

Introduction – Part 1

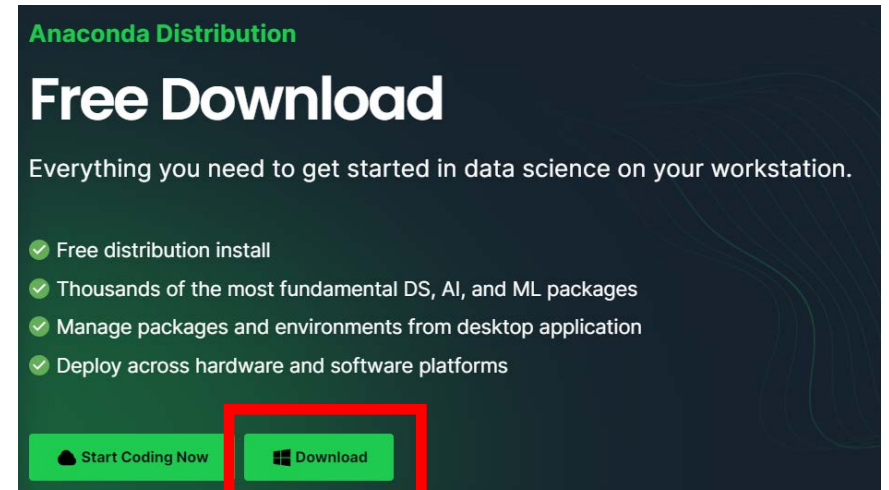
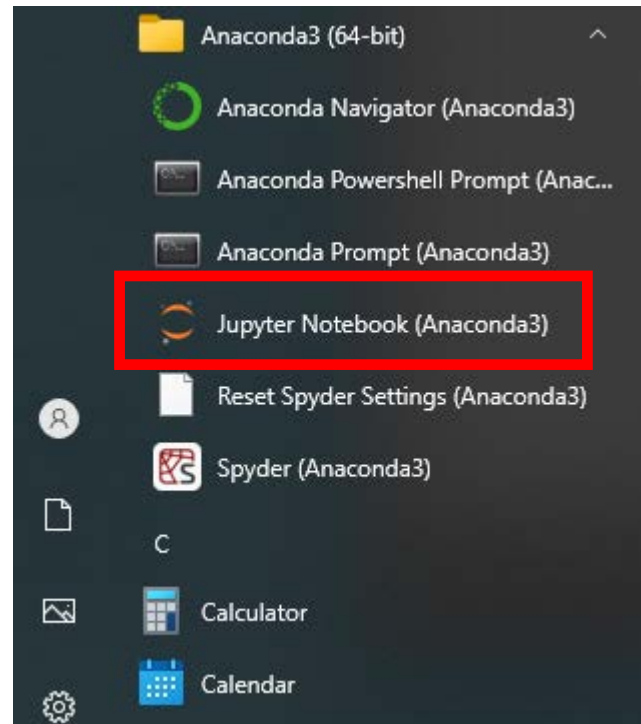
How to install and use Jupyter Notebook

