Introduction to Jupyter

This notebook explores basic and advanced features of Jupyter Notebooks, including Markdown formatting, code execution, data visualization, widgets, and extensions.

What is Jupyter Notebook?

Jupyter is an open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. This can help us run blocks of code, in a divided fashion, keeping the code that was ran earlier in memort, in the sequence we run it.

In [4]:

```
2 + 2
```

Out[4]:

```
4
```

Toolbar

We can use the buttons in toolbar, to add, delete or modify the type of cell that we are working with.

My Markdown Practice

This is **bold**, this is *italic*, and this is `inline code`.

Images and links

Google

My Favorite Programming Languages

- Python
- Java
- C++
- C

In [5]:

```
def factorial(n):
    if n < 0:
        return "Invalid input"
    elif n == 0:
        return 1
    else:
        result = 1
        for i in range(1, n + 1):
            result *= i
        return result
```

In [6]:

```
print(factorial(5))
print(factorial(0))
print(factorial(-3))
```

```
120
1
Invalid input
```

In [7]:

```
import numpy as np

data = [10, 20, 30, 40, 50]
print("Mean:", np.mean(data))
print("Median:", np.median(data))
```

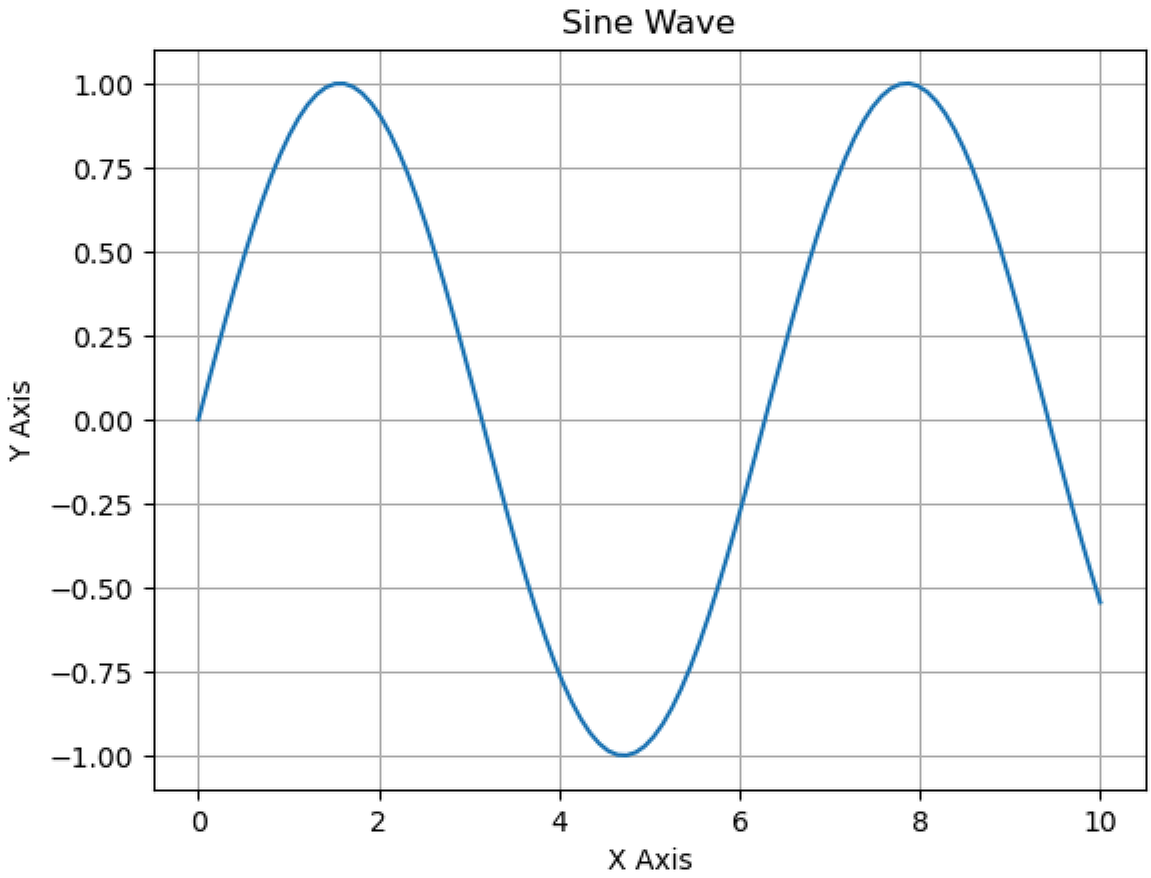
```
Mean: 30.0
Median: 30.0
```

In [8]:

```
import matplotlib.pyplot as plt

x = np.linspace(0, 10, 100)
y = np.sin(x)

plt.plot(x, y)
plt.title("Sine Wave")
plt.xlabel("X Axis")
plt.ylabel("Y Axis")
plt.grid(True)
plt.show()
```



What is a kernel?

A kernel is the computational engine that executes the code in your notebook.

What output formart to use when?

- Use PDF for formal submission or print
- Use HTML for interactive viewing online
- Use Slides when presenting to others

The NBviewer link:

Nbviewer link

Interactive widgets

Widgets make your notebook interactive, allowing input without editing code directly.

In [9]:
```
import ipywidgets as widgets
widgets.IntSlider(min=1, max=10, step=1, value=5)
```

Out[9]:
```
IntSlider(value=5, max=10, min=1)
```

In [10]:

```
from IPython.display import display

def update_plot(n):
    x = np.linspace(0, 10, 100)
    y = np.sin(n * x)
    plt.plot(x, y)
    plt.title(f"Sine Wave with frequency {n}")
    plt.show()

widgets.interact(update_plot, n=widgets.IntSlider(min=1, max=10, step=1, value=5));
```

```
interactive(children=(IntSlider(value=5, description='n', max=10, min=1), Output()),
_dom_classes=('widget-int…
```

In [ ]: