

Introduction:

Python is a high-level and general-purpose programming language that is developed by Guido van Rossum in 1991. It is widely used in various fields, such as software development, web development, numerical analysis, machine learning, and data science.

Python provides many useful features which make it powerful and popular. In the following, you can see a list of the most important features:

1) Free and Open Source:

Python is available on its official website www.python.org. It is completely free to use, even for commercial purposes. There is a large community across the world that makes new python modules and functions. You can download source codes, use them as well as share them.

2) Easy to Learn and Code:

Python is a high-level language. Its syntax is straightforward and much the same as the English language. It's easier to read and write Python programs compared to many languages such as C, C++ or Java.

3) Expressive Language:

Python programs are generally smaller than other programming languages like C or Java and you can perform complex tasks using a few lines of code. This increases the readability of program.

4) Cross Platform:

Python works on various operating systems such as Windows, Linux, Mac, etc. This makes it a cross platform and portable language.

5) Interpreted Language:

Python is an Interpreted Language; it means that the source code is processed at runtime line by line and you do not need to compile your program before executing it. It leads to easier debugging and portability.

6) Large Standard Library:

Python provides a vast range of libraries for various fields such as: machine learning, text processing, image processing, web scraping, etc. The libraries provide a rich set of modules and functions so you do not have to write your own code for every single thing.

7) Dynamic Typing:

Python is a dynamically-typed language. That means the type for a variable (e.g.: int, double, long, ...) is decided at run time not in advance. Therefore, we don't need to specify the type of variable and the memory is automatically allocated to a variable at runtime when it is given a value.

8) Multiple Programming Paradigms Support:

Python supports Object-Oriented programming (a style of programming that encapsulates code within objects) as well as Procedural and Functional paradigms.

9) GUI Programming Support:

One of the key aspects of any programming language is supporting GUI or Graphical User Interface. There are various options for developing GUI in Python including PyQt, Tkinter, and Kivy frameworks.

Introduction to Anaconda:

Anaconda is the world's most popular data science platform, which helps users manage a collection of over 7,500+ open-source packages available to them. Anaconda equips individuals to easily search, install, and run thousands of Python/R packages and access a vast library of community content and support. It also makes creating, saving and loading programs very straightforward.

Installing Anaconda

1) Before installing Anaconda, check the system requirements listed below:

System requirements:

- Operating system: Windows 8 or newer, 64-bit macOS 10.13+, or Linux, including Ubuntu, RedHat, CentOS 7+, and others.

(If your operating system is older than what is currently supported, you can find older versions of the Anaconda installers that might work for you on the Anaconda's archive page.)

- Minimum 5 GB disk space to download and install.

Downloading and installing on Windows:

2) You can download Anaconda installers from

<https://www.anaconda.com/products/distribution>

Click on 'Get Started' and enter your email address in the newly opened window and click on 'Submit'.

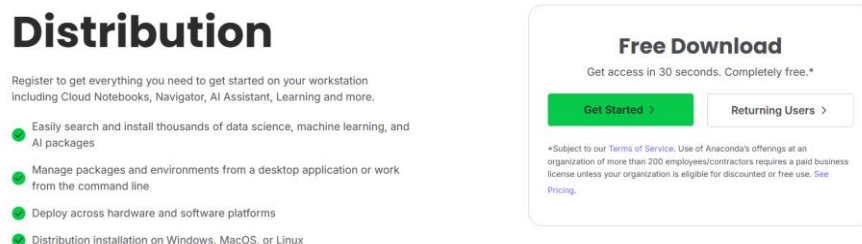


Figure 1. Anaconda distribution

It will direct you to the next page.

Distribution Installers



For installation assistance, refer to [troubleshooting](#).

Windows	▼
Mac	▼
Linux	▼

Miniconda Installers



For installation assistance, refer to [troubleshooting](#).

Windows	▼
Mac	▼
Linux	▼

Figure 2. Downloading Anaconda

- 3) Based on your operating system click on the appropriate link to download the installer from 'Distribution Installer'.
- 4) Go to your Downloads folder and **right click** on the exe file and click on "**Run as administrator**" and click Yes in the window. (If you encounter issues during installation, temporarily disable your anti-virus software during installation, then re-enable it after the installation)
- 5) Click on the next button (Figure 4).

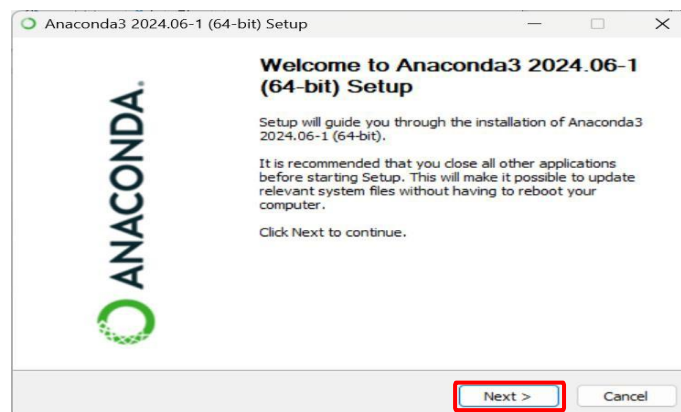


Figure 4. Anaconda setup

- 6) Read the licensing terms and click on the "I Agree".
- 7) In the next step (Figure 5), choose "Just me" option and click on the Next button.

