

# Setting up a working environment

## Machine Learning in Finance for Python (ECON5130)

Richard Foltyn

University of Glasgow







May 20, 2023

This guide gives you some hints on how to set up a working Python environment

- 1 Python environments
- 2 Running in the cloud
  - Running in Google Colab
  - Running in Binder ([mybinder.org](https://mybinder.org))
- 3 Running locally on your computer
  - Running in the browser with Jupyter lite
  - Installing a local Python environment

# Python environments — Feature matrix

There are multiple ways to run the course material:

	Local install  ANACONDA	Jupyter Lite  try  lite now	Binder  launch  binder	Google Colab  Open in Colab
Runs on <b>your</b> computer	✓	✓		
Runs in the cloud			✓	✓
Requires installation	✓			
Runs in browser	✓	✓	✓	✓
Other editors (PyCharm, VS Code)	✓			
Supports all packages	✓		✓	
Account/login required				✓
Saves changes between sessions	✓	✓		✓

# RUNNING IN THE CLOUD

# Running in the cloud


The course is based on interactive notebooks which you can run directly in your browser (with some limitations)

## Two cloud-based options:

- 1 Google Colab (<https://colab.research.google.com>)
- 2 Binder (<https://mybinder.org/>)

# Running notebooks in Google Colab

## How to run a notebook in Google Colab:



- 1 Go to <https://github.com/richardfoltyn/MLFP-ECON5130>
- 2 Click on the button  next to a notebook you want to run (shown on the next slide)

## Limitations:

- The environment is somewhat restricted (not trivial to import custom modules or local data files)
- It is possible to access data stored in Google Drive (but details go beyond this tutorial)
- Requires Google login to run anything

# Running notebooks in Google Colab

Launching notebooks from GitHub repository

 README.md 



## Machine Learning in Finance with Python - Part 1

License [CC BY-NC-SA 4.0](#)  [try](#) [lite now](#)  [launch](#) [binder](#)

Author: Richard Foltyn, University of Glasgow


### Units

This introductory course consists of several units. Each unit corresponds to one interactive Jupyter notebook, which is also available as a static PDF file. Alternatively, you can download the entire course as a [single PDF](#).

Unit	Title	PDF	Google Colab
1	Language and NumPy basics	<a href="#">PDF</a>	 <a href="#">Open in Colab</a>
2	Control flow and list comprehensions	<a href="#">PDF</a>	 <a href="#">Open in Colab</a>
3	Reusing code - Functions, modules and packages	<a href="#">PDF</a>	 <a href="#">Open in Colab</a>

# Running in Binder (mybinder.org)

## How to run a notebook in Binder:

- 1 Go to <https://github.com/richardfoltyn/MLFP-ECON5130>
- 2 Click on the button  (shown on the next slide)
- 3 Once the Binder instance is started, it will take you to an overview page from which you can access all notebooks



## Limitations:

- Binder **does not** save your changes between sessions: you need to download any modified notebooks and upload them later (see instructions below)
- Can take a long time to start
- **Note:** mybinder.org has become very unreliable (as of May 2023) and may not work at all



# Running notebooks in Binder


Launching notebooks from GitHub repository

 README.md 


## Machine Learning in Finance with Python - Part 1

License

CC BY-NC-SA 4.0

 try

lite now




 launch

binder

Author: Richard Foltyn, University of Glasgow

### Units

This introductory course consists of several units. Each unit corresponds to one interactive Jupyter notebook, which is also available as a static PDF file. Alternatively, you can download the entire course as a [single PDF](#).

