PYTHON NOTES: V25.01.08

Print() Function:

In [3]:
```
print("Hello Pratik")
```

```
Hello Pratik
```

Input() Function:

In [44]:
```
value = input("Enter a Value: ")
print(value)
```

```
Enter a Value:  23
23
```

Type() Function:

In [49]:
```
print(type(24))
type(24)
```

```
<class 'int'>
```

Out[49]:
```
int
```

Data Types: V25.01.08

1. Numeric Data Type

In [4]:
```
print("Numeric Data Type")
print(type(23))
print(type(23.5))
print(type(2+2j))
```

```
Numeric Data Type
<class 'int'>
<class 'float'>
<class 'complex'>
```

2. Sequential Data Type

In [23]:
```
print("Sequential Data Type")
print(type([22,22.3,2+2j]))
print(type((22,22.3,2+2j)))
print(type(set("aabbccdd")))
print(type({"name":"Pratik" ,"age":22}))
```

```
Sequential Data Type
<class 'list'>
<class 'tuple'>
<class 'set'>
<class 'dict'>
```

3. String Data Type

In [6]:
```
print("String Data Type")
print(type("Pratik"))
```

```
<class 'str'>
```

4. Boolean Data Type

In [7]:
```
print("Boolean Data Type")
print(type(True))
```

```
<class 'bool'>
```

Operators: V25.01.09 - V25.01.10

1. Assignment Arithmatic Operator

In [2]:
```
print("Assignment Arithmatic Operator")
num1 = int(input("Enter 1st Number: "))
num2 = int(input("Enter 2nd Number: "))
num1+=num2
print("Addition of two Number is: ",num1)
num1-=num2
print("Subtraction of two Number is: ",num1)
num1*=num2
print("Multiplication of two Number is: ",num1)
num1/=num2
print("Division of two Number is: ",num1)
num1//=num2
print("Floor Division of two Number is: ",num1)
num1**=num2
print("Power of two Number is: ",num1)
```

```
Assignment Arithmatic Operator
Addition of two Number is:  4
Subtraction of two Number is:  2
Multiplication of two Number is:  4
Division of two Number is:  2.0
Floor Division of two Number is:  1.0
Power of two Number is:  1.0
```

2. Relational Operator

In [28]:
```
print("Relational Operator")
print(num1>=num2)
print(num1<=num2)
print(num1==num2)
print(num1!=num2)
```

```
Enter 1st Number:  7
Enter 2nd Number:  5
True
False
True
False
False
True
```

3. Logical Operator V25.01.10

In [1]:
```
print("Logical Operator")
print(44>=33 and 44>=22)
print(22<=22 or 33<=33)
print(not True)
```

```
Logical Operator
True
True
False
```

4. Bitwise Operator

In [46]:
```
print("Bitwise Operator")
print(bin(15))
print(1 & 1)
print(15 | 11)
print(15 ^ 11)
print(~11)
print(bin(15>>2))
print(bin(15<<2))
print(15>>2)
print(15<<2)
```

```
0b1111
11
15
4
-12
0b11
0b111100
3
60
```

5. Membership Operator

In [2]:
```
print("Membership Operator:")
a = "ABCD"
print("A" in a)
```

```
Membership Operator:
True
```

6. `Identity Operator`

In [4]:
```
print("Identity Operator:")
a = "ABCD"
b = "ABCD"
print(a is b)
```

```
Identity Operator:
True
```

Jupyter Installation: V25.01.13

Control Statement: V25.01.14

1. `IF Statement`

In [4]:
```
if True:
    print("Welcome Guys!!")
```

```
Welcome Guys!!
```

In [9]:
```
if False:
    print("Welcome Guys!!")
    # Blanck!!  Print Nothing.
```

In [10]:
```
if 1:
    print("Welcome Guys!!")
    # Truth Value
```

```
Welcome Guys!!
```

In [14]:
```
if 0:
    print("Welcome Guys!!")
    # Blanck!!  Print Nothing. Falsy Value
```

In [16]:
```
if -0:
    print("Welcome Guys!!")
    # Blanck!!
    # Print Nothing.
    # Falsy Value
    # Entity : Something that exist.
    # -0 : Technicaly Wrong.
```

2. `If-Else Statement`

In [26]:
```
if True:
    print("Welcome Guys!!")
else:
    print("Bhaag...")
```

```
Welcome Guys!!
```

- `Question:` Accept a no from user and check if it is Even or Odd.

In [22]:
```
print("Print Even or Odd Numbers.")
num = int(input("Enter a Number: "))
if num%2 == 0 :
    print(num," is a Even Number.")
else:
    print(num," is a Odd Number.")
```

```
Print Even or Odd Numbers.
Enter a Number:  5
5  is a Odd Number.
```

- `Question:` Accept a no from user and check if it is divisible by 7 or 13.

In [25]:
```
print("Check a number divisible by 7 or 13.")
```

```
num = int(input("Enter a Number: "))
if num%7 == 0 or num%13 ==0 :
    print(num," is divisible.")
else:
    print(num," is not divisible.")
```

```
Check a number divisible by 7 or 13.
Enter a Number:  14
14  is divisible.
```

3. If-Elif Statement | If-Elif Ladder

- `Question:` Accept a no from user and check if it is Greater, Smaller or Equal to 23.

In [30]:
```
print("Check a Number is Greater, Smaller or Equal.")
num = int(input("Enter a Number: "))
if num>23:
    print(num," is a Greater than 23.")
elif num==23:
    print(num," is a Equal to 23.")
else:
    print(num," is a Smaller than 23.")
```

```
Check a Number is Greater, Smaller or Equal.
Enter a Number:  23
23  is a Equal to 23.
```

- `Question:` Accept a Budget from User.

In [35]:
```
budget = int(input("Enter your Budget: "))
if budget>=300 and budget<=500:
    print("International Trip.")
elif budget>=200 and budget<=299:
    print("National Trip.")
elif budget>=100 and budget<=199:
    print("City Trip.")
elif budget>=50 and budget<=99:
    print("Local Trip.")
elif budget>=1 and budget<=49:
    print("Gully Trip.")
else:
    print("Undefine Budget.")
```

```
Enter your Budget:  1
Gully Trip.
```

- `Question:` Accept a Designation from User and print is they apply for senior Position.

In [2]:
```
designation = input("Enter what Designationyou are Applying? Manager or Executive")
if "Manager" == designation:
    senior = input("Are you  applying for Senior Position? Yes or No:")
    if "yes" == senior:
        print("2Lakh Sallary.")
    else:
        print("1Lakh Sallary.")
elif "Executive" == designation:
    senior = input("Are you applying for Senior Position? Yes or No:")
    if "yes" == senior:
        print("70K Sallary.")
    else:
        print("25K Sallary.")
else:
    print("Undefined Designation.")
```

```
25K Sallary.
```

4. `Match Case`

In [15]:
```
value = 3
match value:
    case 1:
        print("Label 1")
    case 2:
        print("Label 2")
    case 3:
        print("Label 3")
    case _:
        print("Default Case ")
```

```
Label 3
```

Loops: V25.01.15

For Loop - Repetition

In [7]:
```
for i in range(1,10):
    print(i)
```

```
1
2
3
4
5
6
7
8
9
```

In [2]:
```
for i in range(11):
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
10
```

In [5]:
```
for i in range(1,11):
    print(i)
```

```
1
2
3
4
5
6
7
8
9
10
```

In [14]:
```
for i in range(1,11):
    print(i, end=" ")
```

```
1 2 3 4 5 6 7 8 9 10
```

In [13]:
```
for i in range(20,41):
    print(i, end=" ")
```

```
20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
```

- Question: Print Even Numbers 1 to 100

In [12]:
```
for i in range(1,101):
    if i%2==0:
        print(i, end=" ")
```

```
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78
80 82 84 86 88 90 92 94 96 98 100
```

- Question: Display Number Divisible by 2 & 5 from 1 to 100.

In [19]:
```
for i in range(1, 101):
    if i%2==0 and i%5==0:
        print(i, end=" ")c
```

```
10 20 30 40 50 60 70 80 90 100
```

- Question: Display Number Divisible by 13,17 from 1 to 100.

In [19]:
```
for i in range(1, 101):
    if i%13==0 and i%17==0:
        print(i, end=" ")c
```

```
10 20 30 40 50 60 70 80 90 100
```

- `Patterns:`

- `Square Row Number Patterns:`
- `11111`
- `22222`
- `33333`
- `44444`
- `55555`

In [20]:
```
for i in range(1,6):
    for j in range(1,6):
        print(i, end=" ")
    print("")
```

```
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
5 5 5 5 5
```

- `Square Column Number Patterns:`
- `12345`
- `12345`
- `12345`
- `12345`
- `12345`

In [21]:
```
for i in range(1,6):
    for j in range(1,6):
        print(j, end=" ")
    print("")
```

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

- `Right Half Increasing Pyramid Column Number Patterns:`
- `1`
- `12`
- `123`
- `1234`
- `12345`

In [24]:
```
for i in range(1,6):
    for j in range(1,6):
        if i>=j:
            print(j, end=" ")
    print("")
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

- `Right Half Decreasing Pyramid Column Number Patterns:`
- `12345`
- `1234`
- `123`
- `12`
- `1`

In [62]:
```
for i in range(1, 6):
    for j in range(1, 6):
        print(j, end=" ")
    print("")
```

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

- Decreasing Pyramid Patterns:
- 12345
- 2345
- 345
- 45
- 5

In [25]:
```
for i in range(1,6):
    for j in range(1,6):
        if i<=j:
            print(j, end=" ")
    print("")
```

```
1 2 3 4 5
2 3 4 5
3 4 5
4 5
5
```

- Star Square Patterns:
- * * * * *
- * * * * *
- * * * * *
- * * * * *
- * * * * *

In [26]:
```
for i in range(1,6):
    for j in range(1,6):
        print("*", end=" ")
    print("")
```

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

- Right Half Pyramid Patterns:
- *
- * *
- * * *
- * * * *
- * * * * *

In [27]:
```
for i in range(1,6):
    for j in range(1,6):
        if i>=j:
            print("*", end=" ")
    print("")
```

```
*
* *
* * *
* * * *
* * * * *
```

In [13]:
```
for i in range(6):
    for j in range(i):
        print("*",end=" ")
    print()
```

```
*
* *
* * *
* * * *
* * * * *
```

- Left Half Pyramid Patterns:
- * * * * *
- * * * *

- `* * * *`
- `* * *`
- `* *`
- `*`

In [28]:
```
for i in range(1,6):
    for j in range(1,6):
        if i<=j:
            print("*", end=" ")
    print("")
```

```
* * * * *
* * * *
* * *
* *
*
```

In [14]:
```
for i in range(5):
    for j in range(5-i):
        print("*",end=" ")
    print()
```

```
* * * * *
* * * *
* * *
* *
*
```

- `Space & Half Pyrmid Pattern` `V25.01.26`
- `    *`
- `   **`
- `  ***`
- ` ****`
- `*****`

In [6]:
```
for i in range(1,6):
    for j in range(5,0,-1):
        if j>=i:
            print(" ",end="")
    for k in range(1,6):
        if k<=i:
            print("*",end="")
    print("")
```

```
    *
   **
  ***
 ****
*****
```

- `Space & Half Pyrmid Pattern`
- `*****`
- ` ****`
- `  ***`
- `   **`
- `    *`

In [7]:
```
for i in range(1,6):
    for j in range(1,6):
        if j<=i:
            print(" ",end="")
    for k in range(5,0,-1):
        if k>=i:
            print("*",end="")
    print("")
```

```
*****
 ****
  ***
   **
    *
```

- `Space & Pyramid Pattern`
- `    **`
- `   ****`
- `  ******`
- ` ********`

- **********

In [10]:
```
for i in range(1,6):
    for j in range(5,0,-1):
        if j>=i:
            print(" ",end="")
    for k in range(1,6):
        if k<=i:
            print("*",end="")
    for k in range(1,6):
        if k<=i:
            print("*",end="")
    print("")
```

```
    **
   ****
  ******
 ********
**********
```

- Space & Pyramid Pattern
-     *
-    ***
-   *****
- *******
- *********

In [11]:
```
for i in range(1,6):
    for j in range(5,0,-1):
        if j>=i:
            print(" ",end="")
    for k in range(1,6):
        if k<=i:
            print("*",end="")
    for k in range(2,6):
        if k<=i:
            print("*",end="")
    print("")
```

```
    *
   ***
  *****
 *******
*********
```

- Space & Pyramid Pattern
- **********
- ********
-   ******
-    ****
-     **

In [12]:
```
for i in range(1,6):
    for j in range(1,6):
        if j<=i:
            print(" ",end="")
    for k in range(5,0,-1):
        if k>=i:
            print("*",end="")
    for k in range(5,0,-1):
        if k>=i:
            print("*",end="")
    print("")
```

```
**********
 ********
  ******
   ****
    **
```

- Space & Pyramid Pattern
- *********
- *******
-   *****
-    ***
-     *

In [13]:
```
for i in range(1,6):
    for j in range(1,6):
        if j<=i:
            print(" ",end="")
    for k in range(5,0,-1):
        if k>=i:
            print("*",end="")
    for k in range(4,0,-1):
        if k>=i:
            print("*",end="")
    print("")
```

```
*********
 *******
  *****
   ***
    *
```

- Diamond Pattern
- `    *`
- `   ***`
- `  *****`
- ` *******`
- `*********`
- `*********`
- ` *******`
- `  *****`
- `   ***`
- `    *`

In [36]:
```
for i in range(1,6):
    for j in range(5,0,-1):
        if j>=i:
            print(" ",end="")
    for k in range(1,6):
        if k<=i:
            print("*",end="")
    for k in range(2,6):
        if k<=i:
            print("*",end="")
    print("")
for i in range(1,5):
    for j in range(0,5):
        if j<=i:
            print(" ",end="")
    for k in range(4,0,-1):
        if k>=i:
            print("*",end="")
    for k in range(3,0,-1):
        if k>=i:
            print("*",end="")
    print("")
```

```
    *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *
```

Function: V25.01.20

- Block of code which executes targeted Logic.

Non-paametrise Function

In [7]:
```
def Info():
    print("Hello I.T.")
Info()
```

```
Hello I.T.
```

In [8]:
```
def Name():
    print("Hello Pratik..!")
Info()
```

```
Hello I.T.
```

- `return keyword` : `Returns to Function Call Value.`

In [11]:
```
def Data():
    a = int(input("Enter a Number: "))
    return a;
Num = Data()
print("Enter Number is: ",Num)
```

```
Enter Number is:  55
```

- `Question: Accept two values from User and return there Addition, Subtraction, Multiplication, Division.`

In [13]:
```
def add():
    a = int(input("Enter 1'st Number: "))
    b = int(input("Enter 2'nd Number: "))
    return a+b;
result = add()
print("Addition of two Numbers is: ",result)
```

```
Addition of two Numbers is:  10
```

In [16]:
```
def sub():
    a = int(input("Enter 1'st Number: "))
    b = int(input("Enter 2'nd Number: "))
    return a-b;
result = sub()
print("Subtraction of two Numbers is: ",result)
```

```
Subtraction of two Numbers is:  1
```

In [17]:
```
def mul():
    a = int(input("Enter 1'st Number: "))
    b = int(input("Enter 2'nd Number: "))
    return a*b;
result = mul()
print("Multiplication of two Numbers is: ",result)
```

```
Multiplication of two Numbers is:  25
```

In [18]:
```
def div():
    a = int(input("Enter 1'st Number: "))
    b = int(input("Enter 2'nd Number: "))
    return a/b;
result = div()
print("Division of two Numbers is: ",result)
```

```
Division of two Numbers is:  1.0
```

In [19]:
```
def mod():
    a = int(input("Enter 1'st Number: "))
    b = int(input("Enter 2'nd Number: "))
    return a%b;
result = mod()
print("Modulos of two Numbers is: ",result)
```