Go to <https://www.anaconda.com/download>

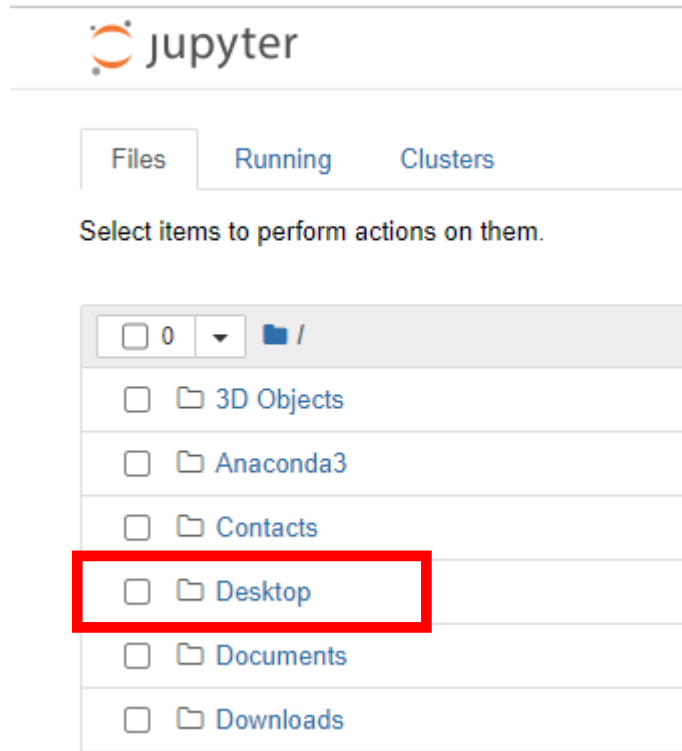
Download **Anaconda3**

Install **Anaconda3**

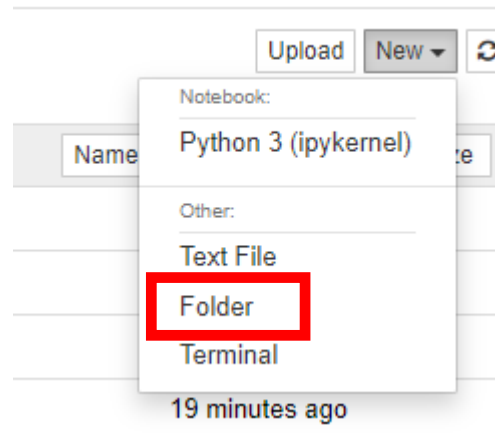
Click **Jupyter Notebook**



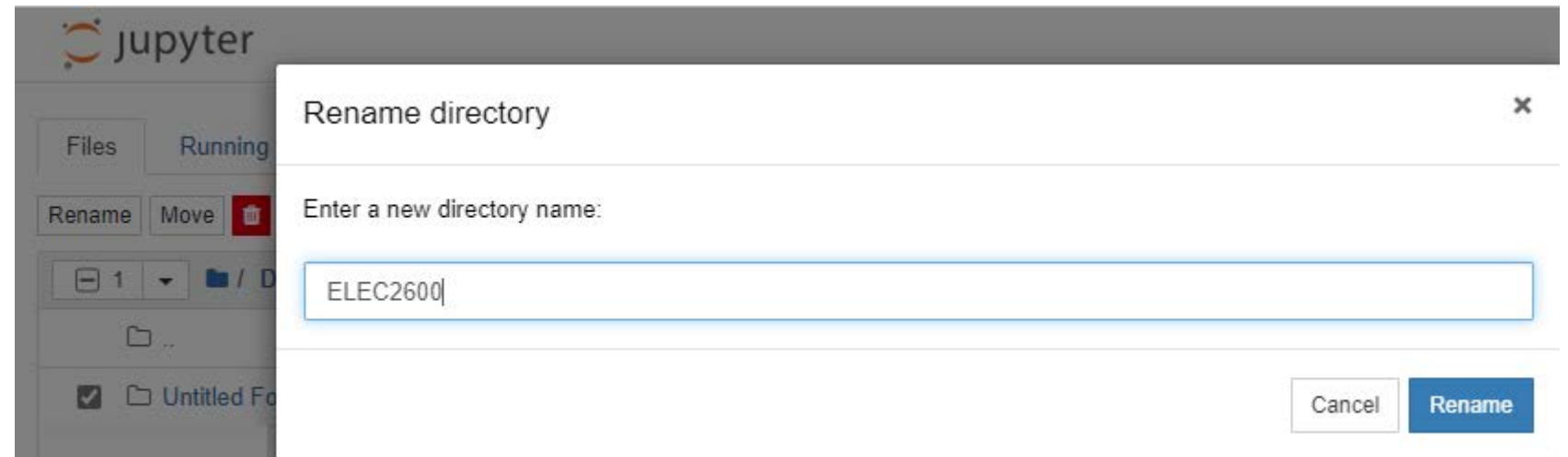
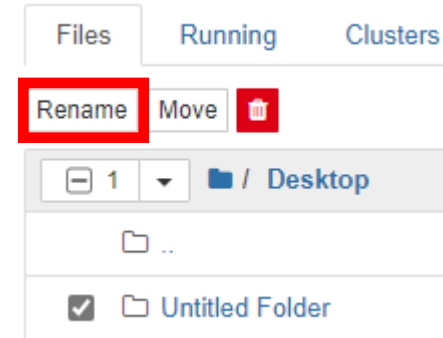
Click Desktop