Unit	Title	PDF	Google Colab
1	Language and NumPy basics	<a href="#">PDF</a>	 <a href="#">Open in Colab</a>
2	Control flow and list comprehensions	<a href="#">PDF</a>	 <a href="#">Open in Colab</a>
3	Reusing code - Functions, modules and packages	<a href="#">PDF</a>	 <a href="#">Open in Colab</a>

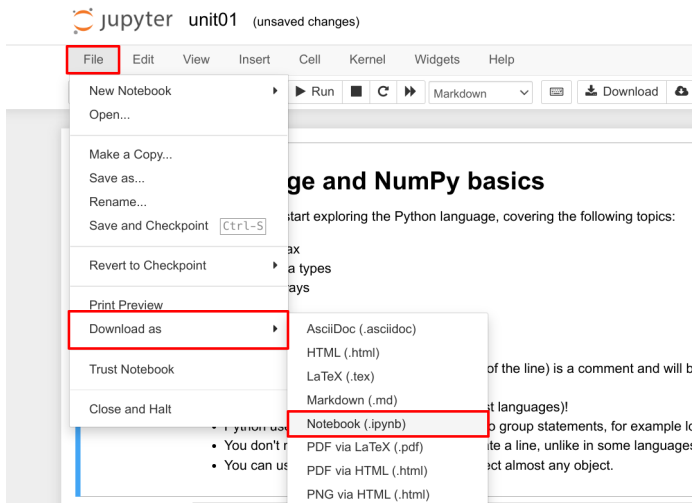
# Saving and opening notebooks

- **Important:** Notebooks launched in binder are not saved and will eventually disappear
- You need to manually save and restore notebooks:
  - 1 Before you stop working on a notebook, make sure to download it to your computer
  - 2 When you want to continue working on an existing notebook, upload it to binder first and then open it

See following slides for instructions!

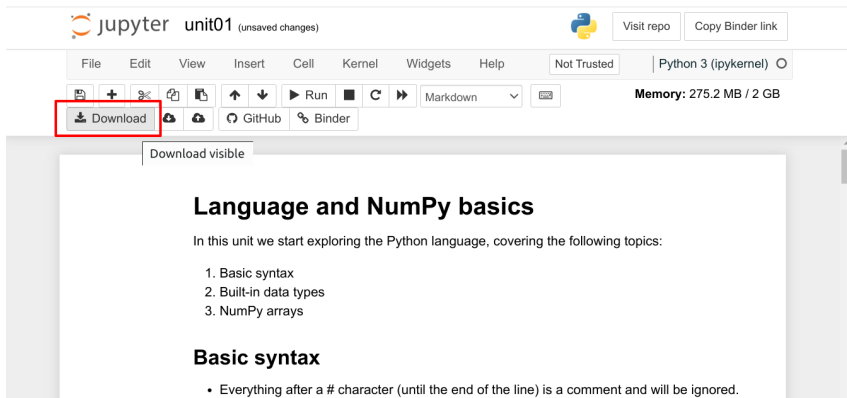
# Saving (downloading) notebooks

Select **File** ► **Download as** ► **Notebook (.ipynb)**



# Saving (downloading) notebooks

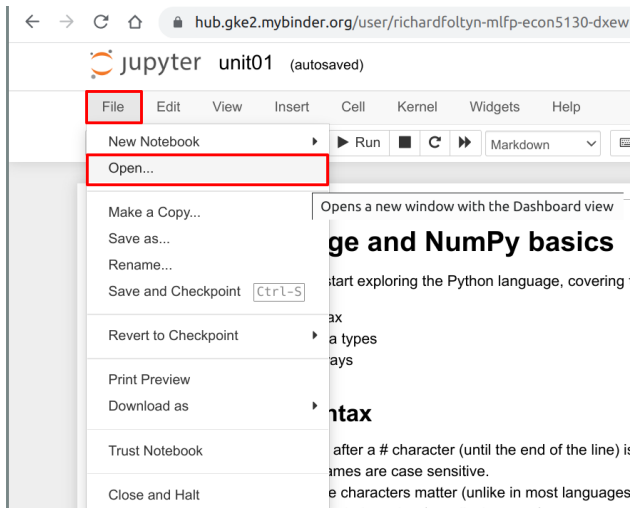
Alternatively, you can click on **Download** in the tool bar.



The screenshot shows the JupyterLab interface. At the top, the header includes the Jupyter logo, the text 'jupyter unit01 (unsaved changes)', a Python logo, and buttons for 'Visit repo' and 'Copy Binder link'. Below the header is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. To the right of the menu bar are 'Not Trusted' and 'Python 3 (ipykernel)' status indicators. The toolbar contains various icons for file operations, including a 'Download' icon (a downward arrow) which is highlighted with a red rectangle. Other icons include 'Run', 'Clear', 'Undo', 'Redo', 'Markdown', 'GitHub', and 'Binder'. The main content area displays a notebook titled 'Language and NumPy basics'. The notebook content includes a paragraph: 'In this unit we start exploring the Python language, covering the following topics:', followed by a numbered list: '1. Basic syntax', '2. Built-in data types', and '3. NumPy arrays'. Below the list is a section titled 'Basic syntax' with a bullet point: '• Everything after a # character (until the end of the line) is a comment and will be ignored.'

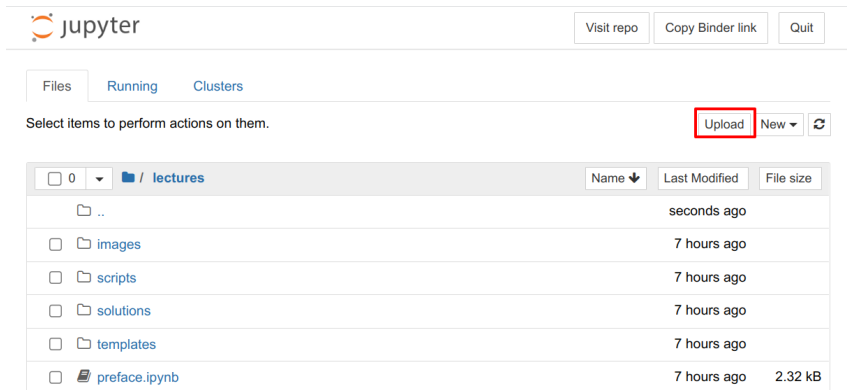
# Opening (uploading) notebooks: step 1

Select **File** ► **Open**. This brings up a file explorer page (see next slide)



# Opening (uploading) notebooks: step 2

Click on **Upload**. You will be prompted to select a file on your computer.

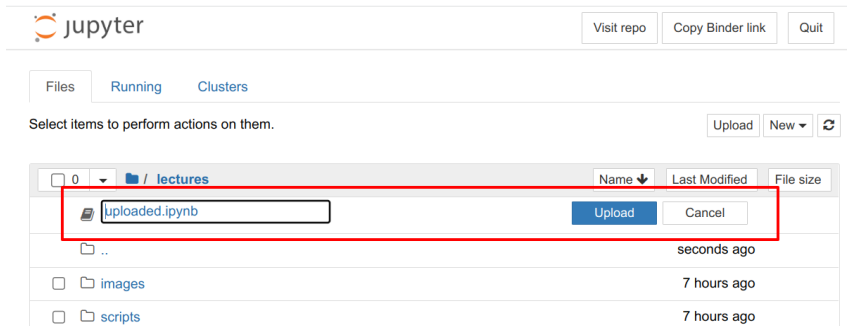


The image shows the JupyterLab web interface. At the top, there's a header with the Jupyter logo and the word "jupyter". To the right of the header are three buttons: "Visit repo", "Copy Binder link", and "Quit". Below the header, there are three tabs: "Files", "Running", and "Clusters". The "Files" tab is active. Below the tabs, there's a text prompt: "Select items to perform actions on them." To the right of this prompt are three buttons: "Upload" (highlighted with a red box), "New", and a refresh icon. Below this is a table showing the file structure of the current directory, which is "/ lectures". The table has three columns: "Name", "Last Modified", and "File size". The table lists several folders and one file: "..", "images", "scripts", "solutions", "templates", and "preface.ipynb".

	Name	Last Modified	File size
<input type="checkbox"/>	0		
	/ lectures		
<input type="checkbox"/>	..	seconds ago	
<input type="checkbox"/>	images	7 hours ago	
<input type="checkbox"/>	scripts	7 hours ago	
<input type="checkbox"/>	solutions	7 hours ago	
<input type="checkbox"/>	templates	7 hours ago	
<input type="checkbox"/>	preface.ipynb	7 hours ago	2.32 kB

## Opening (uploading) notebooks: step 3

The selected file is displayed on top. Select **Upload** one more time.



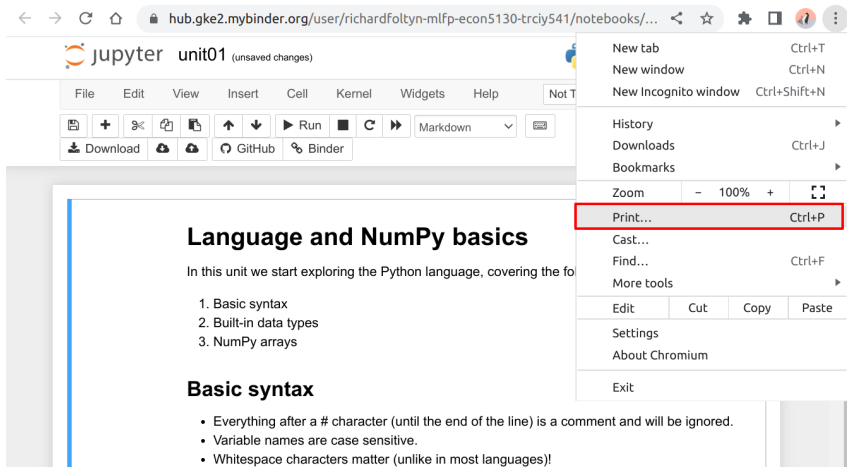
The screenshot shows the JupyterLab interface. At the top, there's a header with the Jupyter logo and the text "jupyter". To the right of the logo are three buttons: "Visit repo", "Copy Binder link", and "Quit". Below the header, there are three tabs: "Files", "Running", and "Clusters". The "Files" tab is active. Below the tabs, there's a text prompt "Select items to perform actions on them." and three buttons: "Upload", "New", and a refresh icon. The main area displays a file browser for the "/ lectures" directory. A red rectangle highlights the file "uploaded.ipynb" which is selected. To the right of the file name are two buttons: "Upload" and "Cancel". Below the highlighted file, there are three other files: "..", "images", and "scripts", each with a checkbox and a timestamp.

	Name	Last Modified	File size
<input type="checkbox"/>	0		
<input checked="" type="checkbox"/>	uploaded.ipynb		
<input type="checkbox"/>	..	seconds ago	
<input type="checkbox"/>	images	7 hours ago	
<input type="checkbox"/>	scripts	7 hours ago	

The uploaded file should now be visible in the list. Click on it to open the notebook.

# Saving as PDF: step 1

Downloading the notebook as PDF will most likely not work.  
Instead, use your browser's print function.



The screenshot shows a web browser window displaying a Jupyter notebook. The browser's address bar shows the URL: `hub.gke2.mybinder.org/user/richardfoltyn-mlfp-econ5130-trciy541/notebooks/...`. The Jupyter interface includes a top bar with the 'jupyter' logo and 'unit01 (unsaved changes)'. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. A toolbar contains icons for file operations, a 'Run' button, and a 'Markdown' dropdown. The notebook content area displays the title 'Language and NumPy basics' and a list of topics: '1. Basic syntax', '2. Built-in data types', and '3. NumPy arrays'. A section titled 'Basic syntax' follows, with a bulleted list: 'Everything after a # character (until the end of the line) is a comment and will be ignored.', 'Variable names are case sensitive.', and 'Whitespace characters matter (unlike in most languages)!'. A browser menu is open on the right, with the 'Print...' option highlighted by a red rectangle. The menu also shows options like 'New tab', 'New window', 'History', 'Downloads', 'Bookmarks', 'Zoom', 'Cast...', 'Find...', 'More tools', 'Edit', 'Cut', 'Copy', 'Paste', 'Settings', 'About Chromium', and 'Exit'.

hub.gke2.mybinder.org/user/richardfoltyn-mlfp-econ5130-trciy541/notebooks/...

jupyter unit01 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not T

Download GitHub Binder

## Language and NumPy basics

In this unit we start exploring the Python language, covering the fo

1. Basic syntax
2. Built-in data types
3. NumPy arrays

### Basic syntax

- Everything after a # character (until the end of the line) is a comment and will be ignored.
- Variable names are case sensitive.
- Whitespace characters matter (unlike in most languages)!