```
Modulos of two Numbers is:  0
```

In [20]:
```
def expo():
    a = int(input("Enter 1'st Number: "))
    b = int(input("Enter 2'nd Number: "))
    return a**b;
result = expo()
print("Power of two Numbers is: ",result)
```

```
Power of two Numbers is:  25
```

- `break keyword is part of loop` -> `Terminates the Execution of Code.`

In [28]:
```
def loop1():
    for i in range(1,11):
        if i == 5:
            break
        else:
            print(i,end=" ")
loop1()
```

```
1 2 3 4
```

- `continue keyword is part of loop` -> `Omits Perticular Situation.`

In [23]:
```
def loop1():
    for i in range(1,11):
        if i == 5:
            continue
        else:
            print(i,end=" ")
loop1()
```

```
1 2 3 4 6 7 8 9 10
```

In [25]:
```
def loop1():
    for i in range(1,11):
        if i == 7:
            continue
        else:
            print(i,end=" ")
loop1()
```

```
1 2 3 4 5 6 8 9 10
```

In [30]:
```
def loop1():
    for i in range(1,11):
        if i == 7:
            print(i,end=" ")
            continue
        else:
            print(i,end=" ")
loop1()
```

```
1 2 3 4 5 6 7 8 9 10
```

In [36]:
```
def data():
    a = 5
    return a
print(data())
```

```
5
```

In [35]:
```
def data():
    a = 5
    return a * 5
print(data())
```

```
25
```

In [39]:
```
def data1():
    a = 5
    return a * 5
def data2():
    a = "Pratik"
    return a
print("Value of Int is:",data1(),"Value of String is:",data2())
# Single Python Code has Mutiple Funtions.
# function name should be Unique.
# Multiple Python Function can be call in single print function.
```

```
Value of Int is: 25 Value of String is: Pratik
```

`Parametrise Function`

- `Compatible - Same Data Type`
- `No. of Argument should Match with No. of Parameters.`

In [43]:
```
def add(a,b,c):
    return a+b+c
print("Addition of two Number is",add(12,13,14))
```

```
Addition of two Number is 39
```

In [44]:
```
def add(a,b,c):
    return a+b+c
# print("Addition of two Number is",add(12,13,14))
def add(a,b,c):
    return a+b+c
print("Addition of two Number is",add(12,13,14))
```

```
Addition of two Number is 39
```

In [48]:
```
def add(a,b):
    return a+b+c
```

```
# print("Addition of two Number is",add(12,13,14))
def add(a,b,c):
    return a+b+c
print("Addition of two Number is",add(12,13,14))
# Processor Executes Closer Function. jo sabse close hoga use execute karega. means here is : def add(a,b,c):
```

```
Addition of two Number is 39
```

In [49]:
```
def add(a,b):
    print("Block 1")
    return a+b+c
# print("Addition of two Number is",add(12,13,14))
def add(a,b,c):
    print("Block 2")
    return a+b+c
print("Addition of two Number is",add(12,13,14))
# Processor Executes Closer Function. jo sabse close hoga use execute karega. means here is : def add(a,b,c):
```

```
Block 2
Addition of two Number is 39
```

In [47]:
```
def add(a,b):
    print("Block 1")
    return a+b+c
# print("Addition of two Number is",add(12,13,14))
def add(a,b,c):
    print("Block 2")
    return a+b+c
print("Addition of two Number is",add(12,13))
# Creates Ambiguity : Multiple Functions has same name. Mis-match Function. priority has closest one Function.
# Processor Executes Closer Function. jo sabse close hoga use execute karega. means here is : def add(a,b):
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[47], line 8
      6     print("Block 2")
      7     return a+b+c
----> 8 print("Addition of two Number is",add(12,13))
      9 # Processor Executes Closer Function. jo sabse close hoga use execute karega. means here is : def add(a,
b):

TypeError: add() missing 1 required positional argument: 'c'
```

- Arithmatic Operatios

In [52]:
```
def add(num1,num2):
    return num1+num2;
num1 = int(input("Enter 1'st Number: "))
num2 = int(input("Enter 2nd Number: "))
print("Addition of two Number is ",add(num1,num2))
```

```
Addition of two Number is  25
```

In [53]:
```
def sub(num1,num2):
    return num1-num2;
num1 = int(input("Enter 1'st Number: "))
num2 = int(input("Enter 2nd Number: "))
print("Subtraction of two Number is ",sub(num1,num2))
```

```
Subtraction of two Number is  -1
```

In [56]:
```
def mul(num1,num2):
    return num1*num2;
num1 = int(input("Enter 1'st Number: "))
num2 = int(input("Enter 2nd Number: "))
print("Product of two Number is ",sub(num1,num2))
```

```
Multiplication of two Number is  25
```

In [57]:
```
def div(num1,num2):
    return num1/num2;
num1 = int(input("Enter 1'st Number: "))
num2 = int(input("Enter 2nd Number: "))
print("Questiont of two Number is ",div(num1,num2))
```

```
Divesion of two Number is  1.0
```

In [58]:
```
def floor(num1,num2):
    return num1//num2;
num1 = int(input("Enter 1'st Number: "))
num2 = int(input("Enter 2nd Number: "))
print("Base Integer value of two Number is ",floor(num1,num2))
```

```
Floor of two Number is  1
```

In [60]:
```
def mod(num1,num2):
    return num1%num2;
num1 = int(input("Enter 1'st Number: "))
num2 = int(input("Enter 2nd Number: "))
print("Remainder of two Number is ",mod(num1,num2))
```

```
Remainder of two Number is  0
```

In [62]:
```
def expo(num1,num2):
    return num1**num2;
num1 = int(input("Enter 1'st Number: "))
num2 = int(input("Enter 2nd Number: "))
print("Power of two Number is ",expo(num1,num2))
```

```
Power of two Number is  125
```

- `Bitwise Operatios`

In [67]:
```
print("Bitwise Operator")
print(bin(15))
print(1 & 1)
print(15 | 11)
print(15 ^ 11)
print(~11)
print(bin(15>>2))
print(bin(15<<2))
print(15>>2)
print(15<<2)
```

```
Bitwise Operator
0b1111
1
15
4
-12
0b11
0b111100
3
60
```

In [70]:
```
def aand(num1,num2):
    return num1 & num2;
num1 = int(input("Enter 1'st Number: "))
num2 = int(input("Enter 2nd Number: "))
result = aand(num1,num2)
print("And of two Number is ",result)
```

```
And of two Number is  0
```

In [71]:
```
def oor(num1,num2):
    return num1 | num2;
num1 = int(input("Enter 1'st Number: "))
num2 = int(input("Enter 2nd Number: "))
result = oor(num1,num2)
print("Or of two Number is ",result)
```

```
Or of two Number is  1
```

In [72]:
```
def exor(num1,num2):
    return num1 ^ num2;
num1 = int(input("Enter 1'st Number: "))
num2 = int(input("Enter 2nd Number: "))
result = exor(num1,num2)
print("Ex-Or of two Number is ",result)
```

```
Ex-Or of two Number is  3
```

In [73]:
```
def exor(num1):
    return ~num1;
num1 = int(input("Enter 1'st Number: "))
result = exor(num1)
print("Ex-Or of two Number is ",result)
```

```
Ex-Or of two Number is  -2
```

In [74]:
```
def lso(num1,num2):
    return num1 << num2;
num1 = int(input("Enter 1'st Number: "))
num2 = int(input("Enter 2nd Number: "))
result = lso(num1,num2)
print("Ex-Or of two Number is ",result)
```

```
Ex-Or of two Number is  60
```

In [75]:
```
def rso(num1,num2):
```

```
    return num1 >> num2;
num1 = int(input("Enter 1'st Number: "))
num2 = int(input("Enter 2nd Number: "))
result = rso(num1,num2)
print("Ex-Or of two Number is ",result)
```

```
Ex-Or of two Number is  3
```

In [79]:
```
def rso(num1,num2):
    return num1 >> num2;
num1 = int(input("Enter 1'st Number: "))
num2 = int(input("Enter 2nd Number: "))
result = rso(num1,num2)
print(bin(num1))
print("Ex-Or of two Number is ",bin(result))
print("Ex-Or of two Number is ",result)
```

```
0b1111
Ex-Or of two Number is  0b11
Ex-Or of two Number is  3
```

In [80]:
```
def lso(num1,num2):
    return num1 << num2;
num1 = int(input("Enter 1'st Number: "))
num2 = int(input("Enter 2nd Number: "))
result = lso(num1,num2)
print(bin(num1))
print("Ex-Or of two Number is ",bin(result))
print("Ex-Or of two Number is ",result)
```

```
0b1111
Ex-Or of two Number is  0b111100
Ex-Or of two Number is  60
```

OOP's: Object Oriented Programming V25.01.21

- Identation
- Class
- Constructor
- Class Constructor
- Method
- Object
- Refrence Variable
- Instance Variable
- Local Variable
- Self keyword
- Self parameter

- Relation between Object and Class:
  - Object is an instance of Class
  - Constructor: Use to Intialize the Object.
  - Object allocates memory by intiallizing class Constructor
  - Object defines the Behaviour of Class
  - class is the container of Data members and Method functions.
  - class is a Blueprint
  - More than one constructor allow in python with diffrent parameters.
  - def keyword use to define Constructor.
  - By using __init__(self) we create Constructor.
  - self parameter is mendatory for constructor and methods.
  - Constructor does not require object

In [2]:
```
class DataInformation:
    def Data(self):
        print("Python Programming!!")
obj = DataInformation()
obj.Data()
```

```
Python Programming!!
```

In [4]:
```
class Data:
    def Method1(self):
        print("Method1")
    def Method2(self,num1):
        print("Method2")
        print("num1 is ",num1)
    def Method3(self,num1,num2):
```

```
        print("Method3")
        print("num1 is ",num1)
        print("num2 is ",num2)
obj = Data()
obj.Method1()
obj.Method2(21)
obj.Method3(31,41)
```

```
Method1
Method2
num1 is  21
Method3
num1 is  31
num2 is  41
```

In [8]:
```
a = 1
while(a<11):
    print(a,end=" ")
    a=a+1
```

```
1 2 3 4 5 6 7 8 9 10
```

In [10]:
```
a = 10
while(a>0):
    print(a,end=" ")
    a=a-1
```

```
10 9 8 7 6 5 4 3 2 1
```

In [20]:
```
class Iteration:
    def Loop1(self):
        a = 1
        while(a<11):
            print(a,end=" ")
            a = a+1
    print()
    def Loop2(self):
        a = 10
        while(a>0):
            print(a,end=" ")
            a = a-1
obj = Iteration()
obj.Loop1()
print()
obj.Loop2()
```

```
1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 4 3 2 1
```

- `Armstrong Number: 153` `A Number whose Summation of Cube of Each Digit is the same Number.`

In [29]:
```
class Armstrong:
    def Evaluate(self):
        num = int(input("Enter a Number"))
        temp = num
        sum = 0
        while(num>0):
            rem = num % 10
            sum = sum + (rem*rem*rem)
            num = num // 10
        if sum == temp:
            print(sum,"is an Armstrong Number.")
        else:
            print(temp,"is not an Armstrong Number.")
obj = Armstrong()
obj.Evaluate()
```

```
121 is not an Armstrong Number.
```

- `Palindrome Number: 121` `A Number Whose Reverce is the same Number.`

In [33]:
```
class Palindrome:
    def Evaluate(self):
        num = int(input("Enter a Number: "))
        temp = num
        sum = 0
        while(num>0):
            rem = num % 10
            sum = (sum *10) + rem
            num = num // 10
        if sum == temp:
            print(sum, "is a Palindrom Number.")
        else:
```

```
            print(temp, "is not a Palindrome Number.")
obj = Palindrome()
obj.Evaluate()
```

```
232 is a Palindrom Number.
```

- `Factorial Number: 5!` `Consider fact = 1 and multiply with ( i ) till the range is satisfy`

In [41]:
```
class Factorial:
    def Evaluate(self):
        fact = 1
        num = int(input("Enter a Number: "))
        for i in range(1,num+1):
            fact = fact*i
        print("Factorial of ",num,"is ",fact)
obj = Factorial()
obj.Evaluate()
```

```
Factorial of  5 is  120
```

- `Fibonacci Series: 0 1 1 2 3 5 8` `Addition of Privious two Numbers.`

In [46]:
```
class Fibonacci:
    def Evaluate(self):
        p1 = 0
        p2 = 1
        num = int(input("Enter the Range: "))
        print(p1,"",p2," ",end="")
        for i in range(2,num):
            p3 = p1 + p2
            print(p3," ",end="")
            p1 = p2
            p2 = p3
obj = Fibonacci()
obj.Evaluate()
```

```
0  1  1  2  3  5  8  13  21  34
```

- `Constructor`

In [47]:
```
class Data:
    a = 10
    b = 20
    def __init__(self):
        print("Constructor..")
Data()
```

```
Constructor..
```

Out[47]:
```
 <__main__.Data at 0x271adbdd3d0>
```

In [70]:
```
class Data:
    num1 = 10
    num2 = 20
    def __init__(self):
        print("Constructor..")
    def Method1(self):
        print("Method1")
    def Method2(self,num1):
        print("Method2")
        print("num1 is: ",num1)
    def Method3(self,num1,num2):
        print("Method3")
        print("num1 is: ",num1)
        print("num2 is: ",num2)
obj = Data()
obj.Method1()
obj.Method2(30)
obj.Method3(40,50)
```

```
Constructor..
Method1
Method2
num1 is:  30
Method3
num1 is:  40
num2 is:  50
```

In [71]:
```
class Data:
    num1 = 10
    num2 = 20
    def __init__(self):
```

```
        print("Constructor..")
    def Method1(self):
        print("Method1")
    def Method2(self,num1):
        print("Method2")
        print("num1 is: ",num1)
    def Method3(self,num1,num2):
        print("Method3")
        print("num1 is: ",self.num1)
        print("num2 is: ",self.num2)
obj = Data()
obj.Method1()
obj.Method2(30)
obj.Method3(40,50)
```

```
Constructor..
Method1
Method2
num1 is:  30
Method3
num1 is:  10
num2 is:  20
```

In [72]:
```
class Data:
    num1 = 10
    num2 = 20
    def __init__(self):
        print("Constructor..")
    def Method1(self):
        self.Method2(30)
        print("Method_1")
    def Method2(self,num1):
        self.Method3(40,50)
        print("Method_2")
        print("num1 is: ",self.num1)
    def Method3(self,num1,num2):
        print("Method_3")
        print("num1 is: ",self.num1)
        print("num2 is: ",self.num2)
obj = Data()
obj.Method1()
```

```
Constructor..
Method_3
num1 is:  10
num2 is:  20
Method_2
num1 is:  10
Method_1
```

In [73]:
```
class Data:
    num1 = 10
    num2 = 20
    def __init__(self,num1,num2):
        print("Constructor..")
        print("num1 is: ",num1)
        print("num2 is: ",num2)
    def Method1(self):
        print("Method1")
    def Method2(self,num1):
        print("Method2")
        print("num1 is: ",num1)
    def Method3(self,num1,num2):
        print("Method3")
        print("num1 is: ",self.num1)
        print("num2 is: ",self.num2)
obj = Data(100,200)
obj.Method1()
obj.Method2(30)
obj.Method3(40,50)
```

```
Constructor..
num1 is:  100
num2 is:  200
Method1
Method2
num1 is:  30
Method3
num1 is:  10
num2 is:  20
```

In [74]:
```
class Data:
    num1 = 10
    num2 = 20
    def __init__(self,num1,num2):
```

```
        print("Constructor..")
        print("num1 is: ",self.num1)
        print("num2 is: ",self.num2)
    def Method1(self):
        print("Method1")
    def Method2(self,num1):
        print("Method2")
        print("num1 is: ",num1)
    def Method3(self,num1,num2):
        print("Method3")
        print("num1 is: ",self.num1)
        print("num2 is: ",self.num2)
obj = Data(100,200)
obj.Method1()
obj.Method2(30)
obj.Method3(40,50)
```

