



# Summer Coding School 2023

Sept 2, 2023

**Taehee Jeong, Ph.D.**

# IF/Else statement

```
1 a = True
2 if a:
3     print('It is true!')
4     print('Also print this')
5 else:
6     print('It is false!')
7 print('Always print this')
```

It is true!

Also print this

Always print this

# For loop

```
1 a = [1,2,3,4,5]
2 for number in a:
3     print(number)
```

1  
2  
3  
4  
5

# While loop

```
1 a = 0
2 while a < 5:
3     print(a)
4     a = a + 1
```

0  
1  
2  
3  
4

# Functions with 'def'

- Function can be defined with 'def'
- Function name
- Function argument or parameter
- indent

```
>>> def greet(lang):  
...     if lang == 'es':  
...         print('Hola')  
...     elif lang == 'fr':  
...         print('Bonjour')  
...     else:  
...         print('Hello')  
...  
>>> greet('en')  
Hello  
>>> greet('es')  
Hola  
>>> greet('fr')  
Bonjour
```

# Return Value

A “fruitful” function is one that produces a result (or return value)

The return statement ends the function execution and “sends back” the result of the function

```
>>> def greet(lang):  
...     if lang == 'es':  
...         return 'Hola'  
...     elif lang == 'fr':  
...         return 'Bonjour'  
...     else:  
...         return 'Hello'  
...  
>>> print(greet('en'), 'Glenn')  
Hello Glenn  
>>> print(greet('es'), 'Sally')  
Hola Sally  
>>> print(greet('fr'), 'Michael')  
Bonjour Michael
```

# Multiple Parameters / Arguments

We can define more than one parameter in the function definition

We simply add more arguments when we call the function

We match the number and order of arguments and parameters

```
>>> def addtwo(a, b):  
>>>     added = a + b  
>>>     return added
```

```
>>> x = addtwo(3, 5)  
>>> print(x)  
>>> 8
```

# Class

```
1 class Dog:
2     _legs = 4
3     def __init__(self, name):
4         self.name = name
5
6     def getLegs(self):
7         return self._legs
8
9     def speak(self):
10        print(self.name + ' says: Bark!')
11
```

```
1 myDog = Dog('Rover')
2 print(myDog.name)
3 print(myDog.getLegs())
```

Rover  
4

```
1 myDog._legs = 3
2 print(myDog.name)
3 print(myDog.getLegs())
4 print(Dog._legs)
```

Rover  
3  
4



# Today's Agenda

- Data Visualization

# Matplotlib

- Matplotlib is an amazing visualization library in Python for 2D plots of arrays.
- Matplotlib consists of several plots like line, bar, scatter, histogram etc.
- Pyplot is a Matplotlib module that provides functions that interact with the figure i.e. creates a figure, decorates the plot with labels, and creates a plotting area in a figure.
- `python -mpip install -U matplotlib`
- `import matplotlib.pyplot as plt`

# line plot

```
# x-axis values
```

```
x = [5, 2, 9, 4, 7]
```

```
# Y-axis values
```

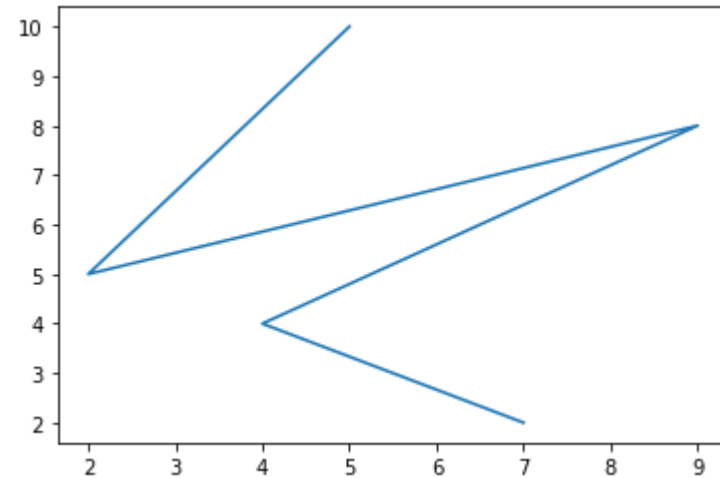
```
y = [10, 5, 8, 4, 2]
```

```
# Function to plot
```

```
plt.plot(x,y)
```

```
# function to show the plot
```

```
plt.show()
```



