

# Configuring the environment

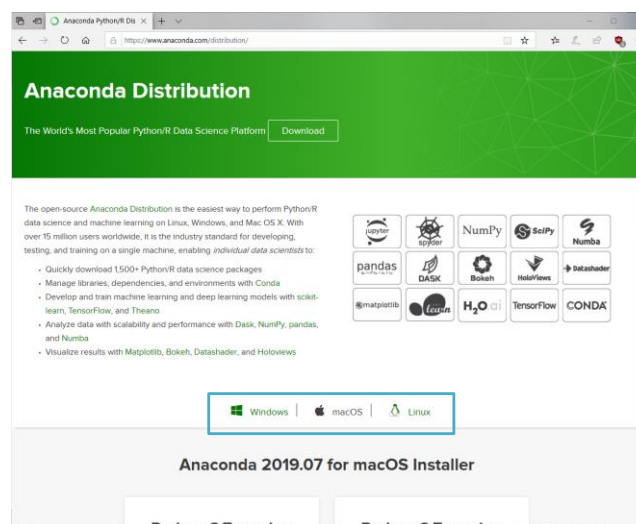
## Introduction

Throughout this module you will use Anaconda combined with Jupyter notebook. This tutorial will describe how to download and install both applications.

## Install Anaconda

Browse to [this link](#) to initialise the download of the Anaconda distribution. The page layout may have changed since these notes were written, but the general links for installation of Anaconda for your OS will be available on this page.

In the centre of this page are links to download Anaconda for your OS. Find the OS link that matches your computer and click on it.

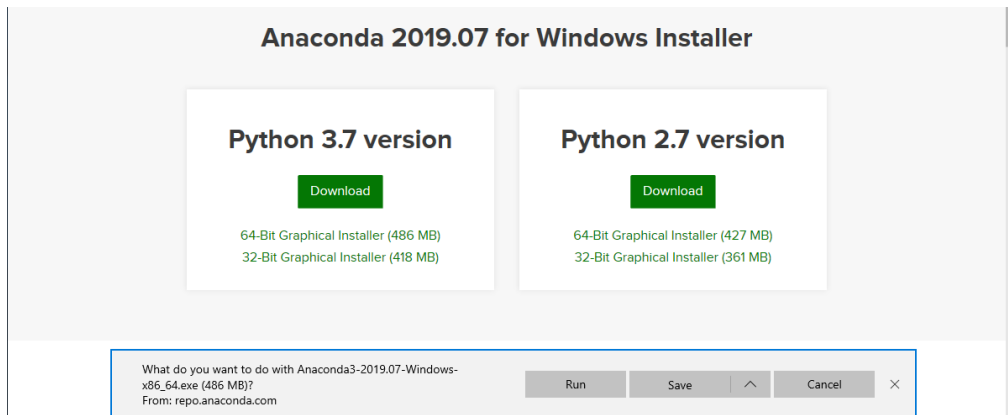


For example, in this document I've selected the **Windows** download option. These install note instructions are the same for all OS.

Scroll down the page to view the installer options for your selected OS. Choose **Python 3.7** version. For a windows installation, be careful to choose the correct graphical installer that is suitable for your computer as there are 64-bit and 32-bit options available to you.

If you click the **green download button**, make sure that the filename that will be downloaded to your computer matches with your OS. For example, the file shown in the screen grab below indicates that the file to be download is for a 64-bit windows as indicated by the name "windows-x86\_64".

## Configuring the environment



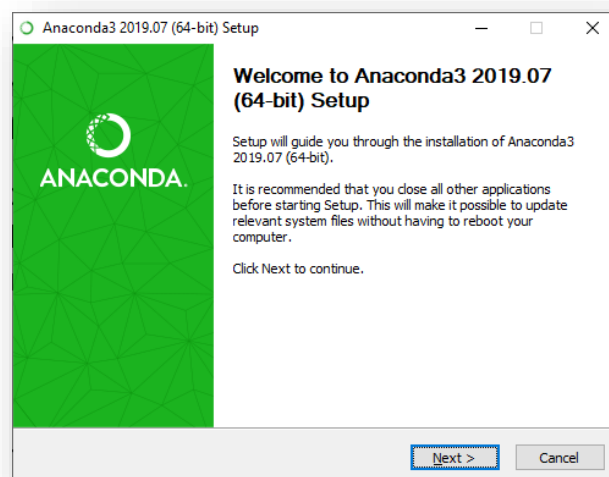
It is very likely that your computer has a 64-bit architecture, so it should be okay to click on the green download button to initialise the download process.

Choose **Run** if your browser gives you an option to do so. If not, download the installer file to your computer and then run it. On an Apple Mac, click on the dmg file in your Downloads folder once the installer file has downloaded it.

Once download, you can refer to the installation instructions at the bottom of the screen, or click on [this link](#). The install instructions are also explained in the following steps. Please note that the install screenshots may be slightly different to the ones presented to you.

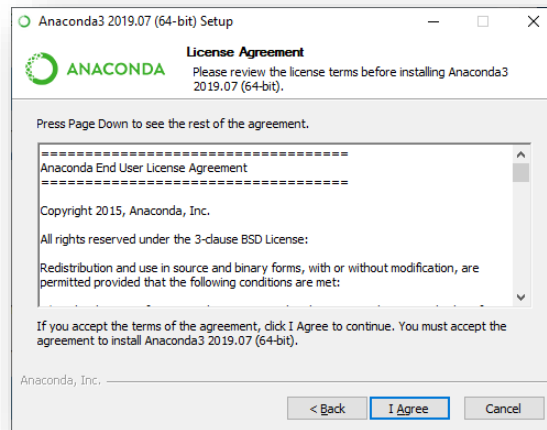
### Install on Windows

If you didn't decide to run the installer when downloading it, browse to it and double-click it to begin the install process. Click **Next**.

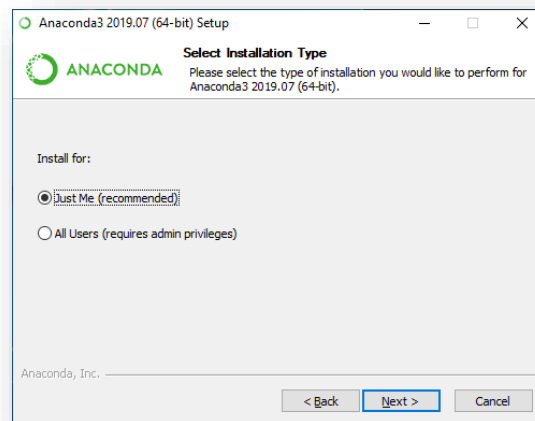


Click **I agree** at the License agreement screen.

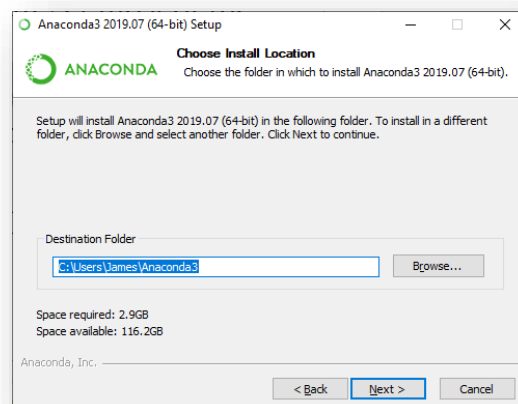
## Configuring the environment



Choose **Just me** at the installation Type screen. Click **Next**.



Select a destination folder to install Anaconda and click **Next**. There's no need to change the destination folder.



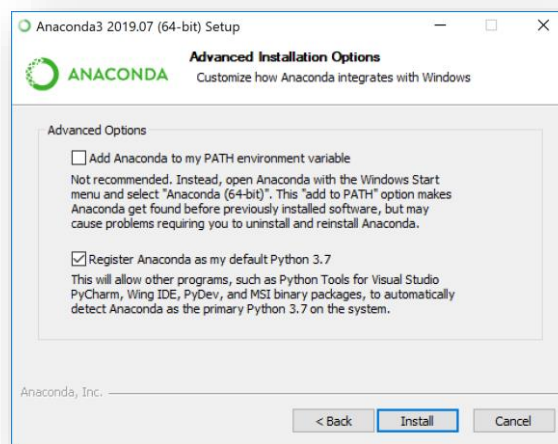
## Configuring the environment

Choose whether to add Anaconda to your PATH environment variable.