```
Constructor..
num1 is:  10
num2 is:  20
Method1
Method2
num1 is:  30
Method3
num1 is:  10
num2 is:  20
```

- `Lambda Function: V25.01.23`

In [1]:
```
value = lambda a: a+10
print(value(25))
```

```
35
```

In [4]:
```
a = lambda c,d,e: c*d*e
print(a(5,7,9))
```

```
315
```

- `Nested function (Inner Function):`

In [10]:
```
def Add(a,b):
    print("Addition is ",(a+b))
    def Sub(a,b):
        print("Subtraction is ",(a-b))
        def Mul(a,b,c):
            print("Multiplication is ",(a*b*c))
            def Div(a,b):
                print("Division is ",(a/b))
            Div(10,5)
        Mul(10,20,30)
    Sub(10,11)
Add(5,7)
```

```
Addition is  12
Subtraction is  -1
Multiplication is  6000
Division is  2.0
```

String & It's functions:

- `isalnum:`

In [14]:
```
data = "Python12345"
print(data.isalnum())
```

```
True
```

In [15]:
```
data = "Python 12345"
print(data.isalnum())
```

```
False
```

- `isalpha:`

In [16]:
```
data = "ABCD"
print(data.isalpha())
```

```
True
```

In [17]:
```
data = "1ABCD"
print(data.isalpha())
```

```
False
```

- `isupper:`

In [21]:
```
data = "ABCD"
print(data.isupper())
```

```
True
```

- `islower:`

In [23]:
```
data = "ABCD"
print(data.islower())
```

```
False
```

In [12]:
```
data = "pqrs"
print(data.islower())
```

```
True
```

In [11]:
```
data = "pqrs"
print(data.isupper())
```

```
False
```

- `isdigit:`

In [25]:
```
data = "1234"
print(data.isdigit())
```

```
True
```

In [13]:
```
data = "1234 "
print(data.isdigit())
```

```
False
```

In [27]:
```
data = "1234A"
print(data.isdigit())
```

```
False
```

- `Concatination:`

In [31]:
```
a = "hi,"
b = " How are you?"
result = a+b
print(result)
```

```
hi, How are you?
```

In [32]:
```
a = "hi,"
b = "How are you?"
result = a+" "+b
print(result)
```

```
hi, How are you?
```

- `join:`

In [35]:
```
s1 = " sad "
s2 = "Happy"
result = s1.join(s2)
print(result)
```

```
H sad a sad p sad p sad y
```

In [37]:
```
s1 = "sad"
s2 = " Happy "
result = s2.join(s1)
print(result)
```

```
s Happy a Happy d
```

- `lower:`

In [38]:
```
s3 = "HAPPY"
```

```
print(s3.lower())
```

```
happy
```

- `upper:`

In [39]:
```
s3 = "happy"
print(s3.upper())
```

```
HAPPY
```

In [40]:
```
s3 = "AbCd"
print(s3.upper())
```

```
ABCD
```

In [41]:
```
s3 = "AbCd"
print(s3.lower())
```

```
abcd
```

- `swapcase:`

In [42]:
```
s3 = "AbCd"
print(s3.swapcase())
```

```
aBcD
```

- `title:`

In [43]:
```
s3 = "coading is cool when you understand it."
print(s3.title())
```

```
Coading Is Cool When You Understand It.
```

- `Format Specifier:`

In [44]:
```
id = 731
name = "Pandu"
percentage = 73.9
print("Id is %d, name is %s, Percentage are %f"%(id,name,percentage))
```

```
Id is 731, name is Pandu, Percentage are 73.900000
```

In [45]:
```
id = 731
name = "Pandu"
percentage = 73.9
print("Id is %d, name is %s, Percentage are %f"%(name,id,percentage))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[45], line 4
      2 name = "Pandu"
      3 percentage = 73.9
----> 4 print("Id is %d, name is %s, Percentage are %f"%(name,id,percentage))

TypeError: %d format: a real number is required, not str
```

In [47]:
```
rollno = 731
name = "Pratik"
percentage = 80.0
print("Roll No is %d, Name is %s, Percentage are %f"%(rollno,name,percentage))
```

```
Roll No is 731, Name is Pratik, Percentage are 80.000000
```

- `replace:`

In [48]:
```
s3 = "Pandu is a bad boy."
print(s3.replace("bad","good"))
```

```
Pandu is a good boy.
```

- `find:`

In [53]:
```
s3 = "abcdefgh"
print(s3.find("b"))
```

```
1
```

In [55]:
```
s3 = "abcdefgh"
print(s3.find("abc"))
```

```
0
```

In [56]:
```
s3 = "abcdefgh"
print(s3.find("z"))
```

```
-1
```

- `index:`

In [57]:
```
s3 = "abcdefgh"
print(s3.index("c"))
```

```
2
```

In [58]:
```
s3 = "abcdefgh"
print(s3.index("z"))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[58], line 2
      1 s3 = "abcdefgh"
----> 2 print(s3.index("z"))

ValueError: substring not found
```

List: V25.01.24

- `append: Add the element at the end of the list.`

In [4]:
```
# append element: full element add
a = [22,23,"Python",4+9j]
a.append("hello")
print(a)
```

```
[22, 23, 'Python', (4+9j), 'hello']
```

In [2]:
```
# append list into the another list:
a = [22,23,"Python",4+9j]
a.append("hello")
print(a)
b = [22,23,"Python",4+9j]
a.append(b)
print(a)
```

```
[22, 23, 'Python', (4+9j), 'hello']
[22, 23, 'Python', (4+9j), 'hello', [22, 23, 'Python', (4+9j)]]
```

In [6]:
```
# append tuple into the list:
a = [22,23,"Python",4+9j]
a.append("hello")
print(a)
b = [22,23,"Python",4+9j]
a.append(b)
print(a)
c = (11,22,33,44)
a.append(c)
print(a)
```

```
[22, 23, 'Python', (4+9j), 'hello']
[22, 23, 'Python', (4+9j), 'hello', [22, 23, 'Python', (4+9j)]]
[22, 23, 'Python', (4+9j), 'hello', [22, 23, 'Python', (4+9j)], (11, 22, 33, 44)]
```

- `extend: merge new iteration into the current list.`

In [8]:
```
# extend list: means mearge with exissting list:
a = [22,23,"Python",4+9j]
a.append("hello")
print(a)
b = [22,23,"Python",4+9j]
a.append(b)
print(a)
c = (11,22,33,44)
a.append(c)
print(a)
d2 = ["happy", "sad","nuteral"]
a.extend(d2)
print(a)
```

```
[22, 23, 'Python', (4+9j), 'hello']
[22, 23, 'Python', (4+9j), 'hello', [22, 23, 'Python', (4+9j)]]
[22, 23, 'Python', (4+9j), 'hello', [22, 23, 'Python', (4+9j)], (11, 22, 33, 44)]
[22, 23, 'Python', (4+9j), 'hello', [22, 23, 'Python', (4+9j)], (11, 22, 33, 44), 'happy', 'sad', 'nuteral']
```

In [5]:
```
# extend with tuple:
a = [22,23,"Python",4+9j]
a.append("hello")
print(a)
b = [22,23,"Python",4+9j]
a.append(b)
print(a)
c = (11,22,33,44)
a.append(c)
print(a)
d2 = ["happy", "sad","nuteral"]
a.extend(d2)
print(a)
d3 = (55,66,77,88)
a.extend(d3)
print(a)
```

```
[22, 23, 'Python', (4+9j), 'hello']
[22, 23, 'Python', (4+9j), 'hello', [22, 23, 'Python', (4+9j)]]
[22, 23, 'Python', (4+9j), 'hello', [22, 23, 'Python', (4+9j)], (11, 22, 33, 44)]
[22, 23, 'Python', (4+9j), 'hello', [22, 23, 'Python', (4+9j)], (11, 22, 33, 44), 'happy', 'sad', 'nuteral']
[22, 23, 'Python', (4+9j), 'hello', [22, 23, 'Python', (4+9j)], (11, 22, 33, 44), 'happy', 'sad', 'nuteral', 55,
66, 77, 88]
```

In [14]:
```
# extend element: one by one single charecter in the element added in list
a = [22,23,"Python",4+9j]
a.append("hello")
print(a)
b = [22,23,"Python",4+9j]
a.append(b)
print(a)
c = (11,22,33,44)
a.append(c)
print(a)
d2 = ["happy", "sad","nuteral"]
a.extend(d2)
print(a)
a.extend("Bye")
print(a)
```

```
[22, 23, 'Python', (4+9j), 'hello']
[22, 23, 'Python', (4+9j), 'hello', [22, 23, 'Python', (4+9j)]]
[22, 23, 'Python', (4+9j), 'hello', [22, 23, 'Python', (4+9j)], (11, 22, 33, 44)]
[22, 23, 'Python', (4+9j), 'hello', [22, 23, 'Python', (4+9j)], (11, 22, 33, 44), 'happy', 'sad', 'nuteral']
[22, 23, 'Python', (4+9j), 'hello', [22, 23, 'Python', (4+9j)], (11, 22, 33, 44), 'happy', 'sad', 'nuteral', 'B'
, 'y', 'e']
```

- `insert: It is use to add element at specific index position inside a list.`
- `remove: It will remove 1'st occurance of the specific element.`

In [32]:
```
# insert: insert at specific index position
b2 = [9+8j,"Guido", "Coding", 4.7]
b2.insert(0,"Hello")
print(b2)
b2 = [9+8j,"Guido", "Coding", 4.7]
b2.insert(2,"Logic")
print(b2)
# remove: remove 1st occurance of the element
b2.remove("Guido")
print(b2)
b2.insert(3,"4.7")
print(b2)
b2.remove(4.7)
print(b2)
# pop: specific index position element will be poped out
print(b2.pop())
print(b2)
print(b2.pop(1))
print(b2)
# index: It will return the index value of the specific element
print(b2.index(9+8J))
print(b2.index("Coding"))
# count: it will return the no of occurance of the specific element
print(b2.count("Coding"))
b2.insert(1, "Coding")
b2.insert(4, "Coding")
print(b2.count("Coding"))
print(b2)
```

```
# reverse: reverse the order of your list
b2.reverse()
print(b2)
```

```
['Hello', (9+8j), 'Guido', 'Coding', 4.7]
[(9+8j), 'Guido', 'Logic', 'Coding', 4.7]
[(9+8j), 'Logic', 'Coding', 4.7]
[(9+8j), 'Logic', 'Coding', '4.7', 4.7]
[(9+8j), 'Logic', 'Coding', '4.7']
4.7
[(9+8j), 'Logic', 'Coding']
Logic
[(9+8j), 'Coding']
0
1
1
3
[(9+8j), 'Coding', 'Coding', 'Coding']
['Coding', 'Coding', 'Coding', (9+8j)]
```

In [41]:
```
# Sort in asending order:
b3 = [9,4,13,1,48,3,7,5]
b3.sort()
print(b3)
# sort in descending order: it is use to return list in descending order.
b3.sort(reverse=True)
print(b3)
# clear: return empty list. delete all element in the list.
# b3.clear()
# print(b3)
# copy: copy's list in another destination.
b4 = b3.copy()
print(b4)
```

```
[1, 3, 4, 5, 7, 9, 13, 48]
[48, 13, 9, 7, 5, 4, 3, 1]
[48, 13, 9, 7, 5, 4, 3, 1]
```

In [2]:
```
# append
a = [1,1.2,2+3j,"Python","Coding","Coding","Coding"]
a.append("Pratik")
print(a)

# extend
a.extend("Bye")
print(a)

# insert
a.insert(3,5)
print(a)

# remove
a.remove("Python")
print(a)

# pop
a.pop(3)
print(a)

# index
print(a.index(1))

# count
print(a.count("Coding"))

# reverse
a.reverse()
print(a)

# sort -> only Numbers in ascending order
b =[1,2,3,4,5,6,7,8,9,10]
b.sort()
print(b)

# sort in reverse -> reverse=True
b.sort(reverse=True)
print(b)

# copy
c = b.copy()
print(c)
```

```
[1, 1.2, (2+3j), 'Python', 'Coding', 'Coding', 'Coding', 'Pratik']
[1, 1.2, (2+3j), 'Python', 'Coding', 'Coding', 'Coding', 'Pratik', 'B', 'y', 'e']
[1, 1.2, (2+3j), 5, 'Python', 'Coding', 'Coding', 'Coding', 'Pratik', 'B', 'y', 'e']
[1, 1.2, (2+3j), 5, 'Coding', 'Coding', 'Coding', 'Pratik', 'B', 'y', 'e']
[1, 1.2, (2+3j), 'Coding', 'Coding', 'Coding', 'Pratik', 'B', 'y', 'e']
0
3
['e', 'y', 'B', 'Pratik', 'Coding', 'Coding', 'Coding', (2+3j), 1.2, 1]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

- `Dictionary`

In [4]:

```python
data1 = {1:"Python", 2:"Class",3:4+9j,4:9.3,5:"Code"}
print(data1)

# get
print(data1.get(1))
print(data1.get(4))

# items -> Tuple Format
print(data1.items())

# keys -> Tuple Format
print(data1.keys())

# values -> Tuple Format
print(data1.values())

# a = (1,2,3)
# b = "India"
# data2.fromkeys(a,b)

# copy
data2 = data1.copy()
print(data1)
print(data2)

# update
data1.update({6:"Object"})
print(data1)
data1.update({1:"Guido"})
print(data1)

# pop -> by default remove the last key:value pair.
data1.pop(2)
print(data1)

# by default not remove without key
# data1.pop()
# print(data1)

# popitem -> not popitems
data1.popitem()
print(data1)

# clear
data1.clear()
print(data1)
```

```
{1: 'Python', 2: 'Class', 3: (4+9j), 4: 9.3, 5: 'Code'}
Python
9.3
dict_items([(1, 'Python'), (2, 'Class'), (3, (4+9j)), (4, 9.3), (5, 'Code')])
dict_keys([1, 2, 3, 4, 5])
dict_values(['Python', 'Class', (4+9j), 9.3, 'Code'])
{1: 'Python', 2: 'Class', 3: (4+9j), 4: 9.3, 5: 'Code'}
{1: 'Python', 2: 'Class', 3: (4+9j), 4: 9.3, 5: 'Code'}
{1: 'Python', 2: 'Class', 3: (4+9j), 4: 9.3, 5: 'Code', 6: 'Object'}
{1: 'Guido', 2: 'Class', 3: (4+9j), 4: 9.3, 5: 'Code', 6: 'Object'}
{1: 'Guido', 3: (4+9j), 4: 9.3, 5: 'Code', 6: 'Object'}
{1: 'Guido', 3: (4+9j), 4: 9.3, 5: 'Code'}
{}
```

`Inheritance: When The Object of child class aquires properties of object of parent class is called Inheritance. When One Child class aquires the properties of parent class is called Inheritance. V25.01.27`

- It is Object Oriented Concept.
- Parent Class: Parent calss|Child Class|Super Class.
- Child Class: Child Class|Derived Class|Sub Class.
- It is Not mendatory to always say object word in the types of Inheritance defination.