New tab Ctrl+T  
New window Ctrl+N  
New Incognito window Ctrl+Shift+N  
History  
Downloads Ctrl+J  
Bookmarks  
Zoom - 100% +  
**Print... Ctrl+P**  
Cast...  
Find... Ctrl+F  
More tools  
Edit Cut Copy Paste  
Settings  
About Chromium  
Exit



# Saving as PDF: step 2

## Select Save as PDF.

The screenshot shows a web browser window displaying a Jupyter Notebook titled "Language and NumPy basics". The notebook content includes a title, a list of topics, a "Basic syntax" section with bullet points, and three code cells. The first code cell shows a variable 'text' being created and printed. The second code cell shows the variable 'text' being printed. The third code cell shows a simple arithmetic expression '2\*3'. The right sidebar of the notebook shows the "Print" section with a "Destination" dropdown menu set to "Save as PDF". The "Pages" dropdown is set to "All" and the "Layout" dropdown is set to "Portrait". At the bottom right of the sidebar, there are "Cancel" and "Save" buttons. The "Save" button is highlighted with a red box.

hub.gke2.mybinder.org/user/richardfoltyn-mlfp-econ5130-trciy541/notebooks/...

1375522 9:28 PM u0051 - Jupyter Notebook

### Language and NumPy basics

In this unit we start exploring the Python language, covering the following topics:

1. Basic syntax
2. Built-in data types
3. NumPy arrays

#### Basic syntax

- Everything after a # character (until the end of the line) is a comment and will be ignored.
- Variable names are case sensitive.
- Whitespace characters matter (unlike in most languages)
- Python uses indentation (usually 4 spaces) to group statements, for example loop bodies, functions, etc.
- You don't need to add a character to terminate a line, unlike in some languages.
- You can use the `print()` function to inspect almost any object.

```
In [1]:  
# First example  
# create a variable named 'text' that stores the string 'Hello, world!'  
text = 'Hello, world!'  
# print contents of 'text'  
print(text)  
  
Hello, world!
```

In Jupyter notebooks and interactive command-line environments, we can also display a value by simply writing the variable name.

```
In [2]:  
text  
  
Out[2]:  
'Hello, world!'
```

Alternatively, we don't even need to create a variable but can instead directly evaluate expressions and print the result:

```
In [3]:  
2*3  
  
Out[3]:  
6
```

This does not print anything in proper Python script files that are run through the interpreter, though.

<https://hub.gke2.mybinder.org/user/richardfoltyn-mlfp-econ5130-trciy541/notebooks/lecture01.1.ipynb> 1/05

1375522 9:28 PM u0051 - Jupyter Notebook

Print 25 pages

Destination **Save as PDF**

Pages All

Layout Portrait

More settings

Cancel Save

**RUNNING LOCALLY ON YOUR COMPUTER**



Interactive notebooks can be also run locally on your computer

**Two options:**

- 1 Running in the browser with Jupyter lite (no installation required)
- 2 Installing a local Python environment

# Simple method: Running locally in browser (Jupyter lite)

## How to run a notebook in Jupyter lite:



- 1 Go to <https://github.com/richardfoltyn/MLFP-ECON5130>
- 2 Click on the button  try  (shown on the next slide)
- 3 You will be presented with a list of notebooks that can be run directly in the browser (locally, not in the cloud!)

## Limitations:





- Not all Python packages work in Jupyter lite, in particular those covered in unit 7 (pandas\_datareader, yfinance, fredapi, openpyxl, nasdaqdatalink) cannot be loaded
- Jupyter lite is experimental, so other things might not work

# Running notebooks locally in your browser

Launching notebooks from GitHub repository

 README.md 




## Machine Learning in Finance with Python - Part 1

License CC BY-NC-SA 4.0  try  lite now  launch  binder

Author: Richard Foltyn, University of Glasgow

### Units

This introductory course consists of several units. Each unit corresponds to one interactive Jupyter notebook, which is also available as a static PDF file. Alternatively, you can download the entire course as a [single PDF](#).

Unit	Title	PDF	Google Colab
1	Language and NumPy basics	<a href="#">PDF</a>	 <a href="#">Open in Colab</a>
2	Control flow and list comprehensions	<a href="#">PDF</a>	 <a href="#">Open in Colab</a>
3	Reusing code - Functions, modules and packages	<a href="#">PDF</a>	 <a href="#">Open in Colab</a>

# Advanced method: install a Python environment

## Getting Python

### ■ Python versions

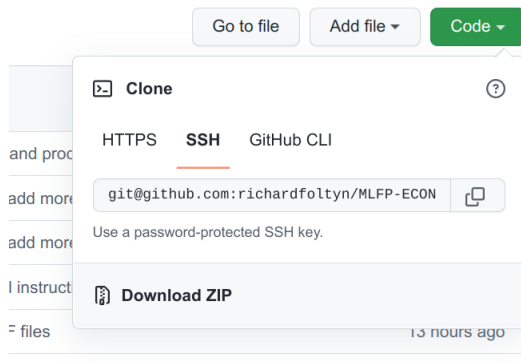
- Current version of Python is 3.10, but earlier version such as 3.9 and 3.8 probably work as well
- Do **not** use Python 2.7, it's no longer supported!

### ■ Python distribution

- The core Python language / runtime directly from the Python project (<https://www.python.org/>) is **not** particularly useful for statistics / data analysis / quantitative work
- Instead use distribution that allows you to easily install required packages
- Most popular distribution for scientific computing is Anaconda (<https://www.anaconda.com/products/distribution>)
  - For Windows, download 64-bit variant
  - For Linux, download 64-bit (x86) variant
  - For Mac there is only one option

# Getting the course material

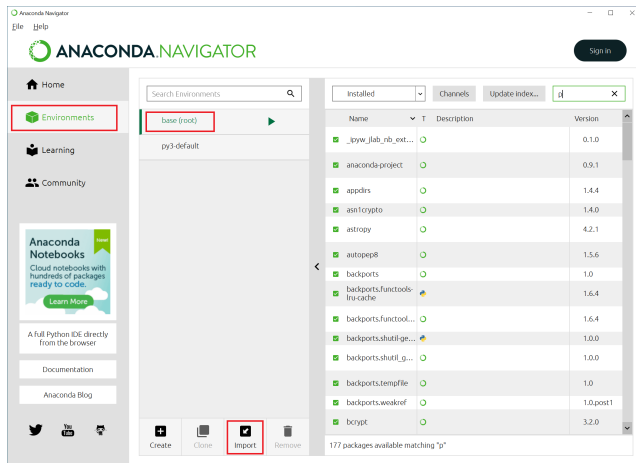
- You need to download all the content from <https://github.com/richardfoltyn/MLFP-ECON5130>
- For example, to download as a ZIP file:



- Alternatively, you can clone the repository if you are familiar with git.

# Setting up Anaconda: Microsoft Windows

Once you installed Anaconda, open the Anaconda Navigator application:



Initially, you'll have a single Python environment called **base (root)**



# Setting up Anaconda: Microsoft Windows and Mac

To make sure you have compatible versions of Python and of various packages, it is best to create a new environment.

- 1 Download the environment definition file [environment.yml](#) from the GitHub repository and save it locally.  
This file defines the packages and their exact versions required for this course.
- 2 In Anacoda Navigator, select **Import** (see previous slide) and enter the path to the environment file you just downloaded.
- 3 Call the new environment **MLFP** (for Machine Learning in Finance with Python), see screenshot on next slide.