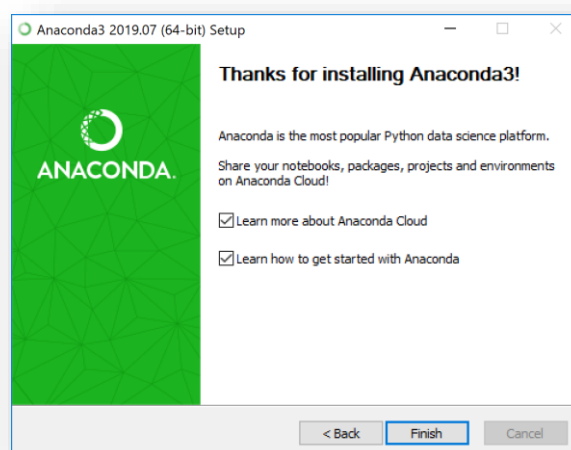
**Please note** - If you've **previously installed** another version of Python onto your computer then I recommend that you choose the option to **Add Anaconda to my PATH environment variable**. Otherwise confusion can arise when you go to use Python program code. If this is the **first time you've installed** Python onto your computer, then I recommend not adding Anaconda to the PATH environment variable, since this can interfere with other software.

Choose **Register Anaconda as my default Python 3.7**.

Click **Install** to begin the installation process.



Once installed, you'll see a thanks message from Anaconda. Click Finish to close it.



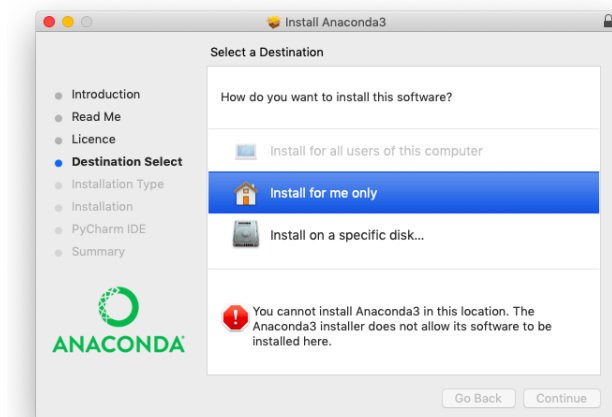
## Configuring the environment

### Installing Anaconda on MacOS

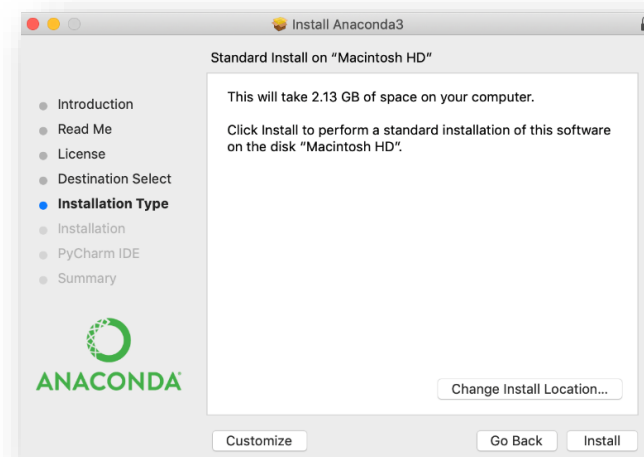
Browse to the location of the installer you downloaded and double-click on it to begin the install process.

Answer the prompts by clicking **Continue** on the Introduction, Read Me, and License screens.

Choose **Install for me only** at the **Select a Destination** screen. Click **Continue**.

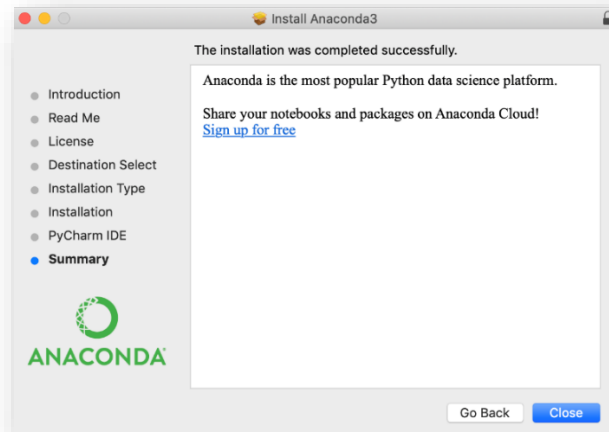


Click the **Install** button to install Anaconda in your home user directory (recommended)



Once installation is complete, Click **Close** to finish the installation process.