Types of Inheritance:

- Single Level Inheritance: When Object of One Child class aquries the properties of object of One Parent class.
- Multilevel Inheritance: When One child class is the parent of another child class such inheritance is called as multilevel inheritance
- Multiple Inheritance: When One child class inherited the properties of multiple parent class such inheritance is called as Multiple Inheritance.
- Hybrid Inheritance:

In [4]:
```
# Single Level Inheritance.
class Parent:
    def parent_method(self):
        print("Parent Method.")
class Child(Parent):
    def child_method(self):
        print("Child Method.")
obj = Child()
obj.child_method()
obj.parent_method()
print(Child.mro())
# By default Constructor call in python when object is created.
# Two Types Calling:
# 1. Child() -> by Default -> Implicit
# 2. obj.Child() -> By Own Hand -> Explicit
# Constructor creates Object(Memory) by default.
# By default Object created Constructor Implictly.
```

```
Child Method.
Parent Method.
[<class '__main__.Child'>, <class '__main__.Parent'>, <class 'object'>]
```

In [3]:
```
# Multilevel Inheritance.
class Pandu:
    def read(self):
        print("Reading..")
class Pandi(Pandu):
    def code(self):
        print("Coading..")
class Bablu(Pandi):
    def drive(self):
        print("Driving..")
class Babli(Bablu):
    def painting(self):
        print("Painting..")
obj = Babli()
obj.painting()
obj.drive()
obj.code()
obj.read()
print(Babli.mro())
# mro() -> Method Resolution Order. It is nothing but scequence of properties in Inheritance. -> Proritywise.
# Babli, Bunty, Pandi, Pandu, Object.
```

```
Painting..
Driving..
Coading..
Reading..
[<class '__main__.Babli'>, <class '__main__.Bablu'>, <class '__main__.Pandi'>, <class '__main__.Pandu'>, <class
'object'>]
```

In [1]:
```
# Another Example of Multilevel Inheritance.
class Add:
    def addition(self, num1, num2):
        print("Addition of two Numbers is",num1+num2)
class Sub(Add):
    def subtraction(self, num1, num2):
        print("Subtraction of two Numbers is",num1-num2)
class Mul(Sub):
    def mutiplication(self, num1, num2):
        print("Product of two Numbers is",num1*num2)
class Div(Mul):
    def division(self, num1, num2):
        print("Questiont of two Numbers is",num1/num2)
class Mod(Div):
```

```
    def modulos(self, num1, num2):
        print("Remainder of two Numbers is",num1%num2)
num1 = int(input("Enter First Number: "))
num2 = int(input("Enter Second Number: "))
obj = Mod()
obj.modulos(num1,num2)
obj.division(num1,num2)
obj.mutiplication(num1,num2)
obj.subtraction(num1,num2)
obj.addition(num1,num2)
print(Mod.mro())
# Mod, Div, Mul, Sub, Add, Object.
```

```
Remainder of two Numbers is 0
Questiont of two Numbers is 1.0
Product of two Numbers is 25
Subtraction of two Numbers is 0
Addition of two Numbers is 10
[<class '__main__.Mod'>, <class '__main__.Div'>, <class '__main__.Mul'>, <class '__main__.Sub'>, <class '__main_
_.Add'>, <class 'object'>]
```

In [5]:
```
# Multiple Inheritance.
class A:
    def m1(self):
        print("A")
class B:
    def m2(self):
        print("B")
class C:
    def m3(self):
        print("C")
class D(A,B,C):
    def m4(self):
        print("D")
obj = D()
obj.m4()
obj.m3()
obj.m2()
obj.m1()
print(D.mro())
# D, A, B, C, Object.
```

```
D
C
B
A
[<class '__main__.D'>, <class '__main__.A'>, <class '__main__.B'>, <class '__main__.C'>, <class 'object'>]
```

- Diffreance Between Multilevel and Multiple Inheritance.

- `Super Keyword:`

1. It is keyword.
2. Super Keyword Access the Imediate parent class properties.
3. It is Time Efficient.
4. Act's as an object of parent class.

In [8]:
```
# Accessing the Element of Class B element from Class C and class A element from class B.
class A:
    def m1(self):
        print("A")
class B(A):
    def m2(self):
        print("B")
        super().m1()
        # Call Imediate Parent Class.
class C(B):
    def m3(self):
        print("C")
        super().m2()
        # Call Imediate Parent Class.
obj = C()
obj.m3()
print(D.mro())
```

```
C
B
A
[<class '__main__.D'>, <class '__main__.A'>, <class '__main__.B'>, <class '__main__.C'>, <class 'object'>]
```

In [9]:
```
# Accessing the Class A Elements from the Class C.
```

```
class A:
    def m1(self):
        print("A")
class B(A):
    def m2(self):
        print("B")
class C(B):
    def m3(self):
        print("C")
        super().m1()
        # C->A
        # Call Imediate Parent Class. Still Implimented this Concept Beacouse,
        # It Accessing the elements of Parent calss B which is inherited from the Parent calss A.
        # Beacouse of Inheritance Still it can access the m1 method into the B class.
obj = C()
obj.m3()
print(D.mro())
# C, A, Object.
```

```
C
A
[<class '__main__.D'>, <class '__main__.A'>, <class '__main__.B'>, <class '__main__.C'>, <class 'object'>]
```

- `Call Imediate Parent Class. Still Implimented this Concept Beacouse, It Accessing the elements of Parent calss B which is inherited from the Parent calss A. Beacouse of Inheritance Still it can access the m1 method into the B class.`

Test V25.01.29

In [79]:
```
# Armstrong
class Armstrong:
    def armstrong(self):
        num = int(input("Enter a Armstrong Number: "))
        temp = num
        sum = 0
        while(num>0):
            rem = num % 10
            sum = sum + (rem * rem *rem)
            num = num//10
        if sum == temp:
            print(sum," is Armstrong Number.")
        else:
            print(temp, "is not armstrong Nnumber.")

# Palindrome
class Palindrome():
    def palindrome(self):
        num = int(input("Enter a Palindrome Number: "))
        temp = num
        sum = 0
        while(num>0):
            rem = num % 10
            sum = (sum * 10) + rem
            num = num // 10
        if sum == temp:
            print(sum,"is Palindrome Number.")
        else:
            print(temp,"is not Palindrome Number.")

# Factorial
class Factorial():
    def factorial(self):
        num = int(input("Enter a Factorial Number: "))
        fact = 1
        for i in range(1,num+1):
            fact = fact*i
        print("Factorial of ",num,"is",fact)

# Fibonacci
class Fibonacci():
    def fibonacci(self):
        num = int(input("Enter a Fibonacci Series Range: "))
        temp = num
        p1 = 0
        p2 = 1
        print(p1," ",p2,end="  ")
        for i in range(2,num-1):
            p3 = p1 + p2
            print(p3," ",end="")
            p1 = p2
```

```
            p2 = p3
        if num == temp:
            print(num,"is Palindrom Number.")
        else:
            print(temp,"is not palindrome number.")

# Multilevel Inheritance
class Evaluate(Armstrong, Palindrome, Factorial, Fibonacci):
    def evaluate(self):
        super().fibonacci()
        super().factorial()
        super().palindrome()
        super().armstrong()

obj = Evaluate()
obj.evaluate()
```

```
0   1  1  2  5 is Palindrom Number.
Factorial of  5 is 120
121 is Palindrome Number.
153  is Armstrong Number.
```

In [80]:

```
# Pyramid
class Pyramid:
    def pyramid(self):
        print("Pyramid: ")
        for i in range(1,6):
            for j in range(5,0,-1):
                if j>=i:
                    print(" ",end="")
            for k in range(1,6):
                if k<=i:
                    print("*",end="")
            for l in range(1,6):
                if l<=i-1:
                    print("*",end="")
            print()

# Reverse Pyramid
class ReversePyramid(Pyramid):
        def reversePyramid(self):
            super().pyramid()
            print()
            print("Reverse Pyramid: ")
            for i in range(1,6):
                for j in range(1,6):
                    if j<=i:
                        print(" ",end="")
                for k in range(5,0,-1):
                    if k>=i:
                        print("*",end="")
                for l in range(4,0,-1):
                    if l>=i:
                        print("*",end="")
                print()

# Diamond
class Diamond(ReversePyramid):
    def diamond(self):
        super().reversePyramid()
        print()
        print("Diamond: ")
        for i in range(1,6):
            for j in range(5,0,-1):
                if j>=i:
                    print(" ",end="")
            for k in range(1,6):
                if k<=i:
                    print("*",end="")
            for l in range(1,6):
                if l<=i-1:
                    print("*",end="")
            print()
        for i in range(1,5):
            for j in range(1,6):
                if j<=i+1:
                    print(" ",end="")
            for k in range(4,0,-1):
                if k>=i:
                    print("*",end="")
            for l in range(3,0,-1):
                if l>=i:
                    print("*",end="")
            print()
```

```
obj3 = Diamond()
obj3.diamond()
```

```
Pyramid:
     *
    ***
   *****
  *******
 *********

Reverse Pyramid:
 *********
  *******
   *****
    ***
     *

Diamond:
     *
    ***
   *****
  *******
 *********
  *******
   *****
    ***
     *
```

In [81]:

```
# List:
# 1. `append:`
# 2. `extend:`
# 3. `insert:`
# 4. `remove:`
# 5. `pop:`
# 6. `index:`
# 7. `count:`
# 8. `reverse:`
# 9. `sort:` & `sort:` `reverse=True`
# 10. `clear:`
# 11.  `copy:`

# String:
# 1. isalnum
# 2. isalpha
# 3. isdigit
# 4. islower
# 5. isupper
# 6. Concatination
# 7. join
# 8. upper
# 9. lower
# 10. title
# 11. swapcase
# 12. replace
# 13. index
# 14. find

class ListStringMethods1:
    def method1(self):

        print("List Functions:")
        data1 = [1,2,3,4,5]
        data2 = [6,7,8,9,10]

        data1.append(data2)
        print(data1)

        data1.extend(data2)
        print(data1)

        data1.insert(10,11)
        print(data1)

        print()
        print("String Functions:")
        name = "Pratik"
        print(name.isalnum())
        print(name.isalpha())
        print(name.isdigit())

class ListStringMethods2(ListStringMethods1):
    def method2(self):
```

```
        super().method1()
        print()

        print("List Functions:")
        data1 = [1,2,3,4,5]
        data2 = [6,7,8,9,10]

        print(data1.pop())

        print(data1.index(3))

        print(data1.count(3))

        print()
        print("String Functions:")
        name = "Pratik"
        print(name.isupper())
        print(name.islower())
        print(name +" " + "Majage")

class ListStringMethods3(ListStringMethods2):
    def method3(self):

        super().method2()
        print()

        print("List Functions:")
        data1 = [1,2,3,4,5]

        data1.remove(1)
        print(data1)

        data1.reverse()
        print(data1)

        data1.clear()
        print(data1)

        print()
        print("String Functions:")
        name = "pratik"
        print(name.upper())
        print(name.lower())
        print(name.title())

class ListStringMethods4(ListStringMethods3):
    def method4(self):

        super().method3()
        print()

        print("List Functions:")
        data1 = [1,2,3,4,5]

        data1.sort()
        print(data1)

        data1.sort(reverse=True)
        print(data1)

        data2 = data1.copy()
        print(data2)

        print()
        print("String Functions:")
        q1 = "Pratik is bad boy"
        print(q1.replace("bad","good"))
        w1 = "aabbccdd"
        print(w1.find("c"))
        print(w1.index("c"))
        name = "Pratik"
        string = "-"
        print(string.join(name))


obj = ListStringMethods4()
obj.method4()
```

```
List Functions:
[1, 2, 3, 4, 5, [6, 7, 8, 9, 10]]
[1, 2, 3, 4, 5, [6, 7, 8, 9, 10], 6, 7, 8, 9, 10]
[1, 2, 3, 4, 5, [6, 7, 8, 9, 10], 6, 7, 8, 9, 11, 10]

String Functions:
True
True
False

List Functions:
5
2
1

String Functions:
False
False
Pratik Majage

List Functions:
[2, 3, 4, 5]
[5, 4, 3, 2]
[]

String Functions:
PRATIK
pratik
Pratik

List Functions:
[1, 2, 3, 4, 5]
[5, 4, 3, 2, 1]
[5, 4, 3, 2, 1]

String Functions:
Pratik is good boy
4
4
P-r-a-t-i-k
```

- `Math Functions`

In [43]:
```python
# Telusko:
import math
print(math.sqrt(25))
print(math.floor(5.9))
print(math.ceil(5.1))
print(math.pow(3,2))
print(math.pi)
print(math.e)
# allice:
import math as m
print(m.sqrt(25))
from math import sqrt, pow
sqrt(9)
pow(4,5)
# help("math")
```

```
5.0
5
6
9.0
3.141592653589793
2.718281828459045
5.0
```

Out[43]: 1024.0

Regular Expression: V25.01.30

- `DateTime:`

In [35]:
```python
import datetime
a = datetime.datetime.now()
print(a) # current full date and time
print("----------------")
print(a.strftime("%a")) # shortform day
print(a.strftime("%A")) # fullform day
print("----------------")
print(a.strftime("%d")) # shortform date
```

```
print(a.strftime("%D")) # fullform day
print("----------------")
print(a.strftime("%b")) # shortform month
print(a.strftime("%B")) # fullform day
print("----------------")
print(a.strftime("%y")) # shortform year
print(a.strftime("%Y")) # fullform year
print("----------------")
print(a.strftime("%w")) # Scequence number of weeks
print(a.strftime("%W")) # Scequence number of weeks
print("----------------")
print(a.strftime("%m")) # month
print(a.strftime("%M")) # Minute
print("----------------")
print(a.strftime("%h")) # month
print(a.strftime("%H")) # Hours
print(a.strftime("%S")) # Seconds
print(a.strftime("%f")) # microseconds in 6 digits
print("----------------")
print(a.strftime("%c")) #day month date time year
print(a.strftime("%C"))
print(a.strftime("%V")) # week date start from sunday sometime in other calenders monday
print("----------------")
print(a.strftime("%p")) # PM | AM
print(a.strftime("%I")) # 12 Hours Format
print(a.strftime("%x")) # date
print(a.strftime("%X")) # Time
```

```
2025-02-08 12:36:36.515547
----------------
Sat
Saturday
----------------
08
02/08/25
----------------
Feb
February
----------------
25
2025
----------------
6
05
----------------
02
36
----------------
Feb
12
36
515547
----------------
Sat Feb  8 12:36:36 2025
20
06
----------------
PM
12
02/08/25
12:36:36
```

- `Meta Charecters:- (Regular Expression.)`

- `search()`

In [36]:
```
import re
data = "Pandu is good boy"
a = re.search("^Pandu",data)
print(a)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
<re.Match object; span=(0, 5), match='Pandu'>
Found.
```

In [37]:
```
import re
data = "Pandu is good boy"
a = re.search("boy$",data)
print(a)
```

```
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
<re.Match object; span=(14, 17), match='boy'>
Found.
```

In [38]:
```
import re
data = "Pandu is good boy"
a = re.search("^Pandu.*boy$",data)
print(a)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
<re.Match object; span=(0, 17), match='Pandu is good boy'>
Found.
```

- `foundall()`

In [39]:
```
import re
data = "No Pain No Gain"
a = re.findall("ain",data)
print(a)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
['ain', 'ain']
Found.
```

In [42]:
```
import re
data = "No Pain No Gain"
a = re.findall("Pain",data)
print(a)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
['Pain']
Found.
```

In [57]:
```
import re
data = "Hello Pandu, how are you?"
a = re.findall("H...o",data)
print(a)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
['Hello']
Found.
```

In [59]:
```
import re
data = "Hello Pandu, how are you?"
a = re.findall("H...p",data)
print(a)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
[]
Not Found.
```

In [60]:
```
import re
data = "Hello Pandu, how are you?"
a = re.findall("H..o",data)
print(a)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
[]
Not Found.
```

In [62]:
```
import re
data = "Hello Pandu, how are you?"
a = re.findall("H.*o",data)
```

```
print(a)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
['Hello Pandu, how are yo']
Found.
```

In [65]:
```
import re
data = "Hello Pandu, how are you?"
a = re.findall("H.{3}o",data)
print(a)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
['Hello']
Found.
```

In [66]:
```
import re
data = "Hello Pandu, how are you?"
a = re.findall("H.{4}o",data)
print(a)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
[]
Not Found.
```