(Only select an installation for All Users if you need to install for all users' accounts on the computer. This requires Windows Administrator privileges)

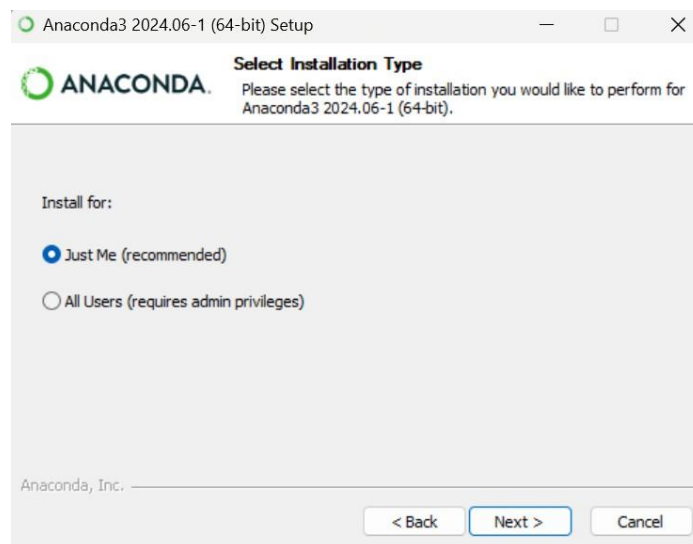


Figure 5. Select installation type

- 8) Select a destination folder to install Anaconda. You can use the Browse button to change the location (The directory path should not contain spaces or unicode characters). After choosing install location click on the next button.

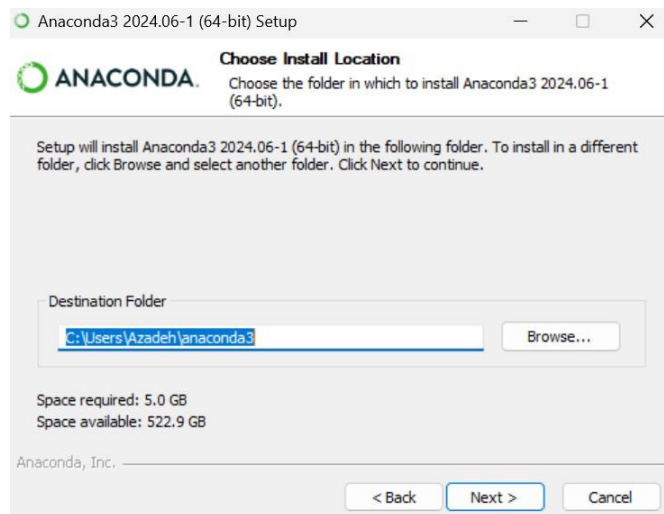


Figure 6. Choose install location

- 9) In the next step (Figure 7.), check “Register Anaconda3 as my default Python 3.12” and click on the **Install** button.

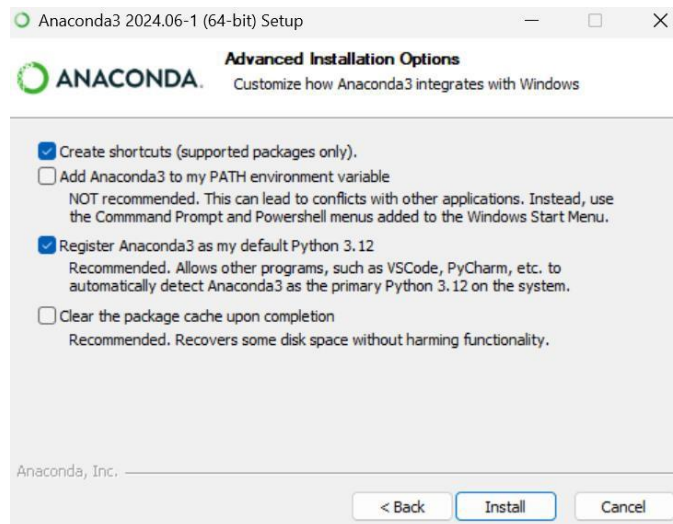


Figure 7. Installation options

- 10) Please wait until Anaconda3 is installed. It will take a few minutes. Then the Next button will be enabled.

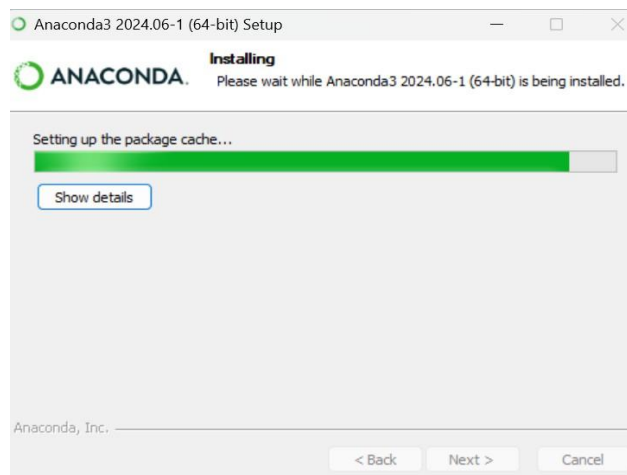


Figure 8. Anaconda3 Setup

- 11) In the next Dialog box, click on the Next button. Finally, you should see the “Completing Anaconda3 Setup” dialog box (Figure 9.). Click the Finish button to

complete the installation (If you wish to read more about Anaconda.org and how to get started with Anaconda, check the boxes “Anaconda Distribution Tutorial” and “Getting started with Anaconda”).

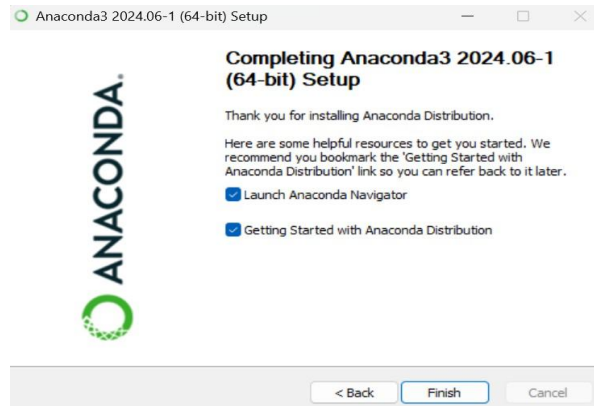


Figure 9. Completing Anaconda3 Setup

How to work with Anaconda:

If you have installed the program successfully, you can find Anaconda Navigator and Conda on your computer.

- Anaconda Navigator

Anaconda Navigator is a graphical user interface (GUI) that is automatically installed with Anaconda.

12) To open it in Windows, click Start, search for Anaconda Navigator, and click on it (Figure 10).

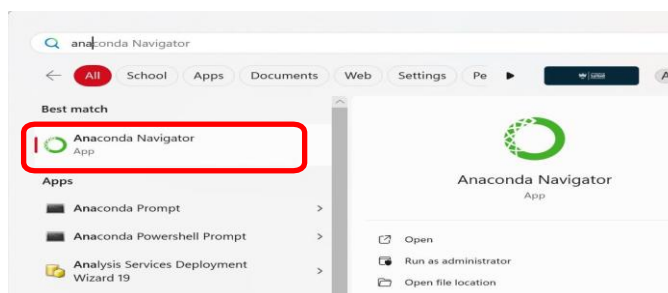


Figure 10. Opening Anaconda Navigator

You should see Figure 11 (It will take few minutes).

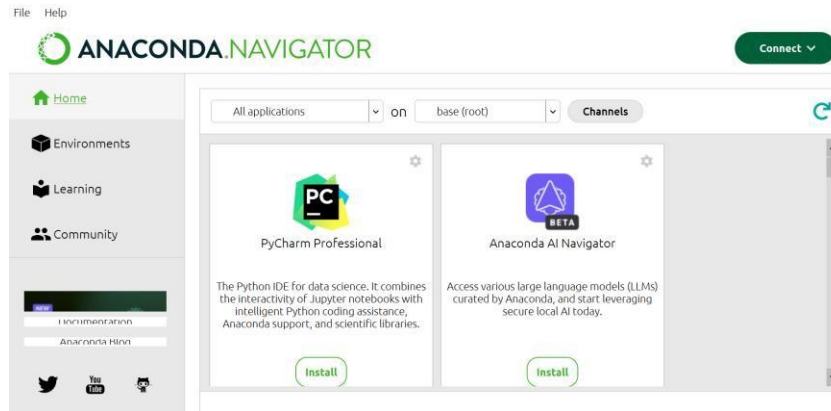


Figure 11. Anaconda Navigator

- Conda

Conda is a command line interface (CLI) that is automatically installed with Anaconda.

- 13) To open it in Windows, click Start, search for Anaconda Prompt, and click on it (Figure 12).

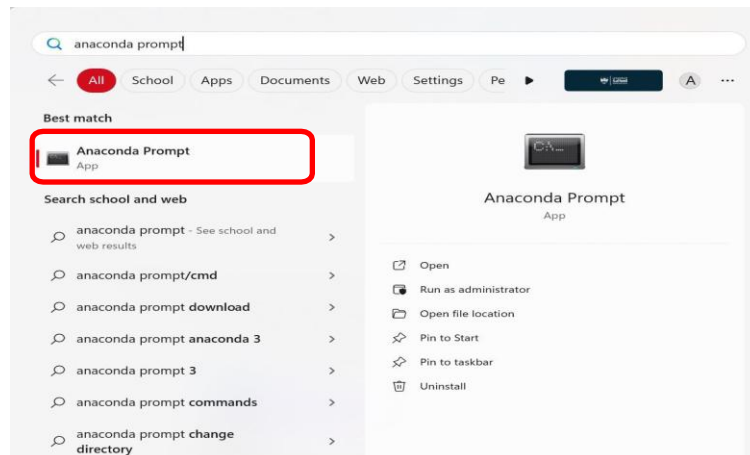
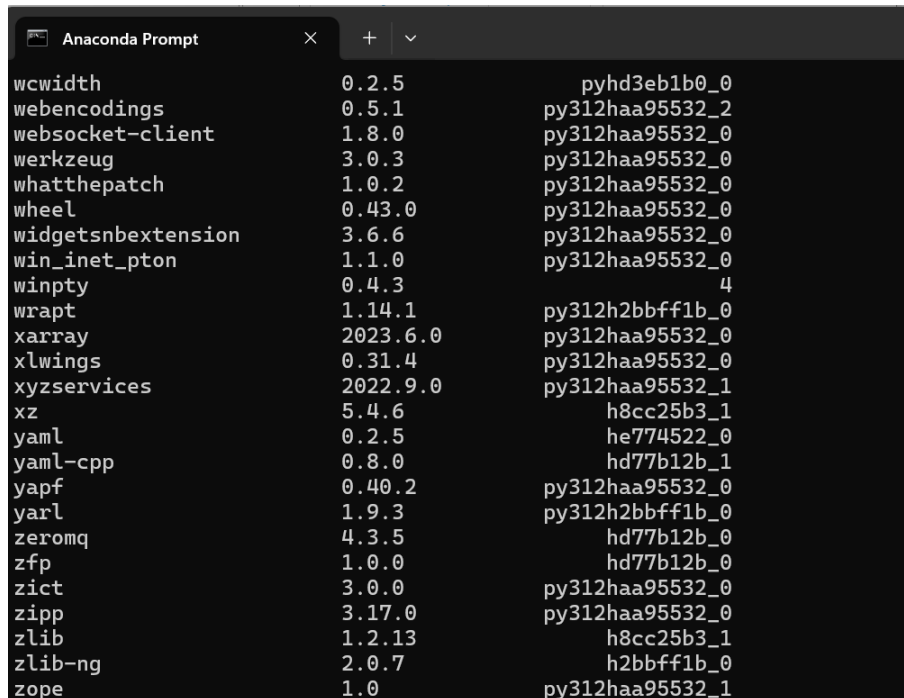


Figure 12. Coda (Anaconda prompt)

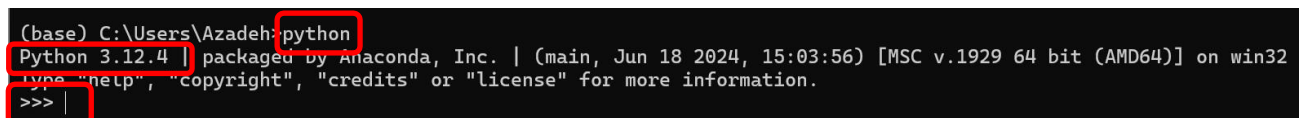
- 14) If you write `conda list` in the Anaconda prompt and enter, a list of installed packages and their versions will be displayed.



wcwidth	0.2.5	pyhd3eb1b0_0
webencodings	0.5.1	py312haa95532_2
websocket-client	1.8.0	py312haa95532_0
werkzeug	3.0.3	py312haa95532_0
whatthepatch	1.0.2	py312haa95532_0
wheel	0.43.0	py312haa95532_0
widgetsnbextension	3.6.6	py312haa95532_0
win_inet_pton	1.1.0	py312haa95532_0
winpty	0.4.3	4
wrapt	1.14.1	py312h2bbff1b_0
xarray	2023.6.0	py312haa95532_0
xlwings	0.31.4	py312haa95532_0
xyzservices	2022.9.0	py312haa95532_1
xz	5.4.6	h8cc25b3_1
yaml	0.2.5	he774522_0
yaml-cpp	0.8.0	hd77b12b_1
yapf	0.40.2	py312haa95532_0
yaml	1.9.3	py312h2bbff1b_0
zeromq	4.3.5	hd77b12b_0
zfp	1.0.0	hd77b12b_0
zict	3.0.0	py312haa95532_0
zipp	3.17.0	py312haa95532_0
zlib	1.2.13	h8cc25b3_1
zlib-ng	2.0.7	h2bbff1b_0
zope	1.0	py312haa95532_1

Figure 13. Coda List

- 15) If you enter the command `python` it runs the Python shell. You will see a python version and the command line will change to `>>>` (Figure 14). Now you can write your python codes and execute them. To exit the Python shell, you should enter the command `quit()`.



```

(base) C:\Users\Azadeh>python
Python 3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:03:56) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>

```

Figure 14. Running python shell in Anaconda prompt

(However, in this lab we use Jupyter Notebooks from Anaconda Navigator for writing and executing python code.)

Run Python in a Jupyter Notebook:

Jupyter Notebook is an interactive development environment, in which you can combine code execution, text, equations, and visualization.



- 16) Open Anaconda Navigator as in step 12. On Navigator's Home tab, scroll to the Jupyter Notebook tile and click the Launch button (Figure 15). This will launch a **new browser window (or a new tab)** showing the Notebook Dashboard (Figure 16). (If Jupyter is not installed, you will see an install button instead and you should click on it to install Jupyter).