## Configuring the environment

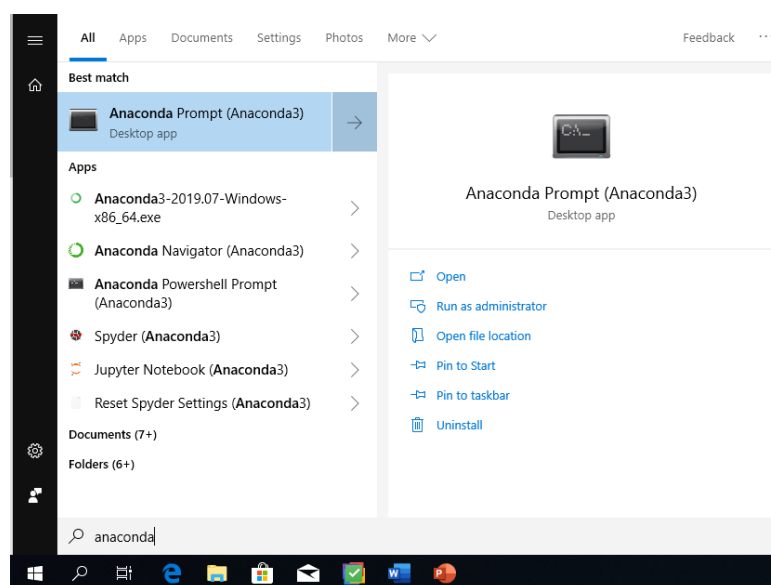


### Configuring the Anaconda environment

We'll use several Python libraries throughout this module. In this step we'll create and activate a virtual environment that has been set up for you. This environment contains all of the libraries you'll need for this module, so you will not need to install the libraries manually at a later date. Before you begin this step, download the **nlp\_environment.yml** file from the **week1** folder on Blackboard and save it to a location on your local computer. In this example, I've copied the yml file to the top directory of local hard drive of my windows computer. Take note of the location where you've downloaded this file.

You'll need to use the Anaconda command prompt to run the yml file and to configure the virtual environment. Please note that you **must not use** the standard command prompt to do this.

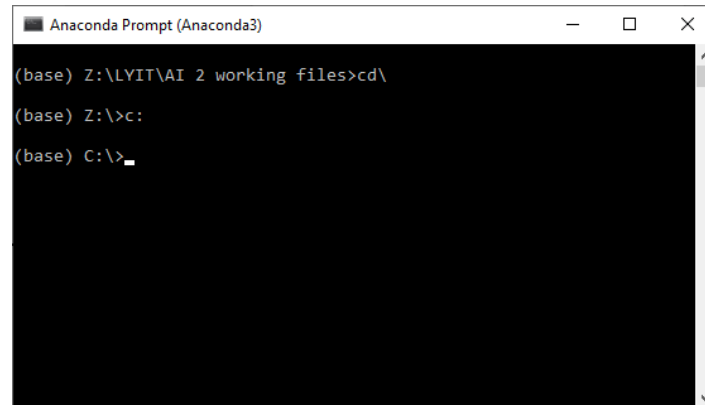
In the windows environment, search for **Anaconda Prompt** from the start menu.



## Configuring the environment

On the MacOS environment, you will need to use the terminal application.

Within the Anaconda prompt, browse to the location where the yml file is stored. In the example below, I changed directory from the Z: drive to the local c: drive. The C: drive contains the yml file that I want to execute.

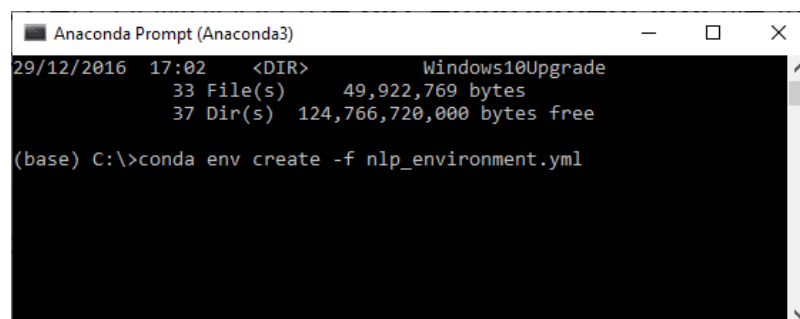


```
Anaconda Prompt (Anaconda3)
(base) Z:\LYIT\AI 2 working files>cd\
(base) Z:\>c:
(base) C:\>_
```

Type **dir** (in windows) or **ls** (in MacOS) to ensure the yml file is in the folder. If the yml file is not available, check that you have the yml file in the same location as the folder you've browsed to.