In [67]:
```
import re
data = "Hello Pandu, how are you?"
a = re.findall("H.+o",data)
print(a)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
['Hello Pandu, how are yo']
Found.
```

In [69]:
```
import re
data = "Hello Pandu, how are you?"
a = re.findall("H.?l",data)
print(a)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
['Hel']
Found.
```

In [70]:
```
import re
data = "Hello Pandu, how are you?"
a = re.findall("H.?o",data)
print(a)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
[]
Not Found.
```

In [72]:
```
import re
data = "Hello Pandu, how are you?"
a = re.findall("Pandu|Pandi",data)
print(a)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
['Pandu']
Found.
```

In [73]:
```
import re
data = "Hello Pandu, how are you?"
a = re.findall("Bunty|Pandi",data)
print(a)
if a:
```

```
    print("Found.")
else:
    print("Not Found.")
```

```
[]
Not Found.
```

- `sub()`

In [50]:
```
import re
data = "No Pain No Gain"
a = re.sub("Pain","Power",data)
print(a)
```

```
No Power No Gain
```

- `split()`

In [49]:
```
import re
data = "No Pain No Gain"
a = re.split(" ",data)
print(a)
```

```
['No', 'Pain', 'No', 'Gain']
```

In [51]:
```
import re
data = "No Pain No Gain"
a = re.split("a",data)
print(a)
```

```
['No P', 'in No G', 'in']
```

In [53]:
```
import re
data = "No Pain No Gain"
a = re.split("",data,3)
print(a)
```

```
['', 'N', 'o', ' Pain No Gain']
```

- `Special Scequence:`

In [84]:
```
data = "No Pain No Gain"
a = re.findall("\ANo",data)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
Found.
<>:2: SyntaxWarning: invalid escape sequence '\A'
<>:2: SyntaxWarning: invalid escape sequence '\A'
C:\Users\majag\AppData\Local\Temp\ipykernel_7892\36357980.py:2: SyntaxWarning: invalid escape sequence '\A'
  a = re.findall("\ANo",data)
```

In [87]:
```
data = "No Pain No Gain"
a = re.findall(r"Gain\b",data)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
Found.
```

In [91]:
```
data = "No Pain No Gain"
a = re.findall(r"ain\b",data)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
Found.
```

In [90]:
```
data = "No Pain No Gain"
a = re.findall(r"\BGain",data)
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
Not Found.
```

In [92]:
```
data = "No Pain No Gain"
a = re.findall(r"\Bain",data)
```

```
if a:
    print("Found.")
else:
    print("Not Found.")
```

```
Found.
```

In [97]:
```
data1 = "abcd1234"
a = re.findall("\d",data1)
print(a)
```

```
['1', '2', '3', '4']
<>:2: SyntaxWarning: invalid escape sequence '\d'
<>:2: SyntaxWarning: invalid escape sequence '\d'
C:\Users\majag\AppData\Local\Temp\ipykernel_7892\2613430423.py:2: SyntaxWarning: invalid escape sequence '\d'
  a = re.findall("\d",data1)
```

In [98]:
```
data1 = "abcd1234"
a = re.findall("\D",data1)
print(a)
```

```
['a', 'b', 'c', 'd']
<>:2: SyntaxWarning: invalid escape sequence '\D'
<>:2: SyntaxWarning: invalid escape sequence '\D'
C:\Users\majag\AppData\Local\Temp\ipykernel_7892\2706297881.py:2: SyntaxWarning: invalid escape sequence '\D'
  a = re.findall("\D",data1)
```

In [99]:
```
data2 = "Pandu is Good Boy"
a = re.findall("\s",data2)
print(a)
```

```
[' ', ' ', ' ']
<>:2: SyntaxWarning: invalid escape sequence '\s'
<>:2: SyntaxWarning: invalid escape sequence '\s'
C:\Users\majag\AppData\Local\Temp\ipykernel_7892\4111360370.py:2: SyntaxWarning: invalid escape sequence '\s'
  a = re.findall("\s",data2)
```

In [100…
```
data2 = "Pandu is Good Boy"
a = re.findall("\S",data2)
print(a)
```

```
['P', 'a', 'n', 'd', 'u', 'i', 's', 'G', 'o', 'o', 'd', 'B', 'o', 'y']
<>:2: SyntaxWarning: invalid escape sequence '\S'
<>:2: SyntaxWarning: invalid escape sequence '\S'
C:\Users\majag\AppData\Local\Temp\ipykernel_7892\2385974374.py:2: SyntaxWarning: invalid escape sequence '\S'
  a = re.findall("\S",data2)
```

In [102…
```
data3 = "Pandu is Good Boy $*&!"
a = re.findall("\W",data3)
print(a)
```

```
[' ', ' ', ' ', ' ', '$', '*', '&', '!']
<>:2: SyntaxWarning: invalid escape sequence '\W'
<>:2: SyntaxWarning: invalid escape sequence '\W'
C:\Users\majag\AppData\Local\Temp\ipykernel_7892\2233658347.py:2: SyntaxWarning: invalid escape sequence '\W'
  a = re.findall("\W",data3)
```

In [103…
```
data3 = "Pandu is Good Boy $*&!"
a = re.findall("\w",data3)
print(a)
```

```
['P', 'a', 'n', 'd', 'u', 'i', 's', 'G', 'o', 'o', 'd', 'B', 'o', 'y']
<>:2: SyntaxWarning: invalid escape sequence '\w'
<>:2: SyntaxWarning: invalid escape sequence '\w'
C:\Users\majag\AppData\Local\Temp\ipykernel_7892\3781952730.py:2: SyntaxWarning: invalid escape sequence '\w'
  a = re.findall("\w",data3)
```

In [106…
```
data4 = "Gudi Van Rossum"
a = re.findall("Rossum\Z",data4)
print(a)
```

```
['Rossum']
<>:2: SyntaxWarning: invalid escape sequence '\Z'
<>:2: SyntaxWarning: invalid escape sequence '\Z'
C:\Users\majag\AppData\Local\Temp\ipykernel_7892\3953898632.py:2: SyntaxWarning: invalid escape sequence '\Z'
  a = re.findall("Rossum\Z",data4)
```

✖ V25.01.31 File Handling -> OFF

In [108…
```
obj = open("Pratik.txt","a+")
print(obj)
```

```
<_io.TextIOWrapper name='Pratik.txt' mode='a+' encoding='cp1252'>
```

In [109…
```
obj = open("Code.pdf","w")
```

In [110…
```
obj = open("Code2.xlsx","w")
```

In [111…
```
obj = open("Pratik.txt","r")
data = obj.read()
print(data)
```

```
Hi, I am Pratik.
It's pleasure to meet you.
love from Pune.
```

In [113…
```
obj = open("Pratik.txt","r")
data = obj.read()
print(data)
obj.close()
```

```
Hi, I am Pratik.
It's pleasure to meet you.
love from Pune.
```

In [114…
```
obj = open("Ramesh.txt","w")
```

In [133…
```
obj = open("Nupur.txt","w")
obj.write("Hello Nupur, How are you?")
obj.write("\nWhere are you heading?")
obj.close()
```

In [134…
```
obj = open("Nupur.txt","r")
data = obj.read()
print(data)
```

```
Hello Nupur, How are you?
Where are you heading?
```

In [135…
```
obj = open("Nupur.txt","r+")
obj.write("\nYou still love me?")
data = obj.read()
print(data)
obj.close()
```

```
 you?
Where are you heading?
```

In [138…
```
obj = open("DataFile.txt","w+")
obj.write("OOP")
obj.write("\nPractice..")
obj.write("\nScalable..")
obj.seek(0)
data = obj.read()
print(data)
obj.write("\nHigh Package..")
obj.seek(0)
data = obj.read()
print(data)
obj.seek(0)
obj.tell()
```

```
OOP
Practice..
Scalable..
OOP
Practice..
Scalable..
High Package..
```

Out[138… 0

In [140…
```
obj = open("DataFile.txt","a")
obj.write("\nInheritance..")
obj.close()
obj = open("DataFile.txt","r")
data = obj.read()
print(data)
```

```
OOP
Practice..
Scalable..
High Package..
Inheritance..
Inheritance..
```

In [143…
```
obj = open("DataFile.txt","a+")
obj.write("\nFile Handling..")
obj.write("\nAgain File Handling")
```

```
obj.seek(0)
data = obj.read()
print(data)
obj.close()
```

```
OOP
Practice..
Scalable..
High Package..
Inheritance..
Inheritance..
File Handling..
Again File Handling
File Handling..
Again File Handling
File Handling..
Again File Handling
```

Numpy V25.02.03

In [3]:
```
pip install numpy
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: numpy in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (1.26.4)
Note: you may need to restart the kernel to use updated packages.
[notice] A new release of pip is available: 24.2 -> 25.0
[notice] To update, run: C:\Users\majag\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip
```

In [4]:
```
!pip install numpy
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: numpy in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (1.26.4)
[notice] A new release of pip is available: 24.2 -> 25.0
[notice] To update, run: C:\Users\majag\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip
```

In [7]:
```
import numpy as np
# List in python
a = [1,2,3,4,5]
print("List: ", a)
print(type(a))
# Array in python
num  = np.array([1,2,3,4,5])
print("Array: ",num)
print(type(num))
```

```
List:  [1, 2, 3, 4, 5]
<class 'list'>
Array:  [1 2 3 4 5]
<class 'numpy.ndarray'>
```

- Dimentions in Array:

In [23]:
```
a = np.array([11])
print(a)
print(a.ndim)
```

```
[11]
1
```

- 1 Dimention | Uni-Dimention Array:

In [10]:
```
a = np.array([1,2,3,4,5])
print(a)
print(a.ndim) # Return Dimentions.
```

```
[1 2 3 4 5]
1
```

- 2-Dimention Array:

In [12]:
```
a = np.array([[1,2,3,4,5],[6,7,8,9,10]])
print(a)
print(a.ndim) # row (x) - column (y)
```

```
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]]
2
```

- 3-Dimention Array:

In [15]:
```
a = np.array([[[1,2,3],[4,5,6]],[[7,8,9],[11,12,13]],[[14,15,16],[17,18,19]]])
print(a)
print(a.ndim) # 3D (x)-(y)-(x)
```

```
[[[ 1  2  3]
  [ 4  5  6]]

 [[ 7  8  9]
  [11 12 13]]

 [[14 15 16]
  [17 18 19]]]
3
```

- Index And Length:

In [30]:
```
a = np.array([11,22,33,44,55])
print(a)
print(a.ndim)
print(a[0])
print(a[2])
print(a[-1])
```

```
[11 22 33 44 55]
1
11
33
55
```

In [40]:
```
a = np.array([[11,22,33],[44,55,66],[77,88,99]])
print(a)
print(a.ndim)
print(a[0])
print(a[2])
print(a[0,0])
print(a[1,1])
print(a[0,-1])
print(a[-1,0])
print(a[1,1]+a[2,2])
print(a[0,1]*a[1,2])
```

```
[[11 22 33]
 [44 55 66]
 [77 88 99]]
2
[11 22 33]
[77 88 99]
11
55
33
77
154
1452
```

In [7]:
```
import numpy as np
value = np.array([[[1,2,3],[4,5,6,]],[[10,20,30],[40,50,60]],[[25,35,45],[55,65,75]]])
print(value)
print(value.ndim)
print(value[0,0,0])
print(value[2,0,0])
print(value[-1,-1,-1])
```

```
[[[ 1  2  3]
  [ 4  5  6]]

 [[10 20 30]
  [40 50 60]]

 [[25 35 45]
  [55 65 75]]]
3
1
25
75
```

- Slicing:

In [15]:
```
a = np.array([17,18,19,20,21,22,23,24,25])
print(a)
```

```
print(a.ndim)
print(a[0:])
print(a[:7])
print(a[0:8:2]) #steps
print(a[0:8:3])
```

```
[17 18 19 20 21 22 23 24 25]
1
[17 18 19 20 21 22 23 24 25]
[17 18 19 20 21 22 23]
[17 19 21 23]
[17 20 23]
```

In [28]:
```
a = np.array([[11,12,13],[14,15,16],[17,18,19]])
print(a)
print(a.ndim)
print(a[0:])
print(a[0:2])
print(a[0:2],2) # 2 is normal value
print(a[1:3])
print(a[-1:])
```

```
[[11 12 13]
 [14 15 16]
 [17 18 19]]
2
[[11 12 13]
 [14 15 16]
 [17 18 19]]
[[11 12 13]
 [14 15 16]]
[[11 12 13]
 [14 15 16]] 2
[[14 15 16]
 [17 18 19]]
[[17 18 19]]
```

- `sort & searchsorted:`

In [38]:
```
a = np.array([22,29,48,56,1,3,9])
print(a)
print(a.ndim)
num = np.sort(a)
print(num)
value = np.searchsorted(num,7)
print(value) #Sorted value index number
value = np.searchsorted(num,43)
print(value)
num[5] = 43
print(num) # It will replace the value
```

```
[22 29 48 56  1  3  9]
1
[ 1  3  9 22 29 48 56]
2
5
[ 1  3  9 22 29 43 56]
```

Polymorphism And Exception Handling

Polymorphism

- `Method Overloading : Python does not support method overloading directly. However, we can achieve it using default arguments or variable-length arguments` (`*args` and `**kwargs`).

In [13]:
```
class Data:
    def m(self, num1=None, num2=None):
        if num1 is not None and num2 is not None:
            return num1 + num2;
        elif num1 is not None:
            return num1;
        else:
            return "Hello"
obj = Data()
print(obj.m(11,22))
print(obj.m())
print(obj.m(7))
```

```
33
Hello
7
```

- `Constructor Overloading:`

In [23]:
```
class Data:
    def __init__(self, num1=None, num2=None):
        if num1 is not None and num2 is not None:
            print(num1+num2)
        elif num1 is not None:
            print(num1)
        else:
            print("Hello")
Data(11,22)
Data()
Data(11)
```

```
33
Hello
11
```

Out[23]:
```
<__main__.Data at 0x2bc5db77290>
```

- `Method Overridding`

In [4]:
```
class A:
    def m1(self):
        print("Hello")
class B(A):
    # Overriding the parent method
    def m1(self):
        print("Hi")
class C(B):
    # Overriding the parent method
    def m1(self):
        print("Bye")

# Creating objects
a = A()
b = B()
c = C()

# Calling the method
a.m1()
b.m1()
c.m1()
```

```
Hello
Hi
Bye
```

- `super() Method: Required Inheritance`

In [34]:
```
class A:
    def m1(self):
        print("Hello")
class B(A):
    def m1(self):
        print("Hi")
        super().m1()
class C(B):
    def m1(self):
        print("Bye")
        super().m1()
obj = C()
obj.m1()
```

```
Bye
Hi
Hello
```

- `Exception Handling:`
  1. `ZeroDivisionError:` `eg,.96/0`
  2. `IndexError:` `List out of bound eg,.a[7], range out of list eg,.(len(a)+2)`
  3. `NameError:` `Function Misspelled eg,.(data2()), Not Defined | Undefined Variable eg,.(print(a2))`
  4. `keyError:` `Dictionary eg,.(data[7])`
  5. `TypeError:` `String and Float Concatination eg,.("Pratik"+7), Iterate String on Integer eg,.`

(for Pratik in 1234567) pass

6. ValueError: Type Conversion eg,.(name = float("Hello")), Unboxing the List eg,.(p,q,r,s,t = data)

In [37]:
```
class Data():
    def logic(self):
        a = 94
        b = 2
        res = a/b
        print("Result is: ",res)
obj = Data()
obj.logic()
print("Executed")
```

```
Result is:  47.0
executed
```