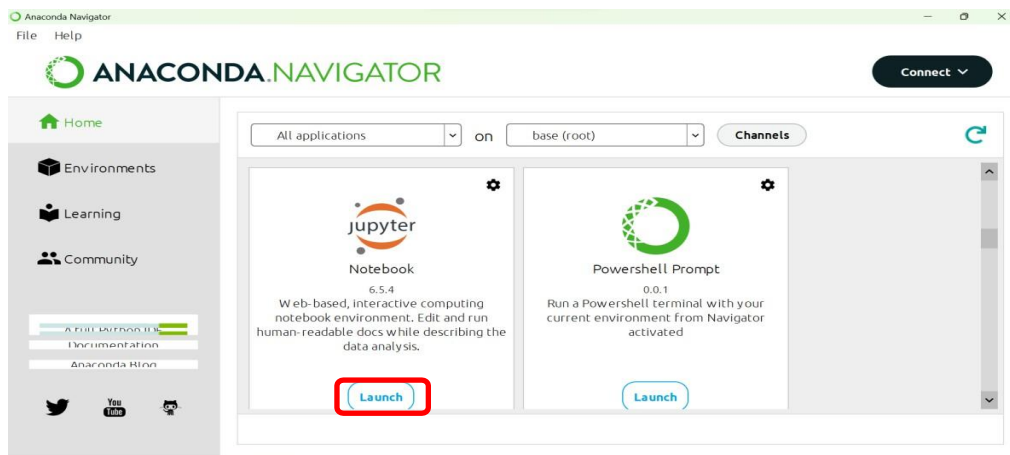


Figure 15: Start Jupyter Notebook

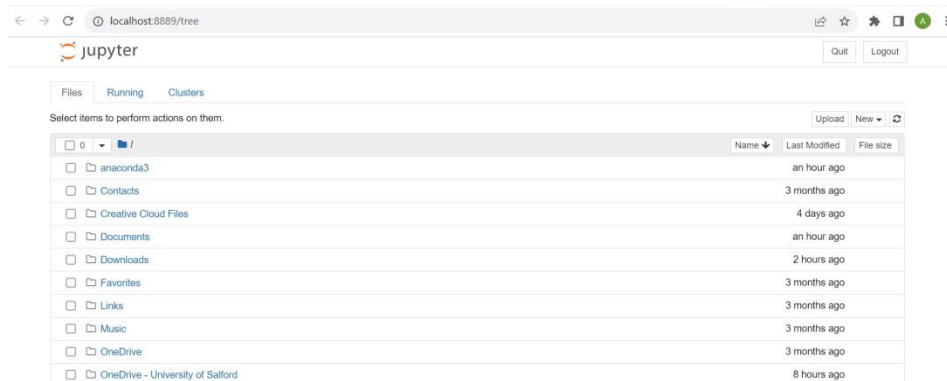


Figure 16. Notebook Dashboard

A **notebook** is a single document where you can run code, display the output, add explanations, formulas, charts. It makes your work more transparent, understandable, and shareable.

17) To create a new notebook, click on “New” dropdown menu on the top right-hand side and select Python3 (Figure 17).

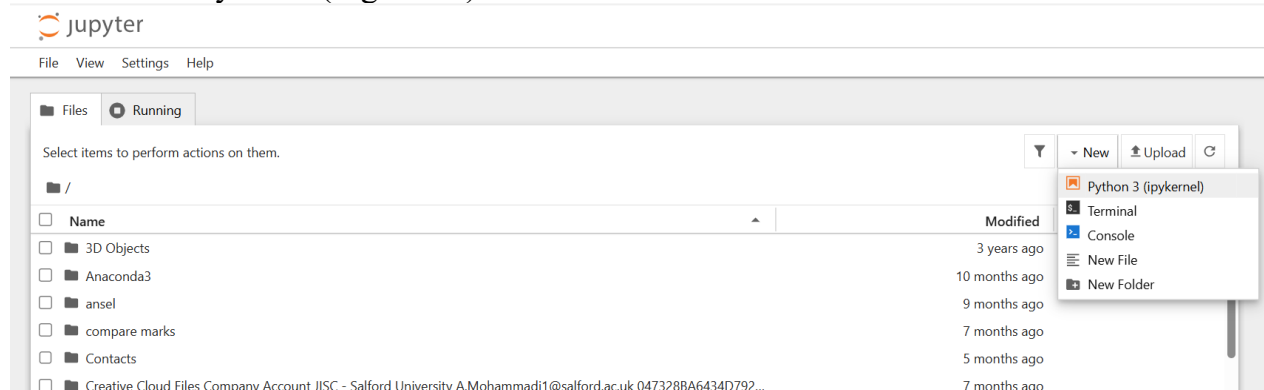


Figure 17. Create a new notebook

18) A new notebook will be opened in a new browser tab.

19) To rename your notebook, either click on the current name and edit it or use File>Rename in the top menu bar (Figure 18). You can rename the notebook to Workshop1.

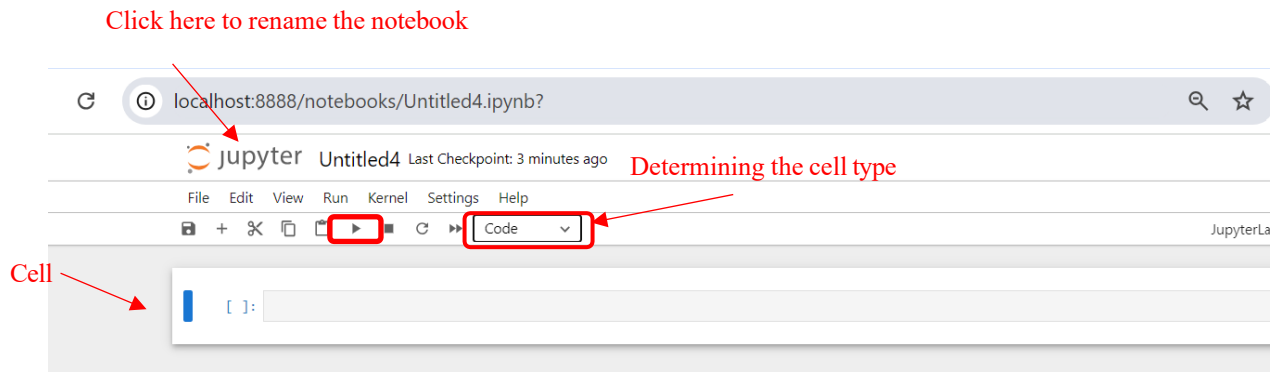


Figure 18. A new notebook

In a notebook, you write the code or texts in different cells. You can determine the cell type by clicking on a dropdown box at the top of notebook (see Figure 18.). You can see the output of each cell by clicking Run button.

