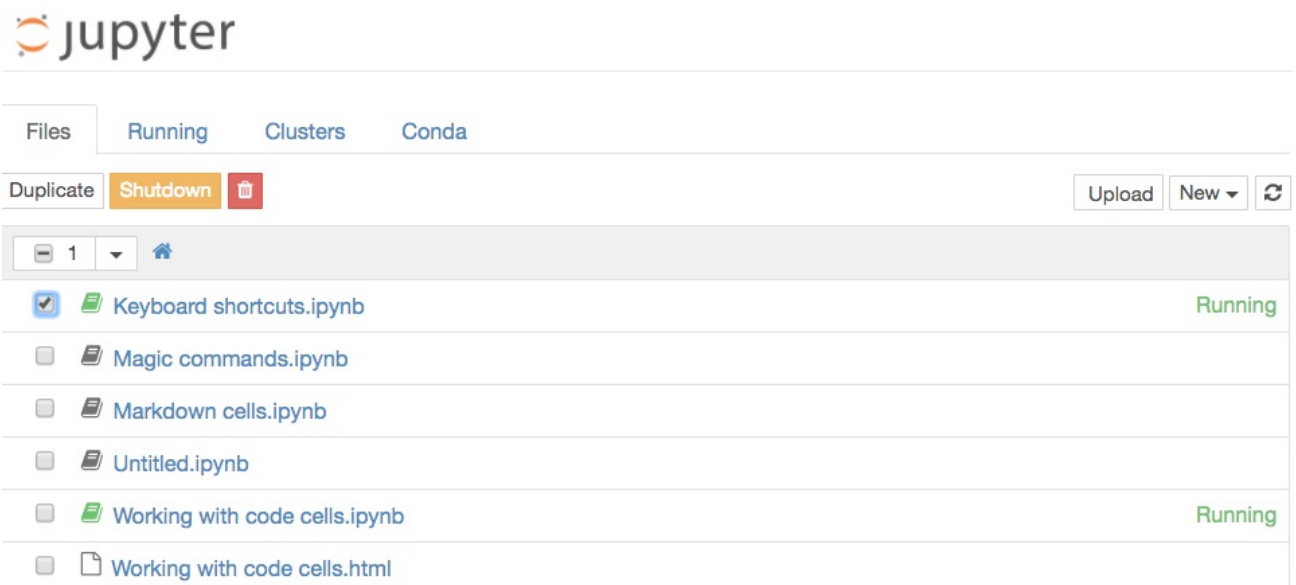
## External links

To link to an external site, use the following code:

```
__[link text](http://url)__
```

## Shutting down Jupyter:

You can shutdown individual notebooks by marking the checkbox next to the notebook on the server home and clicking "Shutdown." Make sure you've saved your work before you do this though! Any changes since the last time you saved will be lost. You'll also need to rerun the code the next time you run the notebook.



The screenshot shows the JupyterLab interface. At the top, there are tabs for 'Files', 'Running', 'Clusters', and 'Conda'. Below these tabs, there are buttons for 'Duplicate', 'Shutdown', and a trash icon. On the right side, there are buttons for 'Upload', 'New', and a refresh icon. The main area displays a list of notebooks and files. The first row shows 'Keyboard shortcuts.ipynb' with a checkbox checked and the status 'Running'. The second row shows 'Magic commands.ipynb' with a checkbox unchecked. The third row shows 'Markdown cells.ipynb' with a checkbox unchecked. The fourth row shows 'Untitled.ipynb' with a checkbox unchecked. The fifth row shows 'Working with code cells.ipynb' with a checkbox unchecked and the status 'Running'. The sixth row shows 'Working with code cells.html' with a checkbox unchecked.

You can shutdown the entire server by pressing `control + C` twice in the terminal. Again, this will immediately shutdown all the running notebooks, so make sure your work is saved!

Processing math: 100%