- 96/0 - Zero Division Error - division by zero

In [38]:
```
class Data():
    def logic(self):
        a = 94
        b = 0
        res = a/b
        print("Result is: ",res)
obj = Data()
obj.logic()
print("Executed")
# Exception = division by zero
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[38], line 8
      6         print("Result is: ",res)
      7 obj = Data()
----> 8 obj.logic()
      9 print("Executed")

Cell In[38], line 5, in Data.logic(self)
      3 a = 94
      4 b = 0
----> 5 res = a/b
      6 print("Result is: ",res)

ZeroDivisionError: division by zero
```

- 96/0 - Zero Division Error - division by zero

In [40]:
```
class Data:
    def logic(self):
        try:
            a = 96
            b = 0
            res = a / b
        except ZeroDivisionError:
            print("Division by zero not Possible.")
obj = Data()
obj.logic()
print("Executed")
```

```
Division by zero not Possible.
Executed
```

- List - Index Error - list index out of range

In [1]:
```
class Data:
    def logic(self):
        a = [11,22,33,44]
        print(a[7])
obj = Data()
obj.logic()
print("Executed")
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[1], line 6
      4         print(a[7])
      5 obj = Data()
----> 6 obj.logic()
      7 print("Executed")

Cell In[1], line 4, in Data.logic(self)
      2 def logic(self):
      3     a = [11,22,33,44]
----> 4     print(a[7])

IndexError: list index out of range
```

- `List - Index Error - list index out of range`

In [2]:
```
class Data:
    def logic(self):
        try:
            a = [11,22,33,44]
            print(a[7])
        except IndexError:
            print("Invalid Index")
obj = Data()
obj.logic()
print("Executed")
```

```
Invalid Index
Executed
```

- `List - Index Error - list index out of range`

In [97]:
```
a = [11,22,33,44,55]
for i in range(len(a)+2):
    print(a[i])
```

```
11
22
33
44
55
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[97], line 3
      1 a = [11,22,33,44,55]
      2 for i in range(len(a)+2):
----> 3     print(a[i])

IndexError: list index out of range
```

- `List - Index Error - list index out of range`

In [27]:
```
try:
    b = [11,22,33,44,55]
    for i in range(len(b)+22):
        # pass
        print(b[i])
except IndexError:
    print("Index out of Bound.")
print("Executed.")
```

```
11
22
33
44
55
Index out of Bound.
Executed.
```

- `variable - Name Error`

In [63]:
```
try:
    print(a2)
except NameError:
    print("Undefined Variable.")
# print(a2)
print("Executed")
```

```
Undefined Variable.
Executed
```

- `Function - Name Error`

In [ ]:
```
def data():
    print("Hello")
try:
    data2()
except NameError:
    print("Misspell Function.")
print("Executed.")
```

- `4/0 - Zero Division Error`

In [73]:
```
try:
    a = 4
    b = 0
    res = int(a/b)
    print(res)
except ZeroDivisionError:
    print("Index out of bound Error.")
print("Executed")
```

```
Index out of bound Error.
Executed
```

- `Function - Name Error`

In [77]:
```
def pandu():
    print("Talking")

try:
    pandi()
except NameError:
    print("NameError")
print("Executed")
```

```
NameError
Executed
```

- `List - Index Error`

In [81]:
```
try:
    a = [11,22,33,44,55]
    print(a[5])
except IndexError:
    print("IndexError")
print("Executed")
```

```
IndexError
Executed
```

- `Dictionary - Key Error`

In [7]:
```
data = {1:"Python", 2:"Java",3:"C#",4:"Kotlin"}
print(data[1])
print(data[5]) # KeyError
```

```
Python
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[7], line 3
      1 data = {1:"Python", 2:"Java",3:"C#",4:"Kotlin"}
      2 print(data[1])
----> 3 print(data[5])

KeyError: 5
```

- `Dictionary - Key Error`

In [9]:
```
try:
    data = {1:"Python", 2:"Java",3:"C#",4:"Kotlin"}
    print(data[1])
    print(data[5]) # KeyError
except KeyError:
```

```
    print("key Error.")
print("Executed.")
```

```
Python
key Error.
Executed.
```

- `'int' and 'str' - Type Error - unsupported operand type(s) for +: 'int' and 'str' - Miss Match Data type Error`

In [10]:
```
sum = 23 + "Python"
print(sum)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[10], line 1
----> 1 sum = 23 + "Python"
      2 print(sum)

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

- `'int' and 'str' - Type Error - unsupported operand type(s) for +: 'int' and 'str' - Miss Match Data type Error`

In [14]:
```
try:
    sum = 23 + "Python"
    print(sum)
except TypeError:
    print("Miss Match Data Type Error.")
print("Executed")
```

```
Miss Match Data Type Error.
Executed
```

- `'str' object is not callable - TypeError`

In [13]:
```
data = "Python"
print(data())
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[13], line 2
      1 data = "Python"
----> 2 print(data())

TypeError: 'str' object is not callable
```

In [16]:
```
try:
    data = "Python"
    print(data())
except:
    print("Miss Match Type Error.")
print("Executed.")
```

```
Miss Match Type Error.
Executed.
```

- `Itrating string on integer - TypeError - 'int' object is not iterable`

In [17]:
```
for python in 12345678:
    pass
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[17], line 1
----> 1 for python in 12345678:
      2     pass

TypeError: 'int' object is not iterable
```

- `Itrating string on Integer - TypeError - 'int' object is not iterable`

In [20]:
```
try:
    for python in 12345678:
        pass
except:
    print("'int' object is not iterable")
print("Executed")
```

```
'int' object is not iterable
Executed
```

- `Int to Float Conversion - Type Conversion`

In [21]:
```
num = 43
print(float(num))
```

```
43.0
```

- `String to Float Conversion - ValueError - could not convert string to float: 'Python'`

In [24]:
```
num = "Python"
print(float(num))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[24], line 2
      1 num = "Python"
----> 2 print(float(num))

ValueError: could not convert string to float: 'Python'
```

- `String to Float Conversion - ValueError - could not convert string to float: 'Python' - Incompatible Value.`

In [25]:
```
try:
    num = "Python"
    print(float(num))
except:
    print("Incompatible Value.")
print("Executed.")
```

```
Incompatible Value.
Executed.
```

In [29]:
```
data = [11,22,33,44,55]
print(data)
# Unboxing the List : Storing Indivisual elements into the Variable.
p,q,r,s,t = data
print(p)
print(q)
print(r)
print(s)
print(t)
```

```
[11, 22, 33, 44, 55]
11
22
33
44
55
```

- `not enough values to unpack (expected 6, got 5)`

In [30]:
```
data = [11,22,33,44,55]
print(data)
# Unboxing the List : Storing Indivisual elements into the Variable.
p,q,r,s,t,u = data
print(p)
print(q)
print(r)
print(s)
print(t)
```

```
[11, 22, 33, 44, 55]
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[30], line 4
      2 print(data)
      3 # Unboxing the List : Storing Indivisual elements into the Variable.
----> 4 p,q,r,s,t,u = data
      5 print(p)
      6 print(q)

ValueError: not enough values to unpack (expected 6, got 5)
```

- `too many values to unpack (expected 4)`

In [31]:
```
data = [11,22,33,44,55]
print(data)
# Unboxing the List : Storing Indivisual elements into the Variable.
p,q,r,s = data
print(p)
print(q)
print(r)
print(s)
print(t)
```

```
[11, 22, 33, 44, 55]
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[31], line 4
      2 print(data)
      3 # Unboxing the List : Storing Indivisual elements into the Variable.
----> 4 p,q,r,s = data
      5 print(p)
      6 print(q)

ValueError: too many values to unpack (expected 4)
```

- Unboxing the List - Value Error - Storing Indivisual elements into the Indivisual Variable.

In [34]:
```
try:
    data = [11,22,33,44,55]
    print(data)
    # Unboxing the List : Storing Indivisual elements into the Variable.
    p,q,r,s = data
    print(p)
    print(q)
    print(r)
    print(s)
    print(t)
except ValueError:
    print("Value Error.")
print("Executed.")
```

```
[11, 22, 33, 44, 55]
Value Error.
Executed.
```

Exception Handling : try, except, else, finally

- eg., ZeroDivisionError
- Except block executed : When exception throw by try block.
- Else block executed : When Exception not throw by try block.
- Finnaly block executed : Whether Exception throw or not throw by try block.

In [51]:
```
a = 96
b = 2
res = a/b
print(res)
```

```
48.0
```

In [48]:
```
a = 96
b = 0
res = a/b
print(res)
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[48], line 3
      1 a = 96
      2 b = 0
----> 3 res = a/b
      4 print(res)

ZeroDivisionError: division by zero
```

In [42]:
```
try:
    a = 96
    b = 2
    res = int(a/b)
    print(res)
except ZeroDivisionError:
    print("Zero Division Error.")
print("Executed")
```

```
48
Executed
```

In [39]:
```
try:
    a = 96
    b = 0
    res = a/b
    print(res)
except ZeroDivisionError:
    print("Zero Division Error.")
print("Executed")
```

```
Zero Division Error.
Executed
```

In [43]:
```
try:
    a = 96
    b = 0
    res = a/b
    print(res)
except ZeroDivisionError:
    print("Zero Division Error.")
else:
    print("Code Successfully Execued.")
print("Executed")
```

```
Zero Division Error.
Executed
```

In [44]:
```
try:
    a = 96
    b = 2
    res = a/b
    print(res)
except ZeroDivisionError:
    print("Zero Division Error.")
else:
    print("Code Successfully Execued.")
print("Executed")
```

```
48.0
Code Successfully Execued.
Executed
```

In [54]:
```
try:
    a = 96
    b = 2
    res = a/b
    print(res)
# except ZeroDivisionError:
    # print("Zero Division Error.")
else:
    print("Code Successfully Execued.")
print("Executed")
```

```
  Cell In[54], line 8
    else:
    ^
SyntaxError: expected 'except' or 'finally' block
```

In [52]:
```
try:
    a = 96
    b = 2
    res = a/b
    print(res)
except ZeroDivisionError:
    print("Zero Division Error.")
else:
    print("Code Successfully Execued.")
finally:
    print("Important Code Executed.")
print("Executed")
```

```
48.0
Code Successfully Execued.
Important Code Executed.
Executed
```

In [53]:
```
try:
    a = 96
    b = 0
    res = a/b
    print(res)
except ZeroDivisionError:
    print("Zero Division Error.")
```

```
else:
    print("Code Successfully Execued.")
finally:
    print("Important Code Executed.")
print("Executed")
```

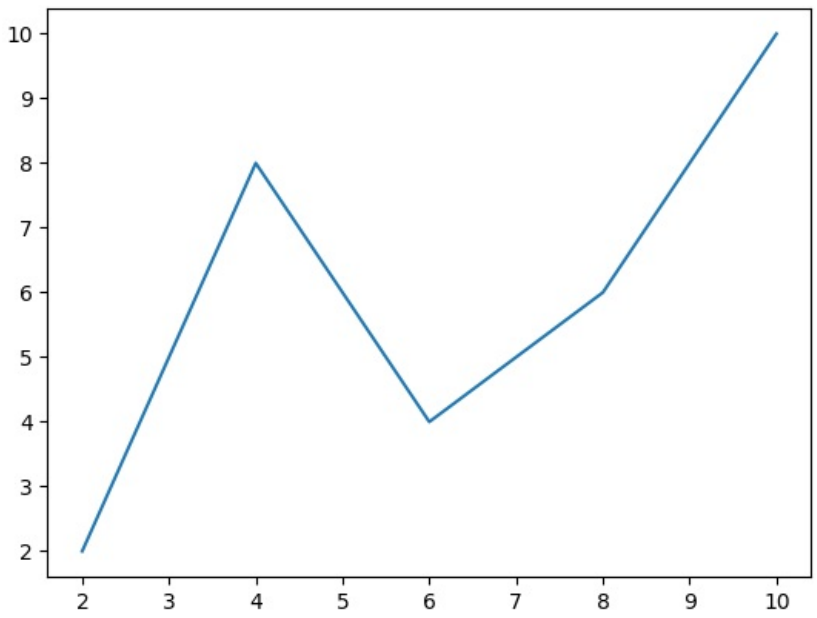
```
Zero Division Error.
Important Code Executed.
Executed
```

In [2]:
```
pip install matplotlib
```

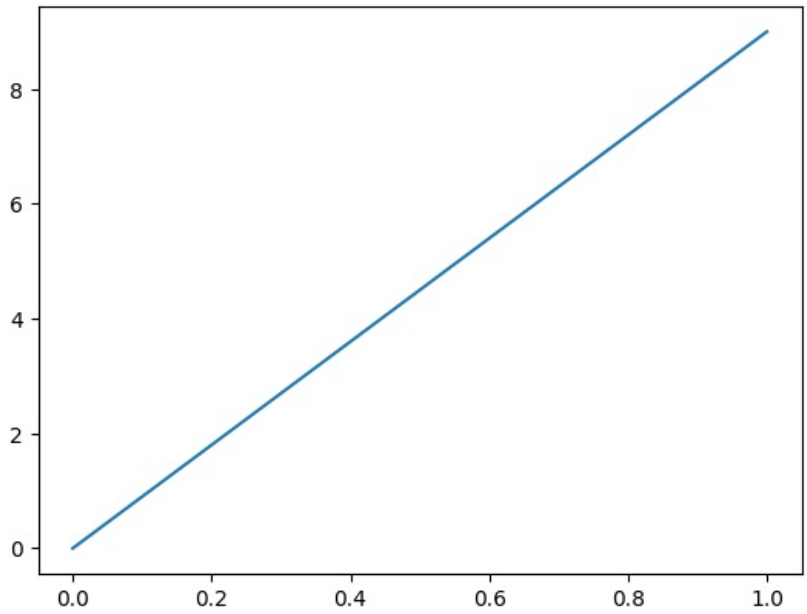
```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: matplotlib in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (3.10.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from matplotlib) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from matplotlib) (1.4.8)
Requirement already satisfied: numpy>=1.23 in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from matplotlib) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from matplotlib) (3.1.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
[notice] A new release of pip is available: 24.2 -> 25.0.1
[notice] To update, run: C:\Users\majag\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip
```

In [3]:
```
import matplotlib.pyplot as pt
import numpy as np
```

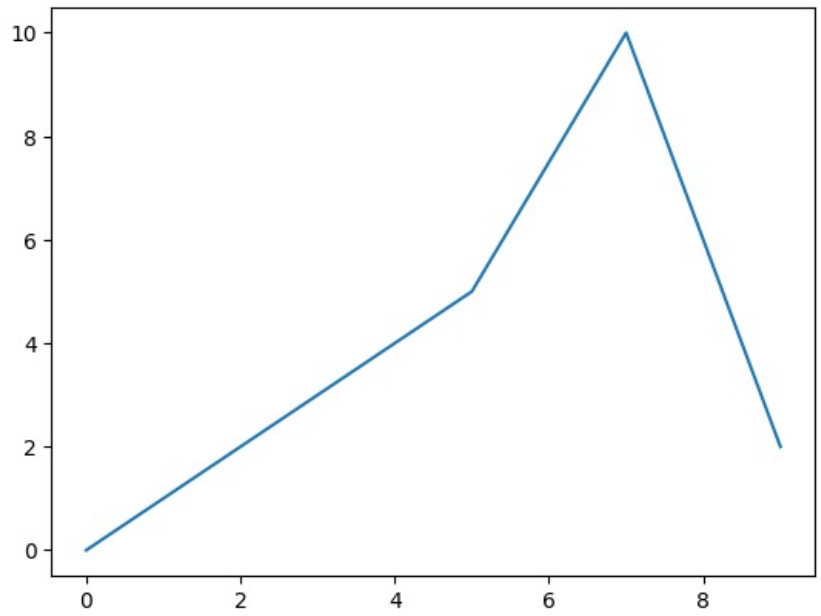
In [6]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y)
pt.show()
```