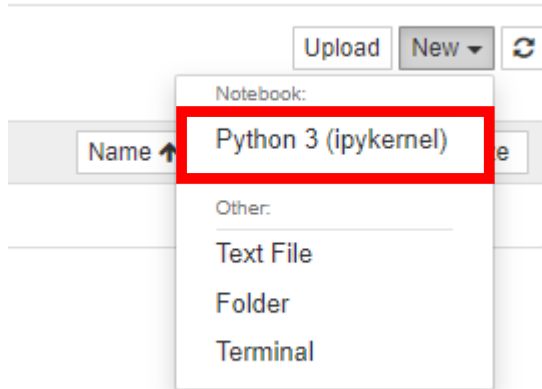
Create a Folder



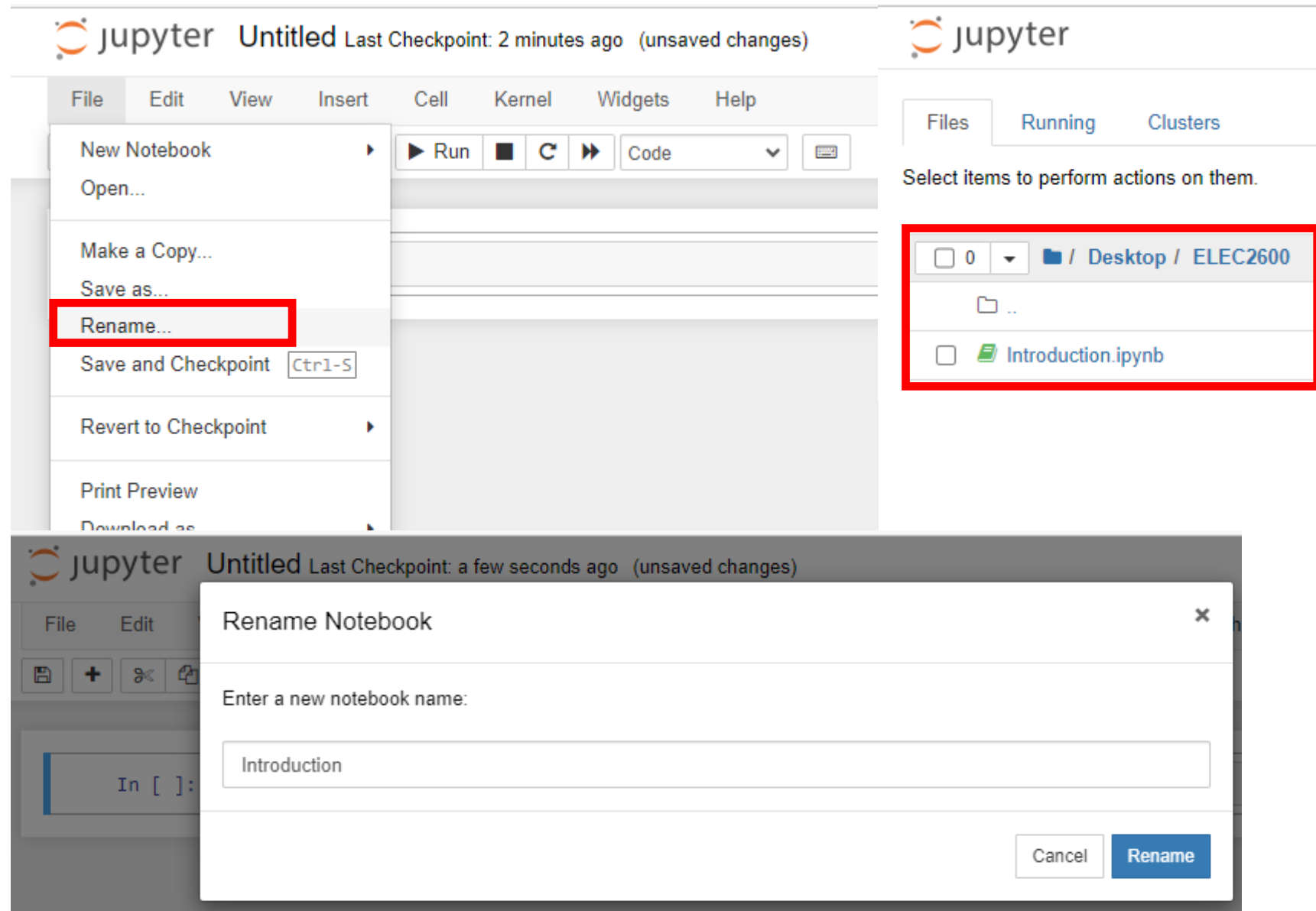
Rename the Folder



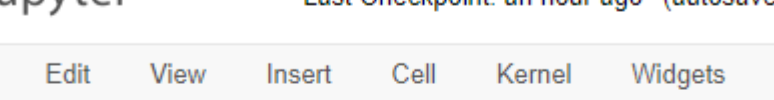
Create a new notebook



Rename the notebook

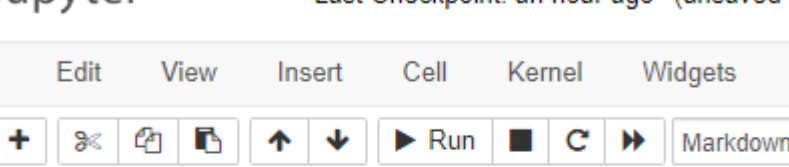


Program code and Run cell



The screenshot shows the JupyterLab interface. At the top, the Jupyter logo and name are on the left, and the text "Last Checkpoint: an hour ago (autosaved)" is on the right. Below this is a horizontal toolbar with tabs for File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Under the "Cell" tab, there is a row of icons: a save icon, a plus icon, a split icon, a copy icon, a paste icon, an up arrow, a down arrow, a "Run" button (a right-pointing triangle), a black square, a refresh icon, a "Code" button (two right-pointing triangles), and a dropdown arrow. The "Run" and "Code" buttons are highlighted with red rectangular boxes. A tooltip with the text "run cell, select below" is positioned below the "Run" button. Below the toolbar is a code editor area. It starts with a blue vertical line on the left, followed by the text "In []:". To the right of this, there is a code cell containing the following Python code:

```
a = 2
b = 5
a + b
```



The screenshot shows the JupyterLab interface. At the top, the Jupyter logo is on the left, and the text "Last Checkpoint: an hour ago (unsaved changes)" is on the right. Below this is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Under the "Cell" menu, a dropdown is open showing "Markdown" with a downward arrow. Below the menu bar is a toolbar with icons for saving, adding a new file, undo, redo, copy, paste, up arrow, down arrow, run (a play button), a square stop button, a refresh button, and a double right arrow. The main area contains a code cell with the prompt "In [12]:" followed by three lines of code: "a = 2", "b = 5", and "a + b". Below the code cell, the output is displayed as "Out[12]: 7".

Jupyter

Last Checkpoint: an hour ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

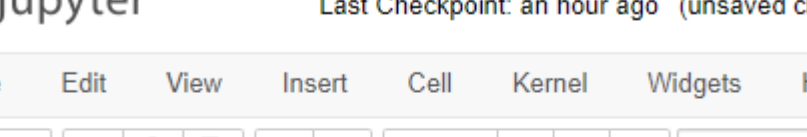
Markdown

```
In [12]: a = 2
          b = 5
          a + b
```

Out[12]: 7

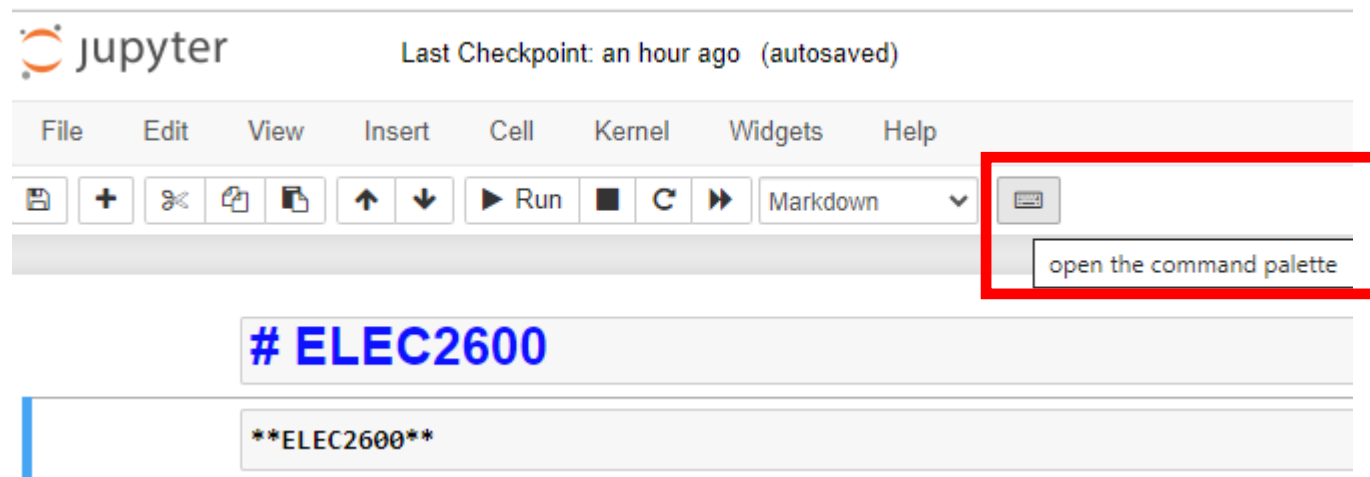
Text (Bold)

A screenshot of the JupyterLab web application interface. At the top left is the Jupyter logo and the word "jupyter". To its right, it says "Last Checkpoint: an hour ago (unsaved changes)". Below this is a horizontal menu bar with tabs: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Underneath the menu bar is a toolbar containing various icons for file operations (save, new, copy, paste), navigation (up, down arrows), execution (run, stop, refresh, next), and a dropdown menu currently set to "Markdown". A red rectangular box highlights the "Markdown" dropdown menu. The main workspace area below the toolbar contains a single code cell with the text "ELEC2600".

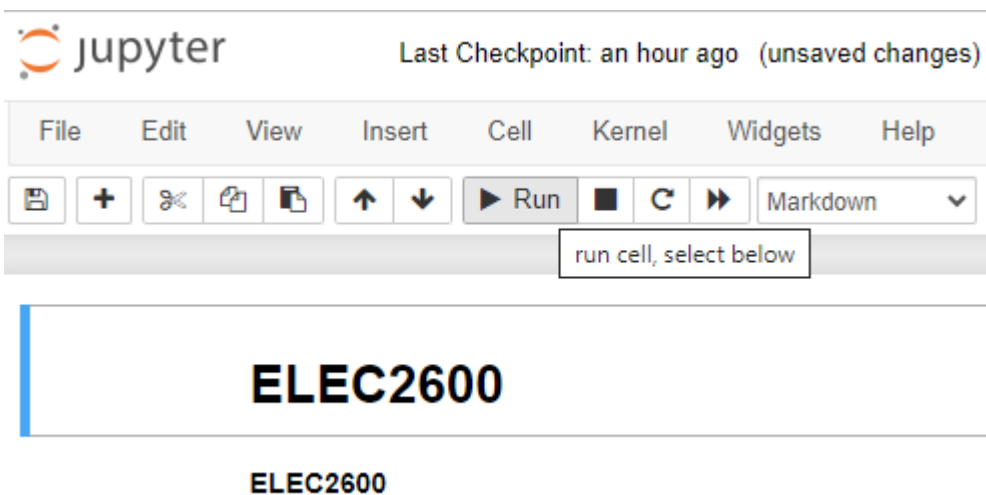


The screenshot shows the Jupyter Notebook interface. At the top, the Jupyter logo is on the left, and the text "Last Checkpoint: an hour ago (unsaved changes)" is on the right. Below this is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Under the "Cell" menu, a toolbar is visible with icons for saving, adding, deleting, duplicating, undo, redo, running, and a dropdown menu currently set to "Markdown". The notebook area contains two cells. The first cell is a code cell with the text "ELEC2600". The second cell is a markdown cell containing the text "**ELEC2600**" followed by the word "Bold".