# scatter plot

```
# x-axis values
```

```
x = [5, 2, 9, 4, 7]
```

```
# Y-axis values
```

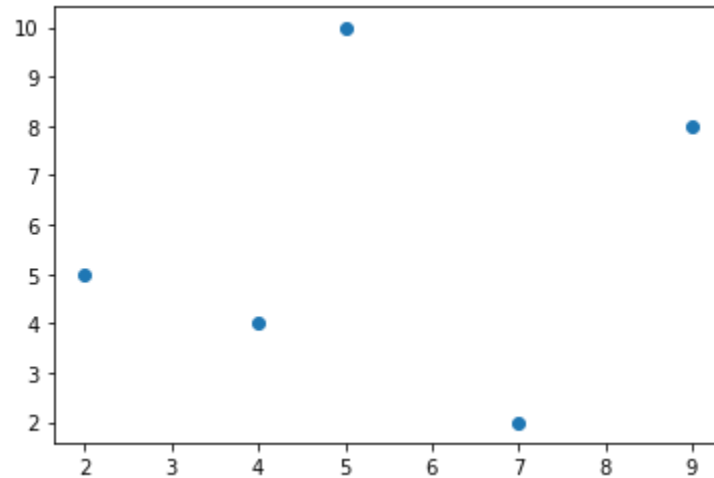
```
y = [10, 5, 8, 4, 2]
```

```
# Function to plot
```

```
plt.scatter(x,y)
```

```
# function to show the plot
```

```
plt.show()
```



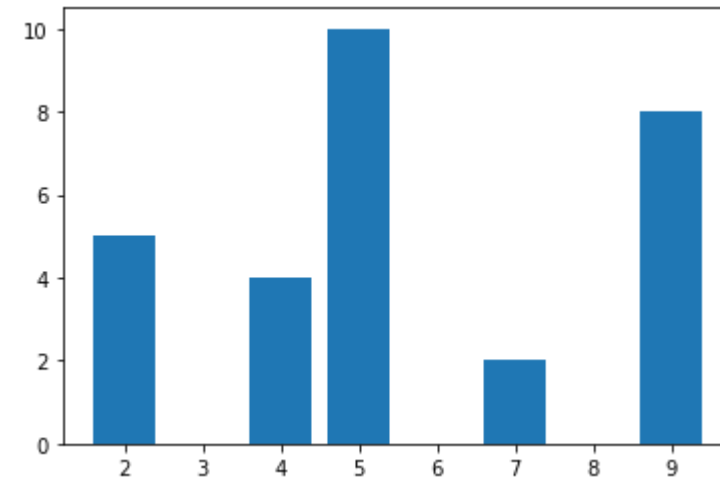
# Bar plot

```
# x-axis values
x = [5, 2, 9, 4, 7]

# Y-axis values
y = [10, 5, 8, 4, 2]

# Function to plot
plt.bar(x,y)

# function to show the plot
plt.show()
```



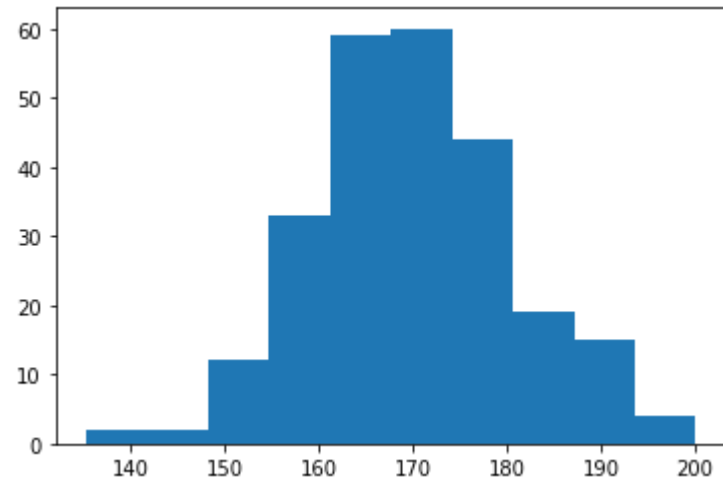
# histogram

```
# importing numpy module
import numpy as np

# values
x = np.random.normal(170, 10, 250)

# Function to plot
plt.hist(x)

# function to show the plot
plt.show()
```



# Color Reference

Color Syntax	Description
'r'	Red
'g'	Green
'b'	Blue
'c'	Cyan
'm'	Magenta
'y'	Yellow
'k'	Black
'w'	White

# Line Reference

Line Syntax	Description
'_'	Solid line
':'	Dotted line
'--'	Dashed line
'-.'	Dashed/dotted line



# Marker Reference

Marker Syntax	Description
'o'	Circle
'*'	Star
'.'	Point
','	Pixel
'x'	X
'+'	Plus
's'	Square
'd'	Diamond

Marker Syntax	Description
'p'	Pentagon
'h'	Hexagon
'v'	Triangle Down
'^'	Triangle Up
'<'	Triangle Left
'>'	Triangle Right

# Adding markers

```
# x-axis values
```

```
x = [5, 2, 9, 4, 7]
```

```
# Y-axis values
```

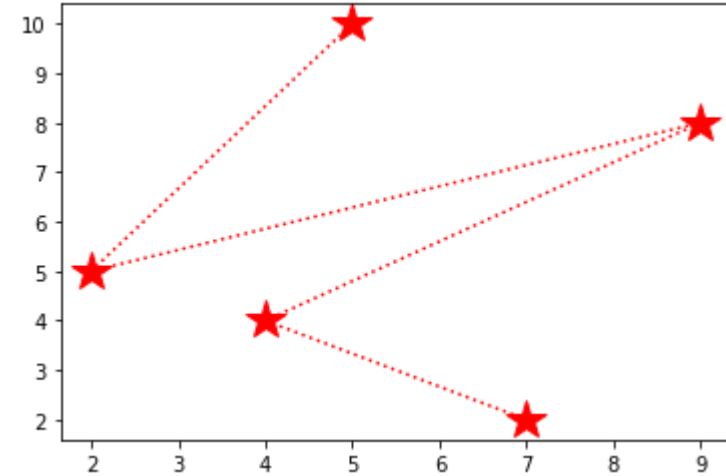
```
y = [10, 5, 8, 4, 2]
```

```
# Function to plot
```

```
plt.plot(x,y, color='r', linestyle = 'dotted', marker='*', ms=20)
```

```
# function to show the plot
```

```
plt.show()
```



# Adding title and labels

```
# x-axis values
x = [5, 2, 9, 4, 7]

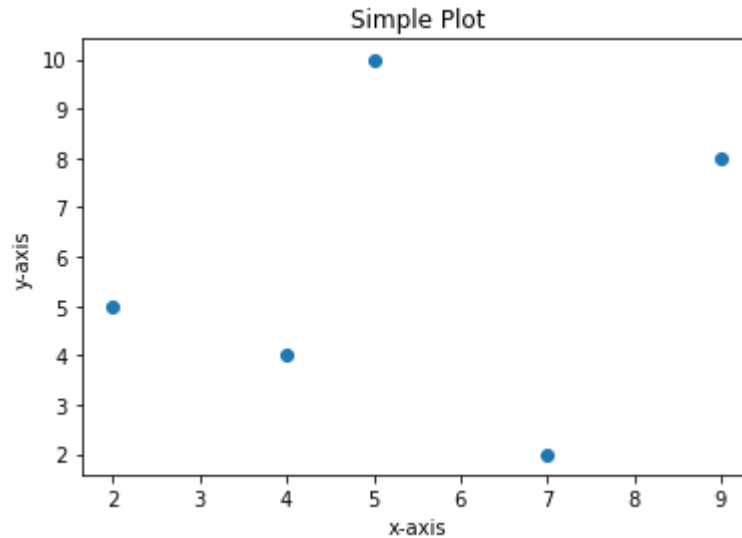
# Y-axis values
y = [10, 5, 8, 4, 2]

# Function to plot
plt.scatter(x,y)

# Adding the title
plt.title("Simple Plot")

# Adding the labels
plt.ylabel("y-axis")
plt.xlabel("x-axis")

# function to show the plot
plt.show()
```



# Adding grid

```
# x-axis values
x = [5, 2, 9, 4, 7]

# Y-axis values
y = [10, 5, 8, 4, 2]

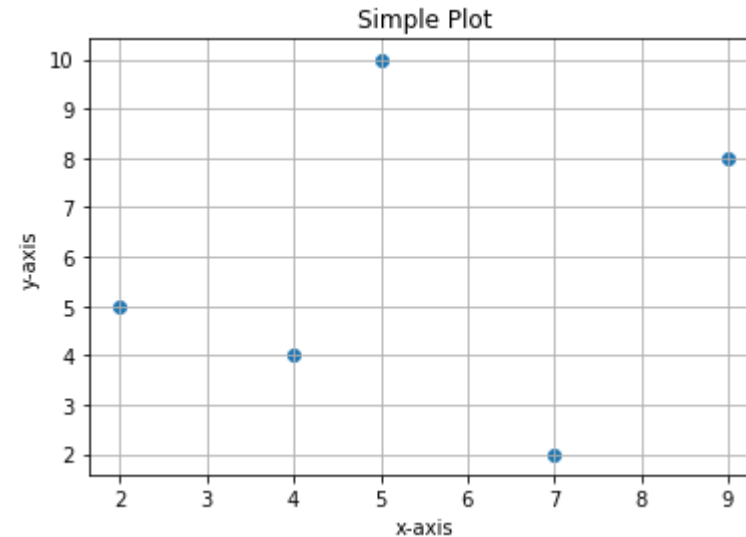
# Function to plot
plt.scatter(x,y)

# Adding the title
plt.title("Simple Plot")

# Adding the labels
plt.ylabel("y-axis")
plt.xlabel("x-axis")

# Adding the grid
plt.grid()

# function to show the plot
plt.show()
```