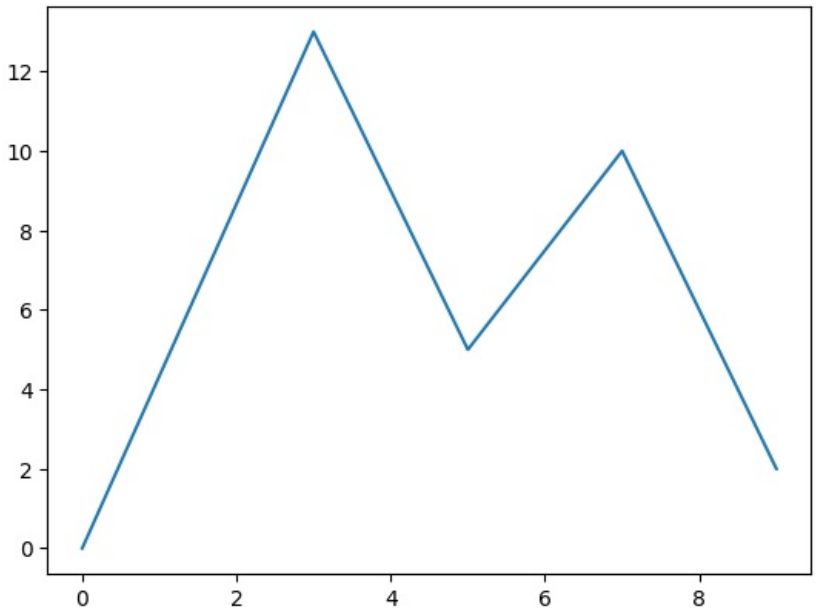
In [7]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([0,1])
y = np.array([0,9])
pt.plot(x,y)
pt.show()
```



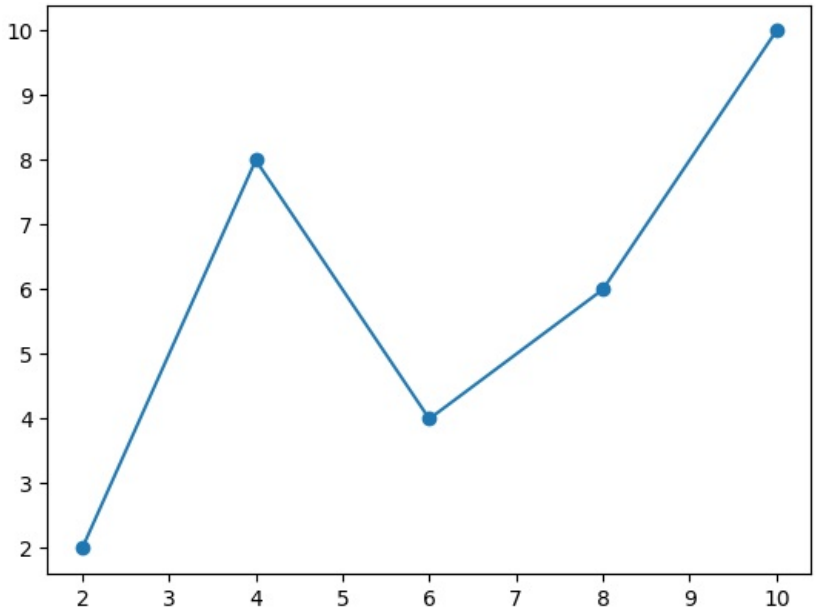
In [8]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([0,3,5,7,9])
y = np.array([0,3,5,10,2])
pt.plot(x,y)
pt.show()
```



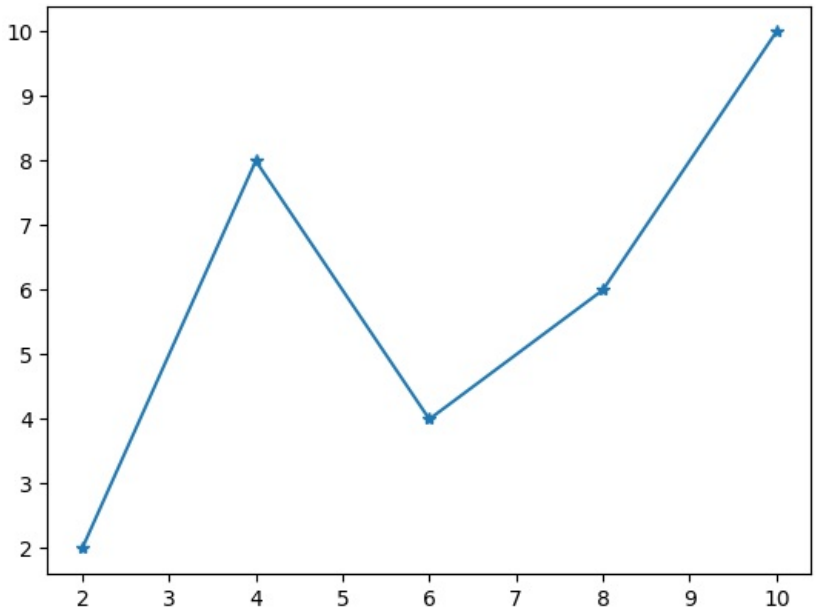
In [9]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([0,3,5,7,9])
y = np.array([0,13,5,10,2])
pt.plot(x,y)
pt.show()
```



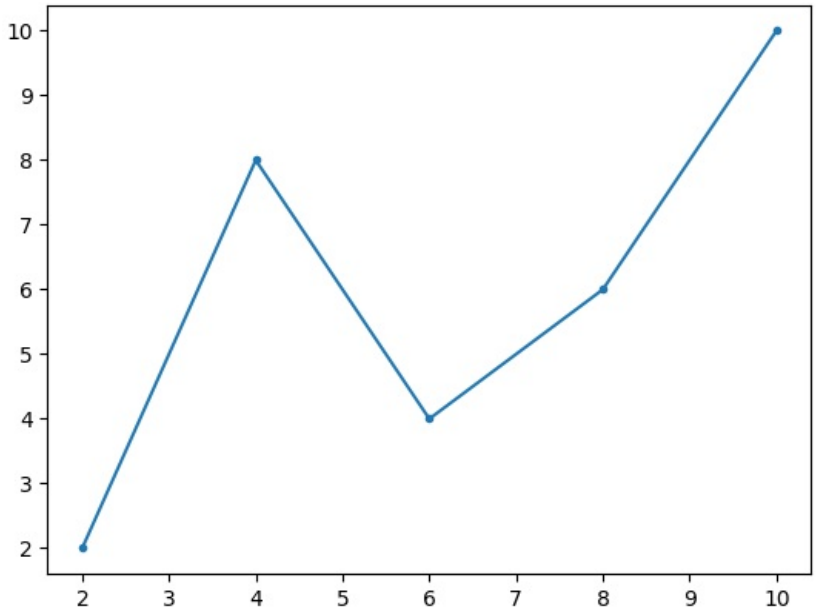
In [11]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='o')
pt.show()
```



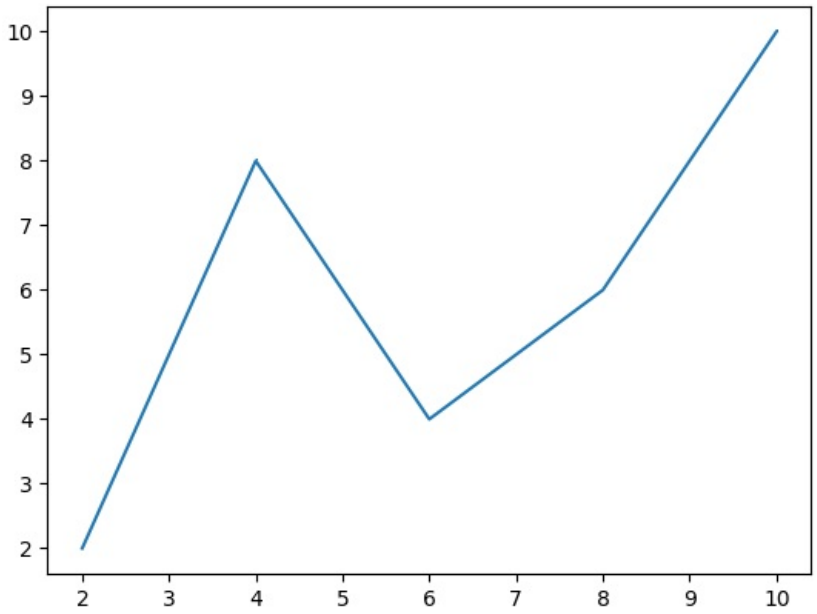
In [12]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='*')
pt.show()
```



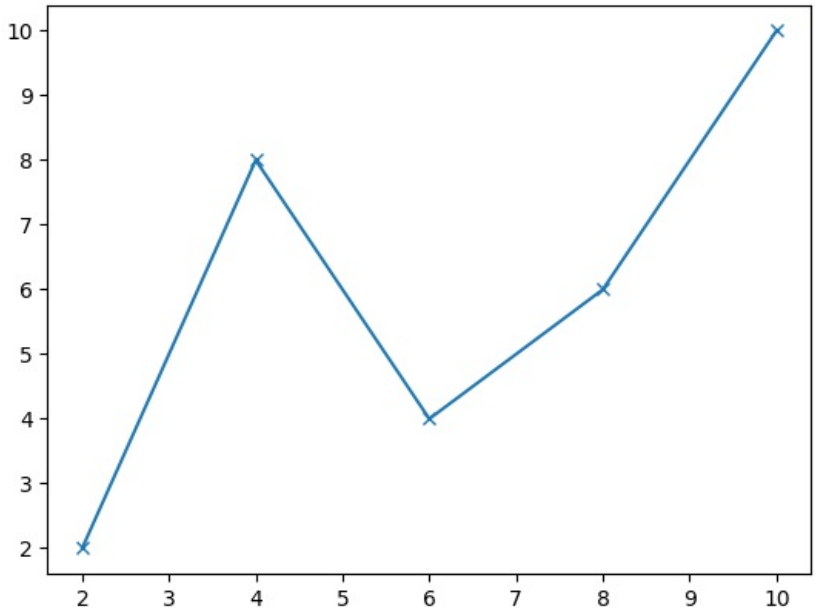
In [13]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='.')
pt.show()
```



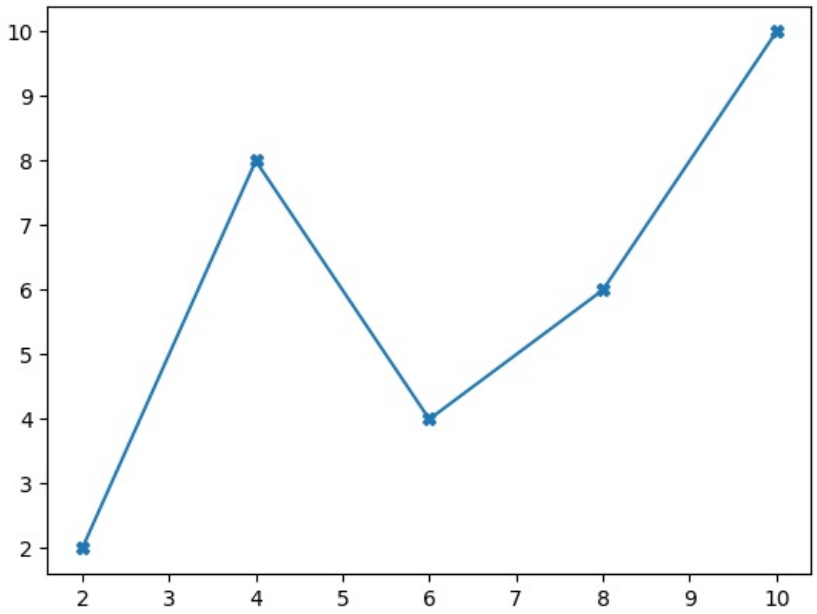
In [14]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker=',')
pt.show()
```



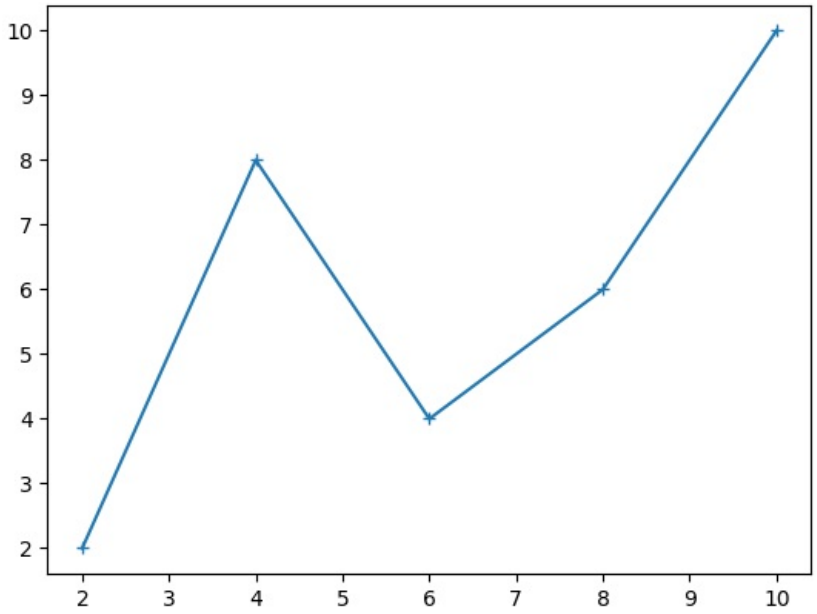
In [15]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='x')
pt.show()
```



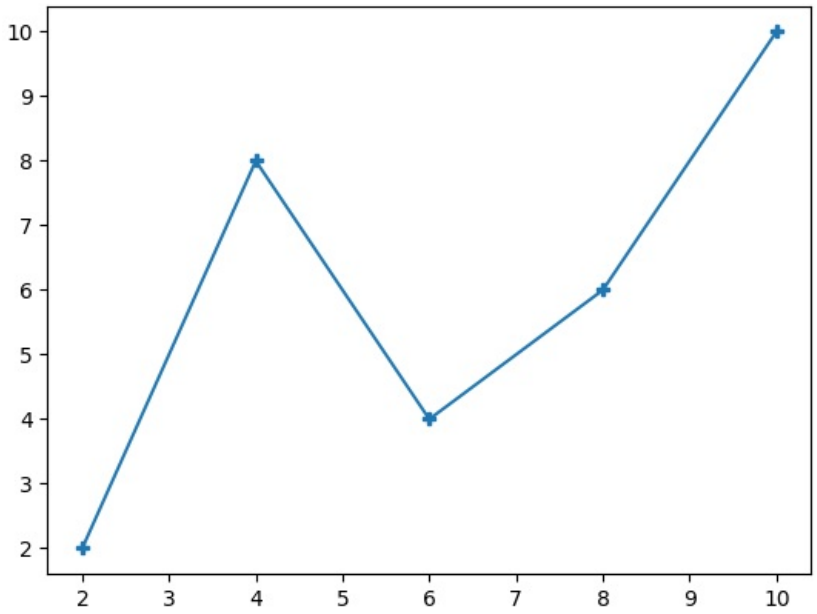
In [16]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='X')
pt.show()
```



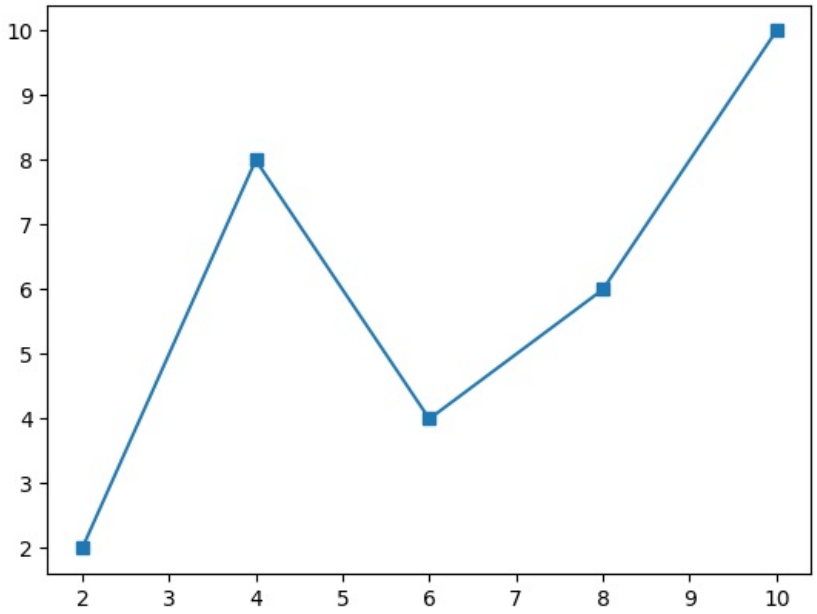
In [17]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='+')
pt.show()
```