There are different types for a cell:

- **Code:** If you want to write a python statement, you should select code type. By clicking Run, the code will be executed.
- **Markdown:** If you write anything in a cell with Markdown type it will be a simple text.

`This is the first workshop`

If you write the above text in a Markdown cell it will change to the image below after clicking on the Run button:

`This is the first workshop`

- For headings, write your headings in Markdown cells using # characters:

`# This is a level 1 heading`

If you write the above text in a Markdown cell, it will change to the image below after clicking on the Run button:

This is a level 1 heading

If you add another # it will change to level2 heading:

This is a level 2 heading

Output after Run:

This is a level 2 heading

20) Exercise: Write a level 3 heading text

- **Raw:** Contents in raw cells are not evaluated by notebook kernel. If you convert the notebook to another format (such as HTML or Latex, cells marked as Raw Format will be converted in a way specific to the output you're targeting.
- 21) In the next cell, write the below html code and change the type of cell to Raw.

```
<html>
<a href="https://www.salford.ac.uk">Click here</a>
</html>
```

Click on the Run and see what happens.

This is in fact an html code which directs you to a link. So, you can see the result if you save it in html format. From File menu, click 'Save and export Notebook As' and select HTML. An HTML file will be downloaded. Click on it in the folder to open it. In the opened browser tab click on the "Click here" to see the result. Change "Click here" in the above code to "Salford University" and download it as html again. What is the difference?

Writing your first program in Python:

print() function:

22) **Printing a message (string):** In the first program, we want to show a "Hello World!" message.

You can use the `print()` function to show a message on the screen or other standard outputs (e.g., files). (**Python is case-sensitive**, so be careful about uppercase and lowercase letters when you write function name, variable name or keywords, e.g., `print()` and `Print()` are different)

The print function either takes direct input or it can take a variable (we will talk about variables later).

A message is a string. Strings in python are surrounded by either single quotation marks, or double quotation marks.

So, to show the message Hello World! on the screen, write `print('Hello World!')` in the first line of the notebook and click on the Run button (Make sure the cell type is set to Code). After running the code, you can see Hello World! in the next cell (output cell)

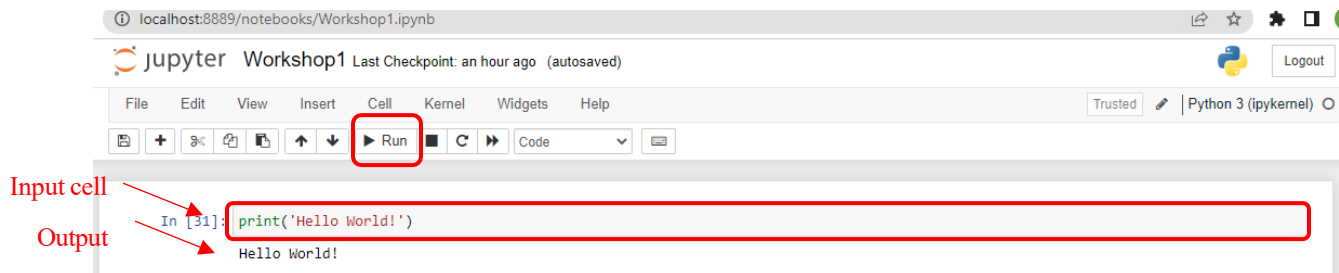


Figure 19. Printing Hello World!

23) Printing other data types: You can print different types such as numbers. Execute `print(10)` and see the output.

24) Printing multiple values: you can print multiple values. You can separate the values with commas. In the next cell, write `print(1,2,3)` and see the output.

The separator between the arguments of print() function is space by default. But you can use 'sep' parameter to change it.

Try the code below:

```
print('a','b','c')
print('a','b','c',sep=',')
print('a','b','c',sep='-')
```

25) Multiline printing: If you want to print values in different lines, there are several

ways:

- a) You can use different print functions for each line. The print function by default creates a new line after its output.

Type the below code in the next cell and see the result.

```
print('Hello')  
print('World!')
```

- b) You can use new line character (\n) between your values. Execute the following code and see the result:

```
print('Hello\nWorld!')
```

- c) If you want to print a message in multiple lines, it can be done by multiline string using three single quotes.

Execute the following code and see the result:

```
print(  
'''  
*****  
*                                     *  
*      Learning Python              *  
*                                     *  
*****  
'''  
)
```

- 26) **Concatenating string variables and values in the print() function:** You can use + symbol in the print function to concatenate string variables or string values in the output

Write the below code in the notebook and see the result:

```
print('Week1'+ ' '+ 'Workshop')
```

What about if you want to concatenate a number to strings? You can find the answers in the next parts.

Variables:

A Python variable is a name given to a memory location. We can store values in the variables. Python is a dynamically typed language. This means that unlike many other languages, we do not need to declare variables or declare their type before using them. A variable is created the moment we first assign a value to it.

Variable name:

- A variable name can only contain alpha-numeric characters and underscores (a-z, A-Z, 0-9, and _).
- A variable name must start with a letter or the underscore character (it cannot start with a number). Variable names can start with an underscore character, but we generally avoid it unless we are writing library codes.
- Underscore characters are often used in names with multiple words, such as `circle_area`.
- Variable names are case-sensitive (`name`, `Name` and `NAME` are three different variables).
- It is better to use meaningful name as the variable name.
- The reserved words(keywords) (Figure 20) shouldn't be used as the variable name.

<code>and</code>	<code>del</code>	<code>from</code>	<code>None</code>	<code>True</code>
<code>as</code>	<code>elif</code>	<code>global</code>	<code>nonlocal</code>	<code>try</code>
<code>assert</code>	<code>else</code>	<code>if</code>	<code>not</code>	<code>while</code>
<code>break</code>	<code>except</code>	<code>import</code>	<code>or</code>	<code>with</code>
<code>class</code>	<code>False</code>	<code>in</code>	<code>pass</code>	<code>yield</code>
<code>continue</code>	<code>finally</code>	<code>is</code>	<code>raise</code>	<code>async</code>
<code>def</code>	<code>for</code>	<code>lambda</code>	<code>return</code>	<code>await</code>

Figure 20. List of keywords

An assignment statement assigns value to a variable. To display the value of a variable, you can use a print statement.

27) Type the code below in a cell and see the result:

```
msg='Area'  
r=2  
pi=3.14  
print(msg)  
print(r)  
print(pi)
```

The type of a variable is the type of value it refers to. You can get the type of variables by `type()` function.

28) Type the below code in 3 different cells and see the results:

```
type(msg)
```

```
type(r)
```

```
type(pi)
```

29) If you want to specify the data type of a variable, this can be done with casting.

Write the code below in a cell and then get the type of each variable.

```
x=str(2)  
y=int(2)  
z=float(2)
```

Expressions:

An expression is a combination of values, variables and operators. Operators are special symbols that represent computations like addition and multiplication.

Operators:

Table 1. List of operators

Operator	Type of operation
+	addition
-	subtraction
*	multiplication
/	float division
//	integer division
%	return remainder
**	exponentiation

30) Write the expressions below and see the results:

(pay attention to the print function as well.)

```
x=3
y=4
add=x+y
sub=x-y
mul=x*y
fdiv=x/y
idiv=x//y
mod=y%x
power=x**y

print('The addition of x and y is: '+str(add))
print('The subtraction of y from x is: '+str(sub))
print('The multiplication of x and y is: '+str(mul))
print('The floating division of x by y is: '+str(fdiv))
print('The integer division of x by y is: '+str(idiv))
print('The remainder of y/x: '+str(mod))
print('x to the power of y: '+str(power))
```

Order of operations: When more than one operator appears in an expression, the order of computation depends on the rules of precedence.

Rules of precedence:

- Parentheses () have the highest precedence and can be used to force the order of evaluation you want.
- Exponentiation has the next highest precedence.

- Multiplication and division have the same precedence that is higher than addition and subtraction.
- Operators with the same precedence are evaluated from left to right.

31) Write the code below and see the results.

```
e1=15-3*2
e2=(15-3)*2
e3=15-3**2/9
e4=15/3*5
e5=15/(3*5)
e6=15%4/2
e7=15%(4/2)
e8=15//4/2
e9=15/4//2
e10=15/(4//2)
```

```
print('e1',e1,sep=' ')
print('e2',e2,sep=' ')
print('e3',e3,sep=' ')
print('e4',e4,sep=' ')
print('e5',e5,sep=' ')
print('e6',e6,sep=' ')
print('e7',e7,sep=' ')
print('e8',e8,sep=' ')
print('e9',e9,sep=' ')
print('e10',e10,sep=' ')
```

Commenting:

Comments can be used to explain the code and make it more readable.

In Jupyter notebook, you can add texts and explanations in Markdown cells. But if you want to add comments in Python language, you should add them in code cells.

A comment in Python will start with a #. When you run the code, comments will not be executed. Comments can be placed at the end of a line, and Python will ignore the rest of the line.

32) Write the below code in a code cell and see the result:

```
# This part prints sum of two numbers
x=5 # First operand
y=6 # Second operand
print('sum:',x+y)
```

If you want to add multiline comments, you can put it in three single quotes.

33) Execute the below code and see the result.

```
...
This part prints sum of two numbers
x is the first operand
y is the second operand
...
x=5
y=6
print('sum:',x+y)
```

Libraries in Python:

A library is a collection of modules (a file consisting of Python code). A library contains bundles of code that can be used repeatedly in different programs. It makes Python programming simpler and more convenient for the programmer because we don't need to write the same code again and again for different programs.

In the following, we mention some of important libraries we use in data science and machine learning.

NumPy:



NumPy is an important library for scientific computing in Python. NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices, and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

The Anaconda distribution of Python comes with NumPy pre-installed and no further installation steps are necessary. But if you are not using the Anaconda distribution, you

should install the NumPy before using it. Executing the following code will install the NumPy.

```
pip install numpy
```

When it is installed, you should import it to your program using import statement:

```
import numpy
```

After that you can use numpy in your program. You can give an alias to numpy and use that in the program, like the following code.

```
import numpy as np
```

34) Execute the code below and see the result:

```
import numpy as np
a=np.arange(6)
print(a)
```

Pandas: pandas

Pandas is a library for data manipulation and analysis in Python programming language. It offers data structures and operations for manipulating tables and time series. It has functions for analyzing, cleaning, exploring, and manipulating data.