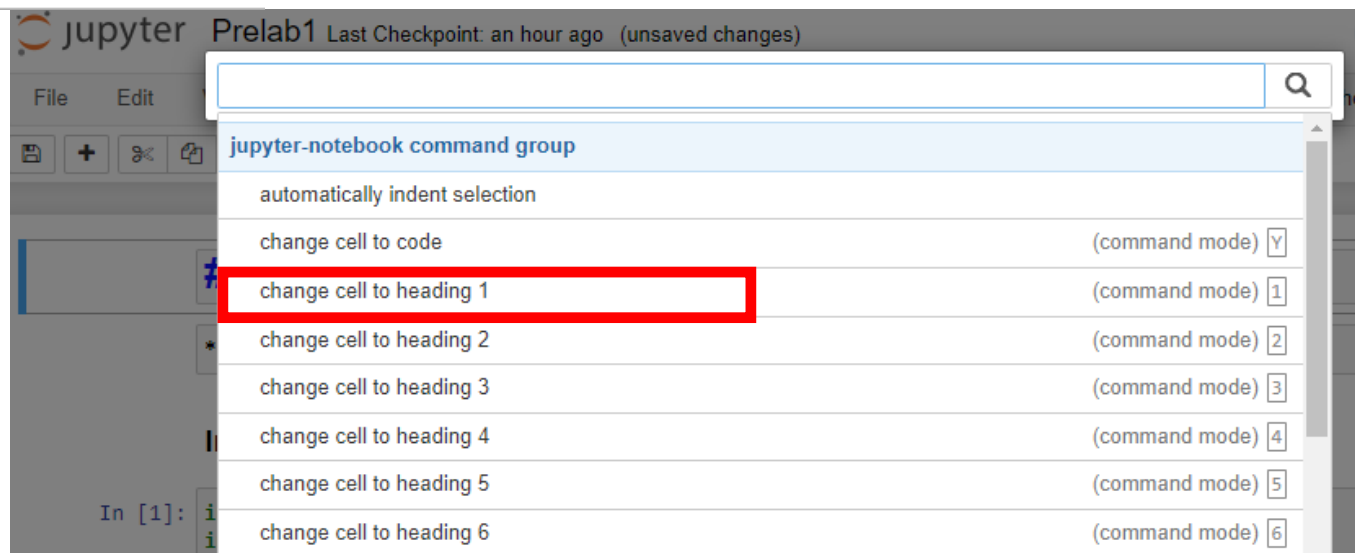
Text (Heading)



The Jupyter Notebook interface shows the top menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. Below the menu bar is a toolbar with icons for saving, adding, undo, redo, and navigation. A red box highlights the 'Command Palette' icon (a rectangle with a smaller rectangle inside) and a tooltip that says 'open the command palette'. The notebook content area shows a blue heading '# ELEC2600' and a code cell containing '**ELEC2600**'.



The Jupyter Notebook interface shows the top menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. Below the menu bar is a toolbar with icons for saving, adding, undo, redo, and navigation. A tooltip that says 'run cell, select below' is visible over the 'Run' button. The notebook content area shows the heading 'ELEC2600' in a large, bold, black font, and below it, the text 'ELEC2600' in a smaller, bold, black font.



The Jupyter Notebook interface shows the top menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. Below the menu bar is a toolbar with icons for saving, adding, undo, redo, and navigation. A search bar is visible at the top right. The command palette is open, showing a list of commands under the heading 'jupyter-notebook command group'. The command 'change cell to heading 1' is highlighted with a red box. Other commands in the list include 'automatically indent selection', 'change cell to code', 'change cell to heading 2', 'change cell to heading 3', 'change cell to heading 4', 'change cell to heading 5', and 'change cell to heading 6'.

Kernel

Jupyter interface showing the Kernel menu options:

- Restart
- Restart & Clear Output
- Restart & Run All
- Reconnect
- Shutdown
- Change kernel

Code cell content:

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
from pylab import *
```

Restart kernel?

Do you want to restart the current kernel? All variables will be lost.

Continue Running

Restart

Restart kernel and clear all output?

Do you want to restart the current kernel and clear all output? All variables and outputs will be lost.

Continue Running

Restart and Clear All Outputs

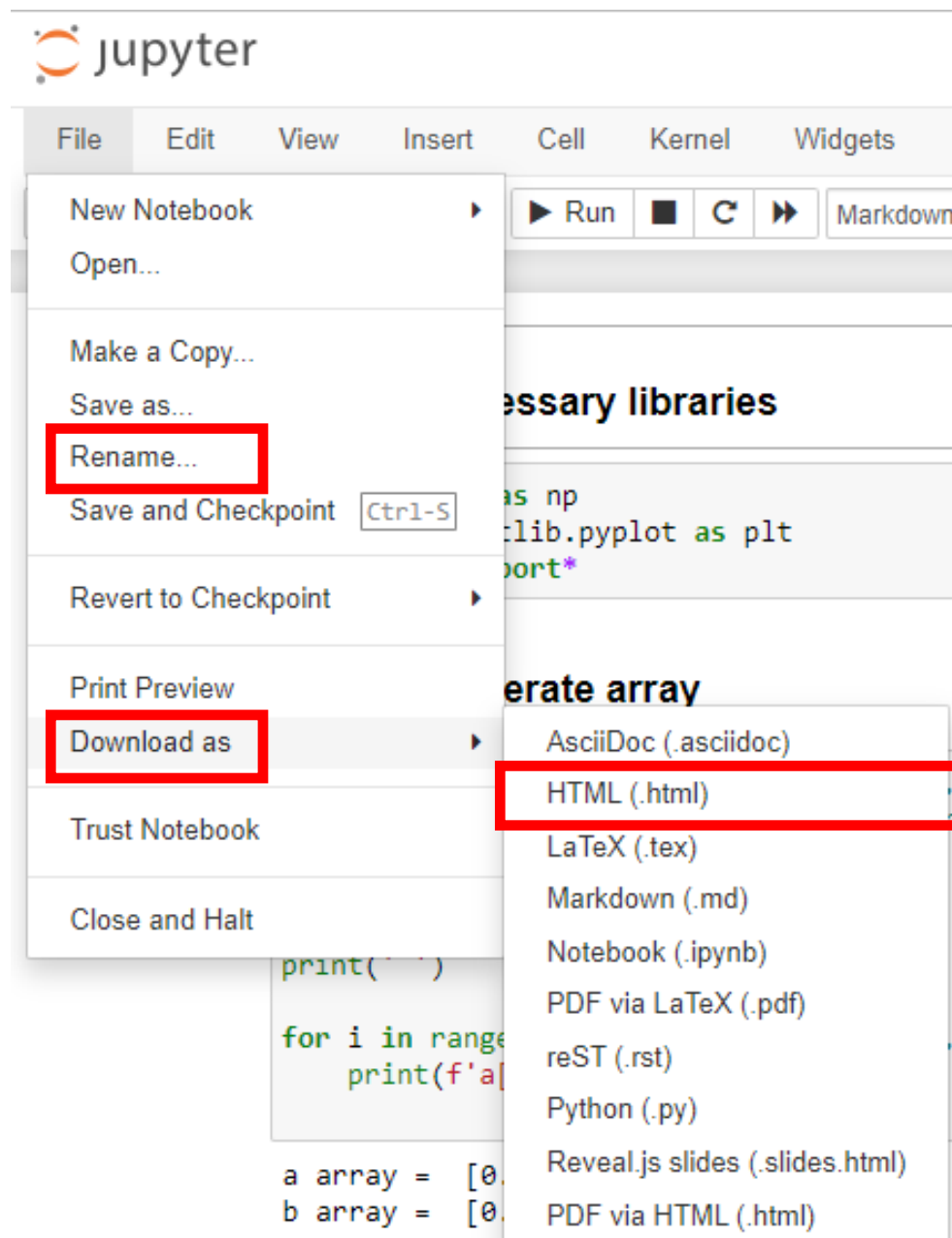
Restart kernel and re-run the whole notebook?

Are you sure you want to restart the current kernel and re-execute the whole notebook? All variables and outputs will be lost.

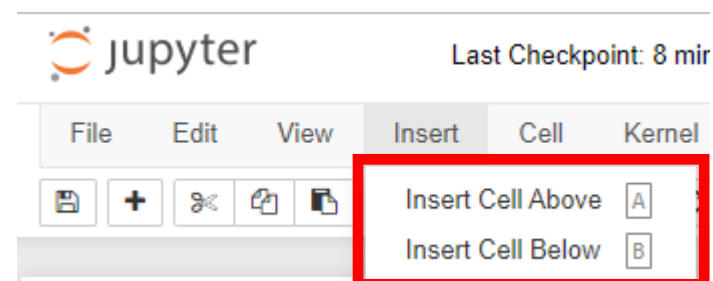
Continue Running

Restart and Run All Cells

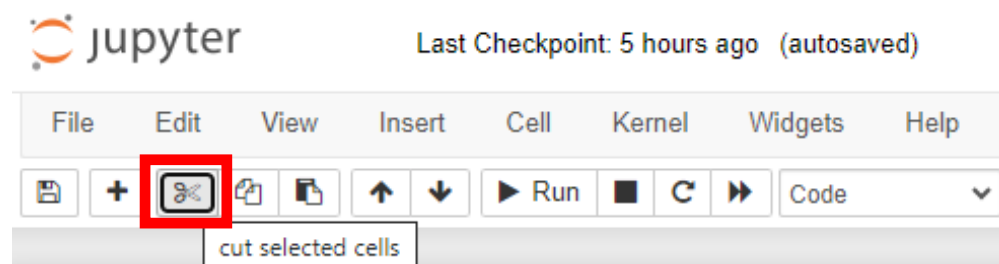
File



Insert



Delete a cell



Ex. 06

```
a = np.array([1, 4, 5])
b = np.array([1, 4, 5, 5, 4, 1, 1, 1])

for i in range(a.size):
    print(f'a[{i}] = ', a[i])
    compare = np.equal(b, a[i])
    print(compare)
    count = np.cumsum(compare)
    print(count)
    print(' ')
```

```
a[0] = 1
[ True False False False False  True  True  True]
[1 1 1 1 1 2 3 4]
```

```
a[1] = 4
[False  True False False  True False False False]
[0 1 1 1 2 2 2 2]
```