This will create a Python environment with all packages required for this course.

# Setting up Anaconda: Microsoft Windows and Mac

Import Environment

Import from:

☒ Local drive

ers/rf165p/repos/teaching/MLFP-ECON5130/environment.yml

☐ Anaconda Nucleus

Sign in to save your environment

New environment name:

MLFP

☐ Overwrite existing environment

Cancel

Import

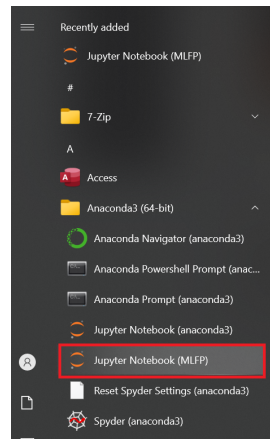
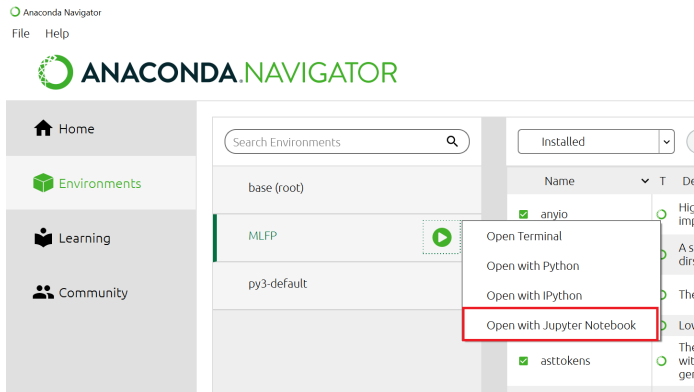
# Setting up Anaconda: Microsoft Windows and Mac

To start a browser-based Jupyter notebook that is running *locally* on your computer, use either of the following methods (see screenshots on next slide):

- 1 In Anaconda Navigator, select the **MLFP** environment, and from the context menu pick **Open with Jupyter Notebook**
  - 2 In the Windows Start Menu, search for and run the entry **Jupyter Notebook (MLFP)**
- Either alternative will launch your browser and open the Jupyter Notebook file manager.
  - Navigate to the folder where you unzipped the git repository contents, and select a notebook from the `lectures` folder, or the overview notebook `index.ipynb`.

# Setting up Anaconda: Microsoft Windows and Mac

## Launching a browser-based Jupyter Notebook



# Setting up Anaconda: Linux

- 1 Once you have installed Anaconda, you need to set up an environment that contains all the packages required to run the code.
- 2 Use environment specification in `environment.yml` from the git repository:
- 3 Active the virtual environment you just created (by default it will be called `MLFP`):

```
conda env create -f environment.yml
```

```
conda activate MLFP
```

- 4 Launch a local Jupyter instance:

```
cd /path/to/repository  
jupyter notebook index.ipynb
```

- The course material is provided as interactive notebooks within your browser
- For more serious programming, use local Python files and an editor!
- Python source files (\*.py) are plain-text files, so in principle you can use any editor you want

# Editors: Recommendations

- Visual Studio Code (<https://code.visualstudio.com>)
  - Flexible code editor with good support for Python
  - Free and open source
  - Official tutorial for Python programming with VS Code:  
<https://code.visualstudio.com/docs/python/python-tutorial>
  - VS Code also has excellent Jupyter Notebook support:  
<https://code.visualstudio.com/docs/datascience/jupyter-notebooks>
  - Data science tutorial with Python and VS Code:  
<https://code.visualstudio.com/docs/datascience/data-science-tutorial>
- PyCharm (<https://www.jetbrains.com/pycharm/>)
  - Most sophisticated integrated development environment (IDE) for Python
  - Proprietary, but has free community edition; professional edition is free for educational purposes
  - Use only if you are an experienced programmer, or willing to invest some time
- Spyder (<https://www.spyder-ide.org>)
  - Free and open source Python IDE
  - Default editor that comes with full Anaconda installation