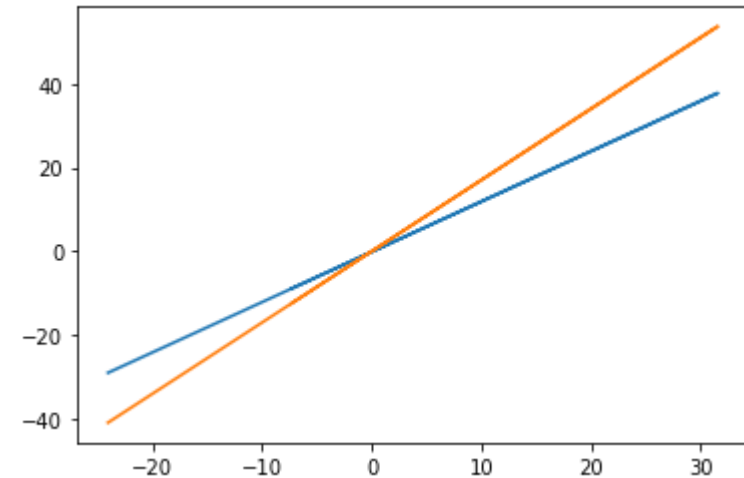
# Multiple plots

```
# importing modules
import matplotlib.pyplot as plt
import numpy as np

# generate sample data
x = np.random.normal(1, 10, 250)
y1 = 1.2 * x + 0.1* np.random.normal(0,1)
y2 = 1.7 * x + 0.1* np.random.normal(0,1)

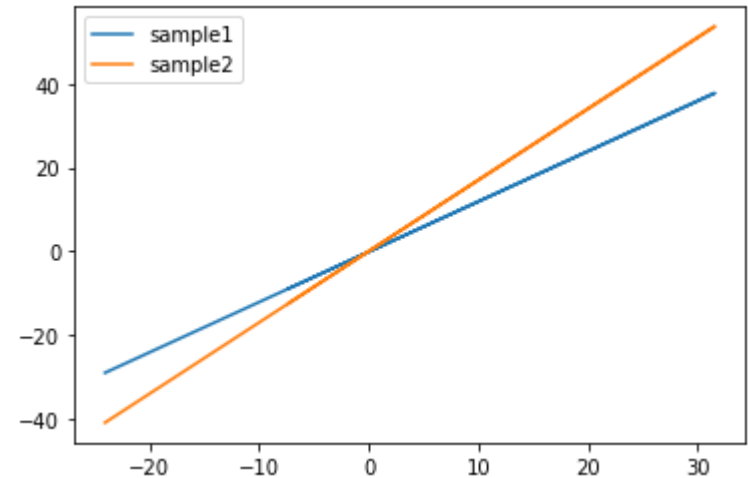
# Function to plot
plt.plot(x,y1)
plt.plot(x,y2)

# function to show the plot
plt.show()
```



# Adding legends

```
# Function to plot  
plt.plot(x,y1, label='sample1')  
plt.plot(x,y2, label='sample2')  
  
# adding legends  
plt.legend()  
  
# function to show the plot  
plt.show()
```



# Adding labels and grid

```
# Function to plot
```

```
plt.plot(x,y1, label='sample1')
```

```
plt.plot(x,y2, label='sample2')
```

```
# adding legends
```

```
plt.legend()
```

```
# Adding the labels
```

```
plt.ylabel("y-axis")
```

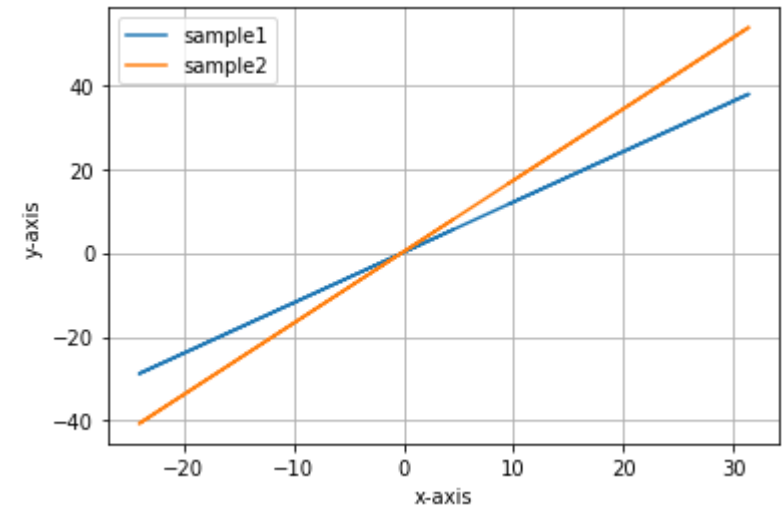
```
plt.xlabel("x-axis")
```

```
# Adding the grid
```

```
plt.grid()
```

```
# function to show the plot
```

```
plt.show()
```



# Interactive mode

```
%matplotlib
```

```
# Function to plot
```

```
plt.plot(x,y1, label='sample1')
```

```
plt.plot(x,y2, label='sample2')
```

```
# adding legends
```

```
plt.legend()
```

```
# Adding the labels
```

```
plt.ylabel("y-axis")
```

```
plt.xlabel("x-axis")
```

```
# Adding the grid
```

```
plt.grid()
```

```
# function to show the plot
```

```
plt.show()
```