Type in the following command into the command prompt window. This command is the same for windows and MacOS

**Conda env create -f nlp\_environment.yml**



```
Anaconda Prompt (Anaconda3)
29/12/2016 17:02 <DIR> Windows10Upgrade
33 File(s) 49,922,769 bytes
37 Dir(s) 124,766,720,000 bytes free
(base) C:\>conda env create -f nlp_environment.yml
```

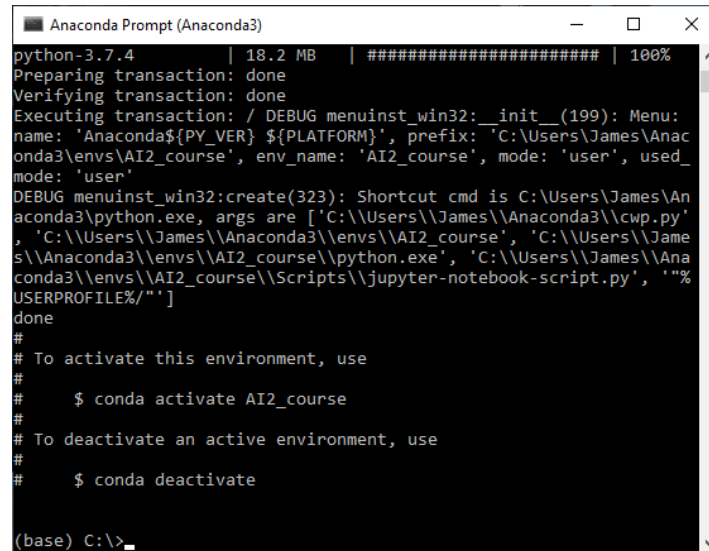
Press **Return** to run the command.

If you receive an error message when running this command, double-check that you've keyed in the correct filename for the environment file (yml file). And ensure you are in the same directory as the location where the yml file is stored.

You may receive an update warning if you've previously installed Anaconda and a newer version of the software is available. Follow the instructions within the command prompt window to update Anaconda to the most recent version once all of the packages have installed.

## Configuring the environment

If you've executed the command correctly, several packages will automatically download and install onto your computer. The process will take a few minutes to complete. Once finished, your command prompt should look like this.



```

Anaconda Prompt (Anaconda3)
python-3.7.4 | 18.2 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: / DEBUG menuinst_win32: __init__(199): Menu:
name: 'Anaconda${PY_VER} ${PLATFORM}', prefix: 'C:\Users\James\Anaconda3\envs\AI2_course', env_name: 'AI2_course', mode: 'user', used_mode: 'user'
DEBUG menuinst_win32:create(323): Shortcut cmd is C:\Users\James\Anaconda3\python.exe, args are ['C:\Users\James\Anaconda3\cwp.py', 'C:\Users\James\Anaconda3\envs\AI2_course', 'C:\Users\James\Anaconda3\envs\AI2_course\python.exe', 'C:\Users\James\Anaconda3\envs\AI2_course\Scripts\jupyter-notebook-script.py', "%USERPROFILE%\"]
done
#
# To activate this environment, use
#
#     $ conda activate AI2_course
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#
(base) C:\>_
  
```

### Activate the AI 2 virtual environment

You can activate the AI 2 virtual environment once all the packages are correctly installed. Type in the following command to activate the newly installed environment for windows and MacOS

**conda activate AI2\_course**

When finished with your NLP session, you can deactivate the environment by typing in

**conda deactivate**

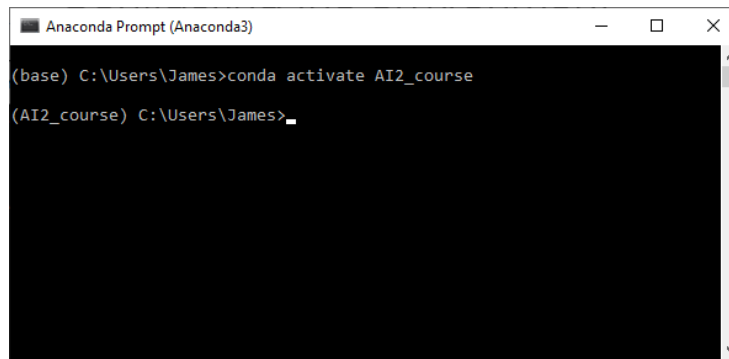
The command window will also display the commands you'll need to activate and deactivate the environment.

You will need to use these commands each time you wish to use the AI2 virtual environment.