The Anaconda distribution of Python comes with pandas pre-installed and no further installation steps are necessary. But if you are not using the Anaconda distribution, you should install the pandas before using it. Executing the following code will install the pandas.

```
pip install pandas
```

When it is installed, you should import it to your program using import statement and then you can use its function in your program:

```
import pandas
```

scikit-learn:



Scikit-learn is a machine learning library for the Python programming language. This library is focused on modeling data and provides several regression, classification and clustering algorithms.

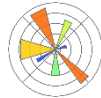
The Anaconda distribution of Python comes with scikit-learn pre-installed and no further installation steps are necessary. But if you are not using the Anaconda distribution, you should install the scikit-learn before using it. Executing the following code will install the scikit-learn.

```
pip install scikit-learn
```

Once you are done with the installation, you can use scikit-learn easily in your Python code by importing it as:

```
import sklearn
```

Matplotlib:



Matplotlib is a data visualization library that is used for creating static, interactive and animated charts in Python.

The Anaconda distribution of Python comes with Matplotlib pre-installed and no further installation steps are necessary. But if you are not using the Anaconda distribution, you should install the matplotlib before using it. Executing the following code will install the matplotlib.

```
pip install matplotlib
```

When it is installed, you should import it to your program using import statement:

```
import matplotlib
```

Most of the Matplotlib utilities lies under the pyplot submodule, and are usually imported under the plt alias. You can then use the functions available in the plt object.

```
import matplotlib.pyplot as plt
```

Seaborn:

Seaborn is a Python data visualization library based on matplotlib. It extends the Matplotlib library for creating beautiful graphics with Python using a more straightforward set of methods and provides a high-level interface for drawing attractive and informative statistical graphics. If one is doing statistics then Seaborn is a good choice because it has a lot of things suitable for statistical tasks, such as distribution, trends and relationships of variables.

The Anaconda distribution of Python comes with Seaborn pre-installed and no further installation steps are necessary. But if you are not using the Anaconda distribution, you should install the seaborn before using it. Executing the following code will install the seaborn.

```
pip install seaborn
```

When it is installed, you should import it to your program using import statement:

```
import seaborn as sns
```

You will learn more about different libraries in future sessions.

Saving a Jupyter notebook:

For saving your Jupyter notebook you can click on disk icon in the upper left of the Jupyter tool bar (Figure 21).

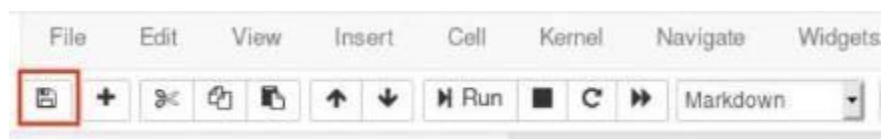


Figure 21. Saving your notebook

Also, you can save a notebook using the "File" > "Save Notebook as" option from the menu (Figure 22).

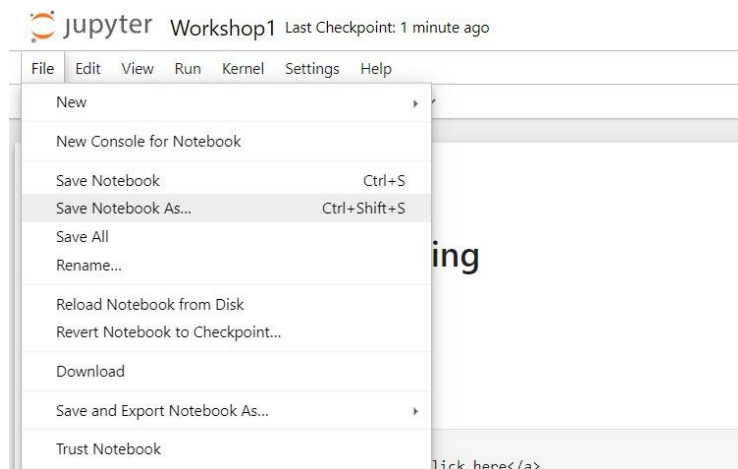


Figure 22. Download your file in different formats

Close your Jupyter Notebook File:

After saving your file, from File menu click on Shutdown and then you can close it by clicking close button (X) on the browser tab.

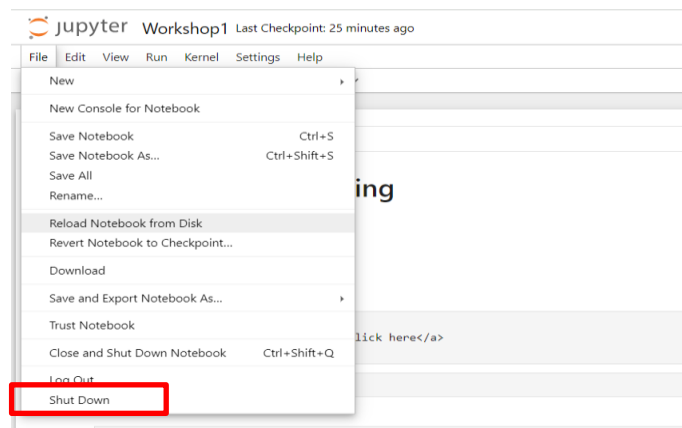


Figure 23. Shutdown the file