```
a[2] = 5
[False False  True  True False False False False]
[0 0 1 2 2 2 2 2]
```

np.equal

np.cumsum

How to obtain the relative frequency versus n ?

How to plot the relative frequency versus n ?

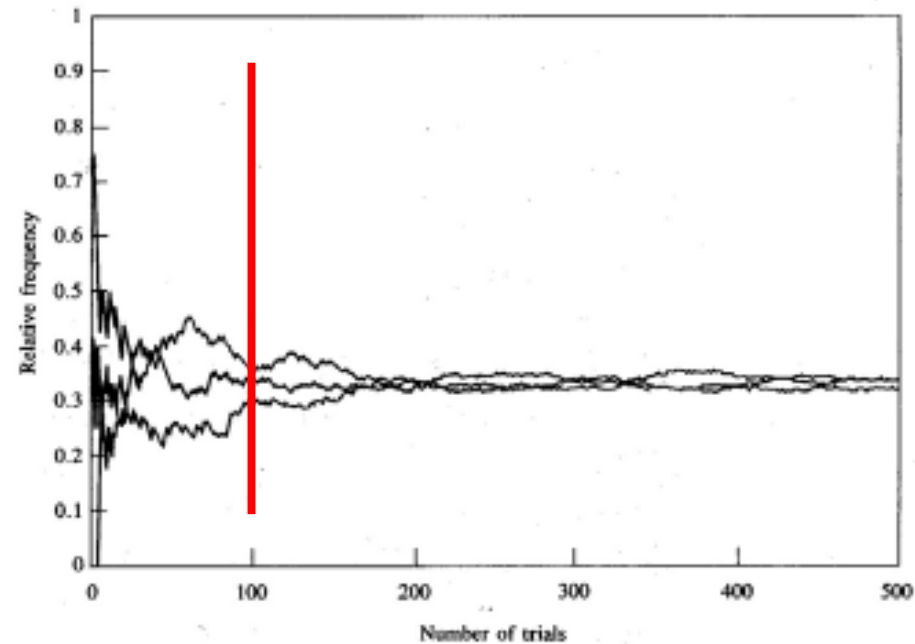
Relative frequency

- Let $N_1(n)$, $N_2(n)$ and $N_3(n)$ be the *number of times* that we pick balls 1, 2, and 3 in n trials (**events**).
- Define the **relative frequency** of the outcome k as $f_k(n) = \frac{N_k(n)}{n}$
- This experiment exhibits **statistical regularity**: as n increases, the relative frequency approaches a **constant value**

$$\lim_{n \rightarrow \infty} f_k(n) = p_k$$

where p_k indicates the probability of outcome k .

Provides a key connection between measurement of physical quantities and probability models!

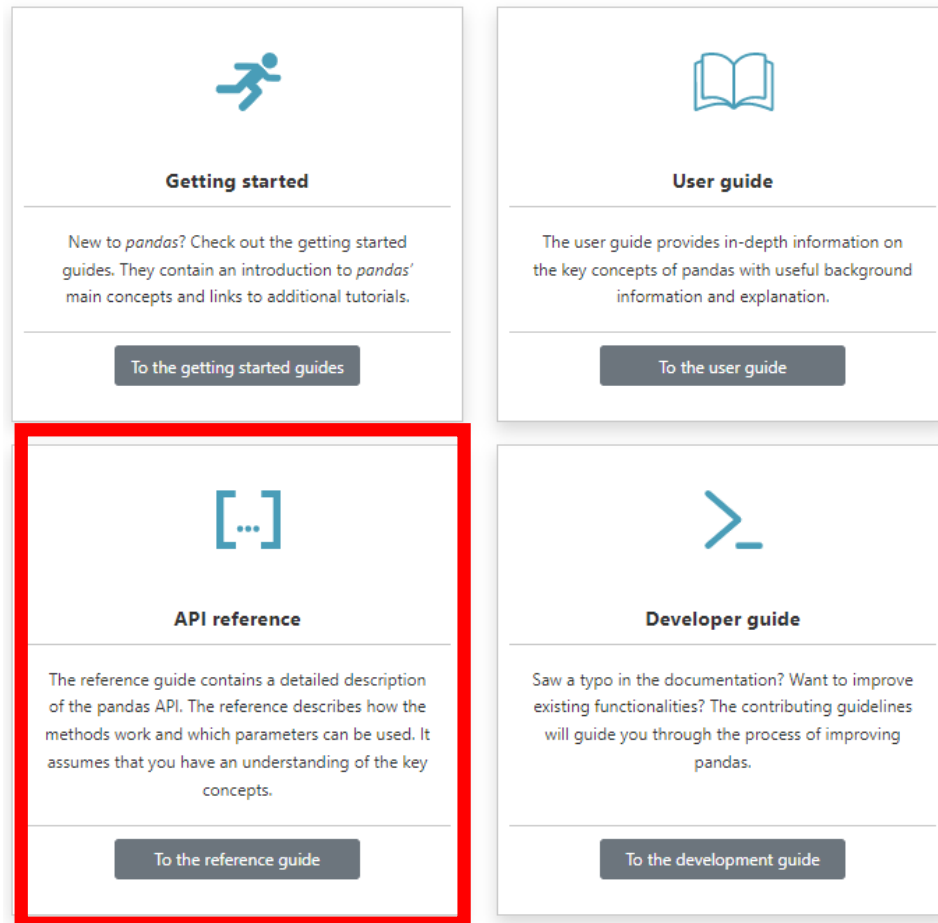


Introduction – Part 2

How to handle excel file

Go to <https://pandas.pydata.org/pandas-docs/stable/index.html>

Click **API reference**



The screenshot displays the pandas documentation index page, which is organized into four distinct sections arranged in a 2x2 grid. Each section features a unique icon, a title, a brief description, and a button for further exploration. The 'API reference' section, located in the bottom-left quadrant, is highlighted with a prominent red rectangular border. This section includes a code icon, the title 'API reference', a description of the reference guide's content, and a button labeled 'To the reference guide'. The other sections include 'Getting started' (runner icon), 'User guide' (book icon), and 'Developer guide' (greater-than sign icon).

Section	Icon	Description	Button
Getting started	Runner	New to <i>pandas</i> ? Check out the getting started guides. They contain an introduction to <i>pandas</i> ' main concepts and links to additional tutorials.	To the getting started guides
User guide	Book	The user guide provides in-depth information on the key concepts of pandas with useful background information and explanation.	To the user guide
API reference	Code [...]	The reference guide contains a detailed description of the pandas API. The reference describes how the methods work and which parameters can be used. It assumes that you have an understanding of the key concepts.	To the reference guide
Developer guide	>	Saw a typo in the documentation? Want to improve existing functionalities? The contributing guidelines will guide you through the process of improving pandas.	To the development guide

Click **DateFrame**



Getting started User Guide **API reference** Development Release notes

Input/output

General functions

Series

DataFrame

pandas.DataFrame

pandas.DataFrame.index

pandas.DataFrame.columns

pandas.DataFrame.dtypes

pandas.DataFrame.info

pandas.DataFrame.groupby

DataFrame.groupby(*by=None, axis=0, level=None, as_index=True, sort=True, group_keys=True, observed=False, dropna=True*) # [\[source\]](#)

Group DataFrame using a mapper or by a Series of columns.

A groupby operation involves some combination of splitting the object, applying a function, and combining the results. This can be used to group large amounts of data and compute operations on these groups.

Click and read the description of the followings (groupby, count and sum)

Function application, GroupBy & window

<code>DataFrame.apply</code> (func[, axis, raw, ...])	Apply a function along an axis of the DataFrame.
<code>DataFrame.applymap</code> (func[, na_action])	Apply a function to a Dataframe elementwise.
<code>DataFrame.pipe</code> (func, *args, **kwargs)	Apply chainable functions that expect Series or DataFrames.
<code>DataFrame.agg</code> ([func, axis])	Aggregate using one or more operations over the specified axis.
<code>DataFrame.aggregate</code> ([func, axis])	Aggregate using one or more operations over the specified axis.
<code>DataFrame.transform</code> (func[, axis])	Call <code>func</code> on self producing a DataFrame with the same axis shape as self.
<code>DataFrame.groupby</code> ([by, axis, level, ...])	Group DataFrame using a mapper or by a Series of columns.
<code>DataFrame.rolling</code> (window[, min_periods, ...])	Provide rolling window calculations.
<code>DataFrame.expanding</code> ([min_periods, axis, method])	Provide expanding window calculations.
<code>DataFrame.ewm</code> ([com, span, halflife, alpha, ...])	Provide exponentially weighted (EW) calculations.

Computations / descriptive stats

<code>DataFrame.abs</code> ()	Return a Series/DataFrame with absolute numeric value of each element.
<code>DataFrame.all</code> ([axis, bool_only, skipna])	Return whether all elements are True, potentially over an axis.
<code>DataFrame.any</code> (*[, axis, bool_only, skipna])	Return whether any element is True, potentially over an axis.
<code>DataFrame.clip</code> ([lower, upper, axis, inplace])	Trim values at input threshold(s).
<code>DataFrame.corr</code> ([method, min_periods, ...])	Compute pairwise correlation of columns, excluding NA/null values.
<code>DataFrame.corrwith</code> (other[, axis, drop, ...])	Compute pairwise correlation.
<code>DataFrame.count</code> ([axis, numeric_only])	Count non-NA cells for each column or row.
<code>DataFrame.sem</code> ([axis, skipna, ddof, numeric_only])	Return unbiased standard error of the mean over requested axis.
<code>DataFrame.skew</code> ([axis, skipna, numeric_only])	Return unbiased skew over requested axis.
<code>DataFrame.sum</code> ([axis, skipna, numeric_only, ...])	Return the sum of the values over the requested axis.
<code>DataFrame.std</code> ([axis, skipna, ddof, numeric_only])	Return sample standard deviation over requested axis.

Ex.01 – Groupby and Count

```
print(data)
```

	Class	Price
Index		
1	fruit	5
2	fruit	5
3	fruit	10
4	fruit	10
5	fruit	3
6	vegetable	1
7	vegetable	1
8	vegetable	5
9	vegetable	2
10	vegetable	4
11	fruit	5
12	fruit	5
13	vegetable	1
14	vegetable	5
15	vegetable	2
16	vegetable	4

Ex.01 - Groupby and Count

```
n_class = data.groupby('Class')['Price'].count()
print(n_class)
print(' ')

print('Number of fruits = ', n_class['fruit'])
print(' ')

print('Total number of products', sum(n_class))
```

```
Class
fruit      7
vegetable  9
Name: Price, dtype: int64
```

```
Number of fruits = 7
```

```
Total number of products 16
```

Relative frequency ?

Ex.02 – Groupby and Sum

```
print(data)
print(' ')

n_class = data.groupby('Class')['Price'].sum()
print(n_class)
```

Total price of fruit = ?

Total price of vegetable = ?

Index	Class	Price
1	fruit	5
2	fruit	5
3	fruit	10
4	fruit	10
5	fruit	3
6	vegetable	1
7	vegetable	1
8	vegetable	5
9	vegetable	2
10	vegetable	4
11	fruit	5
12	fruit	5
13	vegetable	1
14	vegetable	5
15	vegetable	2
16	vegetable	4

```
Class
fruit      43
vegetable   25
Name: Price, dtype: int64
```


Ex.03 – Create a new column

```
data['data_price'] = data['Price'] > 4
print(data)
print(' ')

print('The sum of data_price = ', sum(data['data_price']))
print(' ')

n_fruit_4 = data.groupby('Class')['data_price'].sum()
print(n_fruit_4)
```

True = 1 **False = 0**

fruit (Price > 4) + vegetable (Price > 4) = 8

fruit (Price > 4) = ?

vegetable (Price > 4) = ?

	Class	Price	data_price
Index			
1	fruit	5	True
2	fruit	5	True
3	fruit	10	True
4	fruit	10	True
5	fruit	3	False
6	vegetable	1	False
7	vegetable	1	False
8	vegetable	5	True
9	vegetable	2	False
10	vegetable	4	False
11	fruit	5	True
12	fruit	5	True
13	vegetable	1	False
14	vegetable	5	True
15	vegetable	2	False
16	vegetable	4	False

The sum of data_price = 8

```
Class
fruit      6
vegetable  2
Name: data_price, dtype: int64
```

Event A

Event B

Prior probability of event A

$$P(A)$$

Prior probability of event B

$$P(B)$$

Event A and Event B both occur

$$P(A \cap B)$$

Conditional probability

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

What is the prior probability of fruit ?

What is the probability $P[(\text{Price} > 4) \cap \text{Fruit}]$?

What is the conditional probability $P[(\text{Price} > 4) | \text{Fruit}]$?