You can quickly check and confirm that you are in the new environment by checking the command line looks similar to the screenshot below. Notice that the AI2\_course environment name is shown inside brackets at the beginning of the command line.



## Configuring the environment



```
Anaconda Prompt (Anaconda3)
(base) C:\Users\James>conda activate AI2_course
(AI2_course) C:\Users\James>_
```

### Jupyter notebook

We'll use Jupyter notebook throughout the NLP section of the module. Jupyter notebook is an application that provides a facility to combine both text and computer code in one interactive document. The software runs on either your local computer, or on a remote server. We'll use Jupyter notebook on the local computer.

A notebook is linked to a particular kernel that is used to execute any code presented within a particular notebook. For example, in this module we will use the **ipython kernel** to execute Python code presented in any notebook. There are several other kernels available such as IRkernel (R language) and IJulia (Julia language). Read [this link](#) for more information on Jupyter kernels, and [this link](#) from GitHub if you are interested in using Jupyter with another programming language. And check out [this link](#) for more information on the Jupyter Notebook application.

### Starting a Jupyter Notebook in a specific folder

Before we begin a new Jupyter notebook session, first we must browse to the folder where we'll store our notebook files. In this example, we are going to use a folder on the local computer. **I recommend you should create a folder in your cloud environment such as OneDrive and store your work there.**

On windows, create a folder inside the **Documents** folder on my computer called **AI2**. Note that I didn't put a space in the name of this folder.

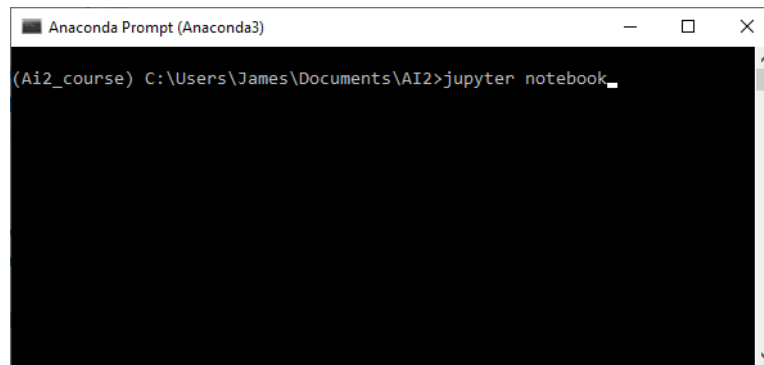
On MacOS, create a folder called **AI2** in the documents folder using **Finder**.

Browse to the AI2 folder using the Anaconda command prompt.

Using the Anaconda command prompt, start a new Jupyter notebook session by typing in the following command

**jupyter notebook**

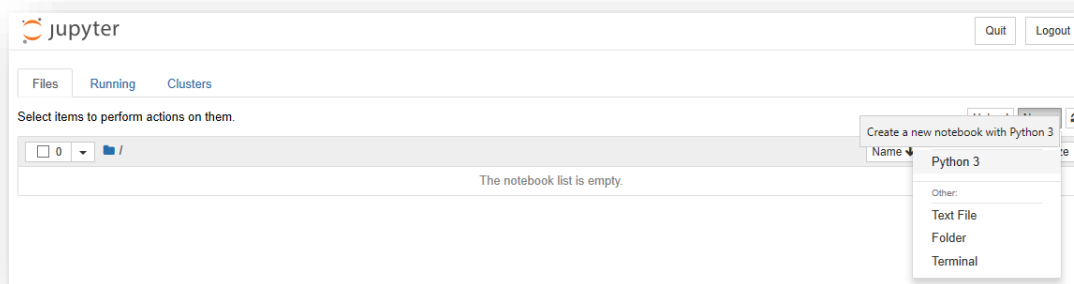
## Configuring the environment



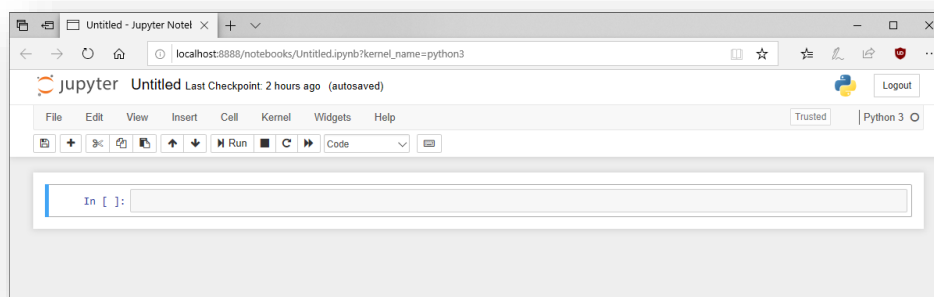
This command initialises a local Jupyter notebook server instance and opens a new notebook in your default web browser.

The default directory for this new notebook is inside the **AI2** folder that you browsed to earlier through the command prompt. If you didn't create a folder, the default directory will be the current folder that your command prompt is showing. This will probably be your user folder on your local computer.

To begin a new Python notebook, open your new jupyter notebook page in your browser, click on the **New** button at the top right of the jupyter notebook page in your browser. Choose **Python 3**.



Once the jupyter server is initialised, your web browser should now show a screen like this. You are not running a Notebook at that stage, but instead you are just running a Notebook server.



# Configuring the environment

## Hello World

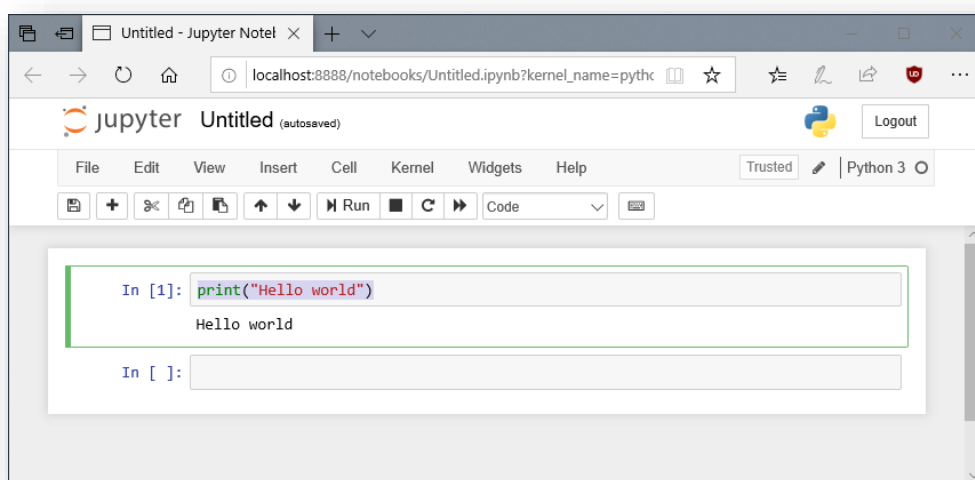
Let's enter some code into a cell.

Click in the first cell and type in this Python code.

```
print("Hello world")
```

Then press **Shift + Enter** to run this program code. The cell is not like a Python interpreter where the code executes immediately. We need to run the code too, so we need to use these keys. Alternatively, click on the **Run** button on the menu bar at the top of the jupyter notebook screen.

Your screen should now look like this.



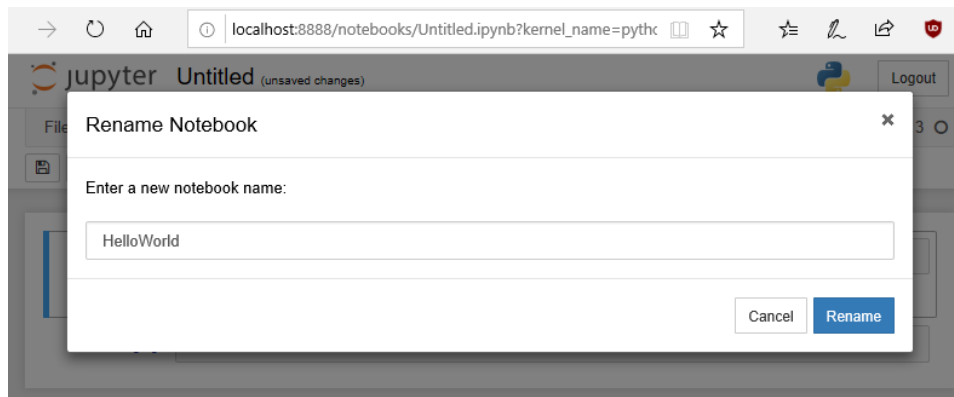
Congratulations! You've just executed your first line of Python code within a new Jupyter notebook. For more information on notebook basics, click [this link](#).

## Saving your notebook

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.

Move your mouse pointer 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 **HelloWorld**.

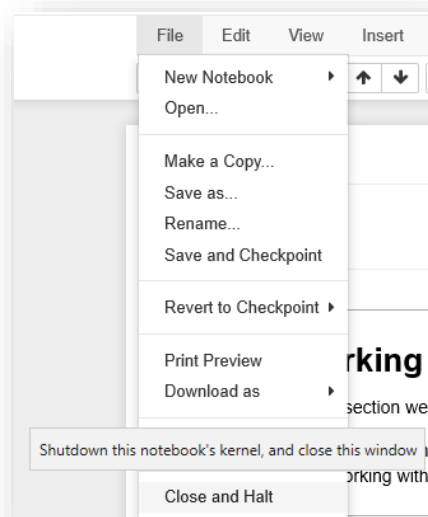
## Configuring the environment



Click **Rename**. You've now renamed your notebook to **HelloWorld** and in the background, your notebook has been saved as **HelloWorld.ipynb** which is located inside the AI2 folder of your local hard drive.

When you run a cell with **Shift + Enter**, you will notice that there are some square brackets next to the word **In** to the left of the cell. The square brackets will auto fill with a number that indicates the order that you ran the cells. For example, if you open a fresh Notebook and run the first cell at the top of the Notebook, the square brackets will fill with the number 1. Read this [link](#) for more information on various options within a notebook.

You can close your notebook by either closing the browser window containing the jupyter page, or by choosing **File** → **Close and halt** from the toolbar at the top of the Jupyter notebook screen.



You can easily reopen a jupyter notebook file by opening the Anaconda prompt and then launching jupyter notebook. Then navigate to the notebook

## Configuring the environment

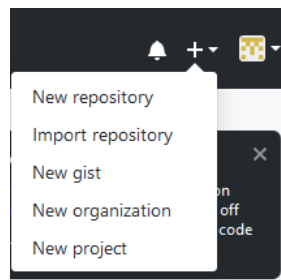
file you wish to open and click its filename. A jupyter notebook will have a file extension of .ipnyb.

### Saving your work to GitHub

In this section I'm assuming you have already set up a GitHub account. If you haven't done so, refer to the notes on Blackboard on how to register for a free student GitHub account and then continue at this point.

Log into your GitHub account.

Create a new repository (repo) to store your jupyter notebooks in. Click on the **+** icon to the top right of the GitHub page and choose **New repository**.



Call the new repo **AI2HelloWorld**. Enter a description if you want.

Choose **Public** as the repository type.

Check **Initialise this repository with a README**.

Click **Create repository**.

**Create a new repository**  
 A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

---

Owner: Jamesconnollyyt / Repository name: AI2HelloWorld ✓

Great repository names are short and memorable. Need inspiration? How about [reimagined-doodle?](#)

Description (optional)

---

☒ **Public**  
 Anyone can see this repository. You choose who can commit.

☐ **Private**  
 You choose who can see and commit to this repository.

---

Skip this step if you're importing an existing repository.

☒ **Initialize this repository with a README**  
 This will let you immediately clone the repository to your computer.

Add .gitignore: None Add a license: None ⓘ

---

**Create repository**


Once created, your new repo should be available on-screen.

## Configuring the environment

Click on **upload files**.

Drag the **HelloWorld** jupyter notebook file you created into the appropriate location on-screen.

AI2HelloWorld /



**Drag files here to add them to your repository**

[Or choose your files](#)

**Commit changes**

Add files via upload

Add an optional extended description...

☒ Commit directly to the `master` branch.
 ☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes
Cancel

Once you've dragged the file you want to upload to GitHub, click on the **Commit changes** button at the bottom of the screen to commit your new file to the **AI2HelloWorld** repo.

Once uploaded, your new **HelloWorld.ipynb** should be available within the **AI2HelloWorld** repo.

Branch: master ▾ New pull request

Create new file
Upload files
Find file
Clone or download ▾

jamesconnollyyit Add files via upload		Latest commit b20d8e6 now
HelloWorld.ipynb	Add files via upload	now
README.md	Initial commit	4 minutes ago

README.md

AI2HelloWorld

## Sharing your GitHub file

Open the repo containing the file you wish to share. Click on the file you want to share to open it. Copy the URL of the file once it is opened. Then use this link to share the contents of this file in your GitHub repo. For example, the link to my HelloWorld file is

<https://github.com/jamesconnollyyit/AI2HelloWorld/blob/master/HelloWorld.ipynb>