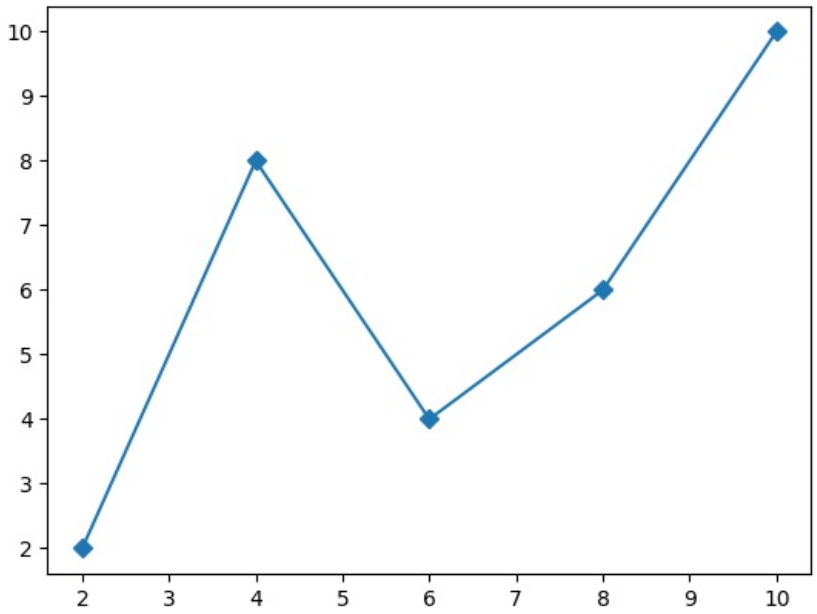
In [18]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='P')
pt.show()
```



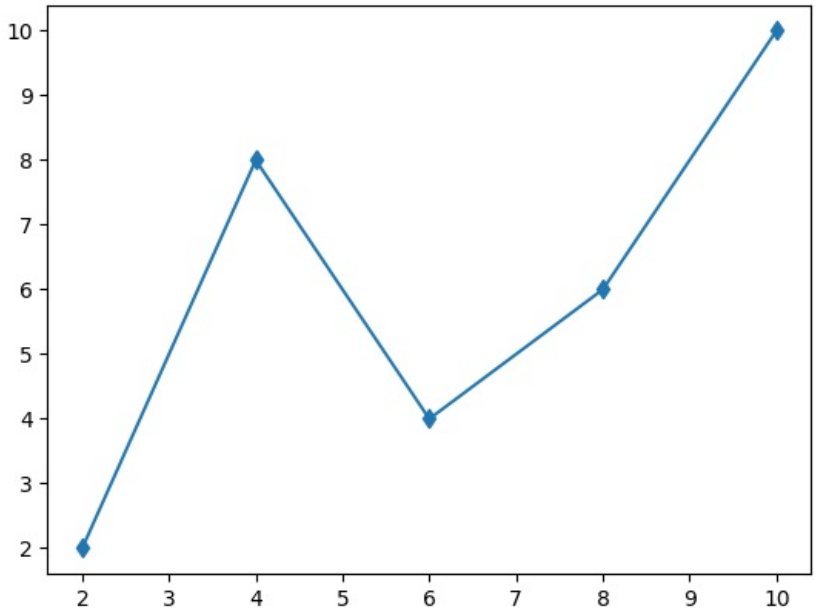
In [19]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='s')
pt.show()
```



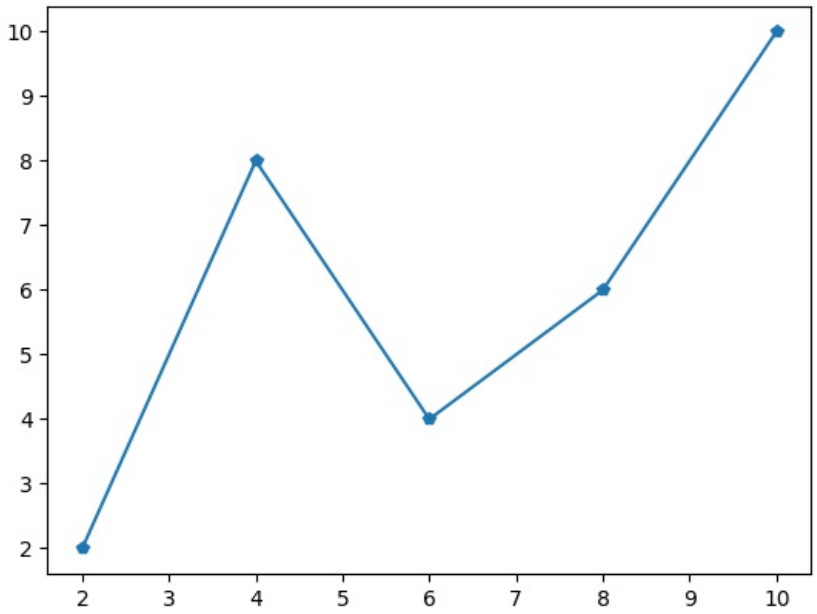
In [20]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='D')
pt.show()
```



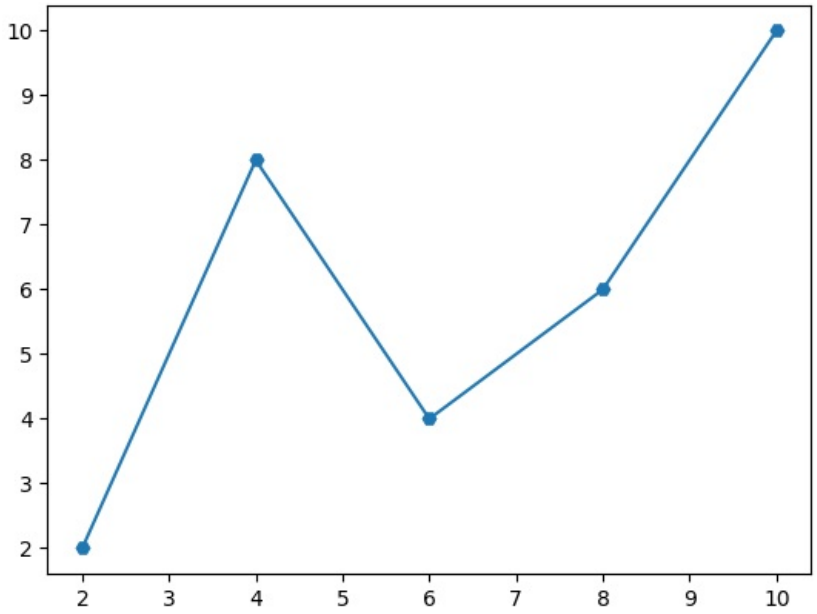
In [21]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='d')
pt.show()
```



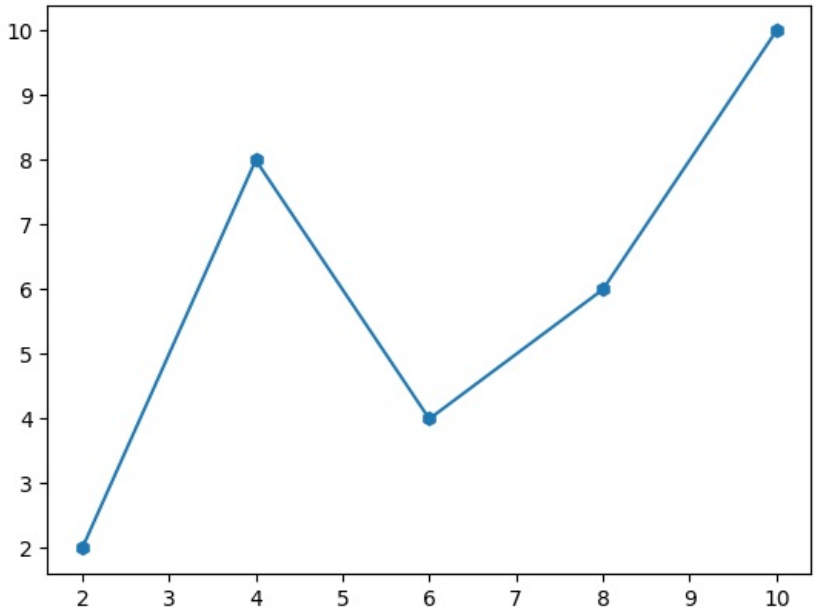
In [22]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='p')
pt.show()
```



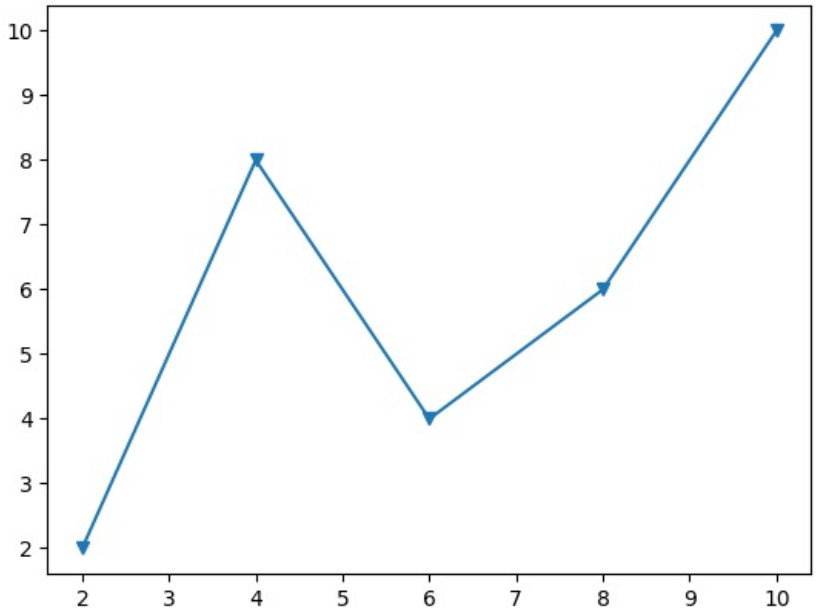
In [23]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='H')
pt.show()
```



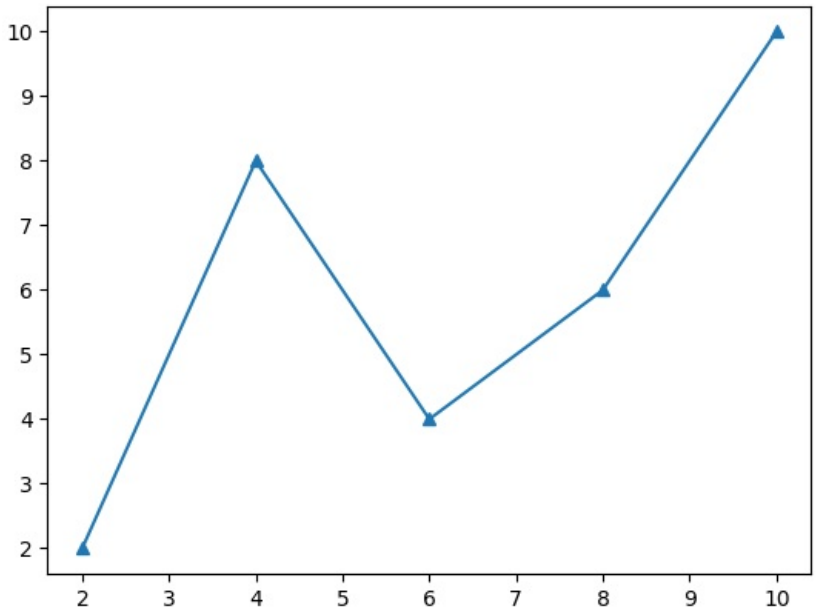
In [24]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='h')
pt.show()
```



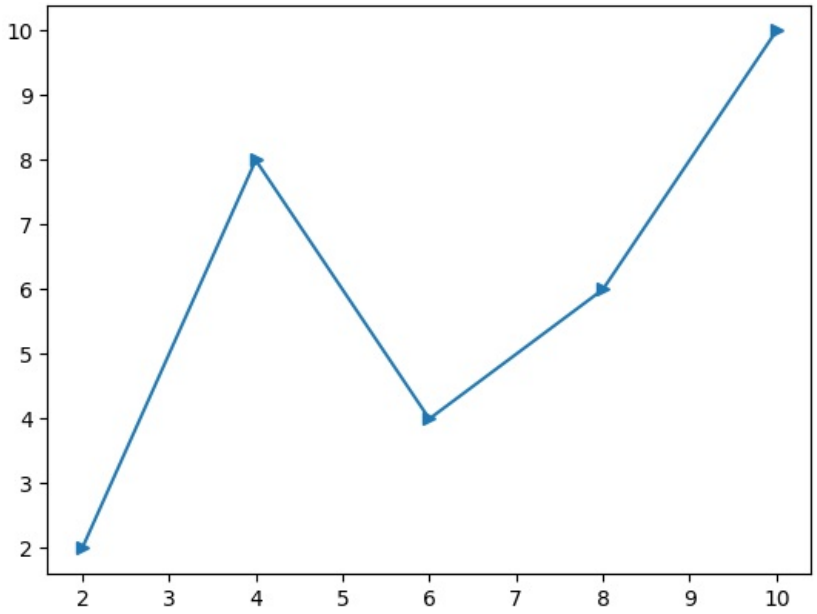
In [26]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='v')
pt.show()
```



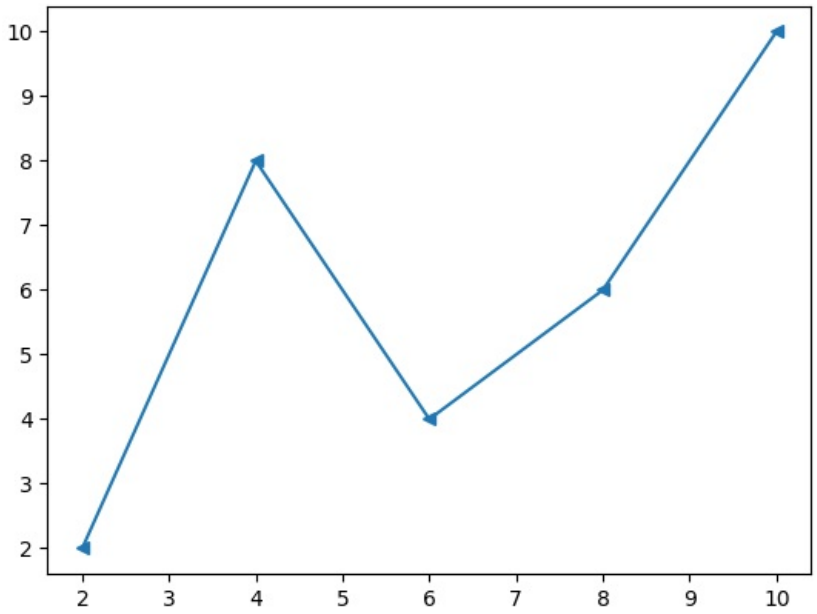
In [27]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='^')
pt.show()
```



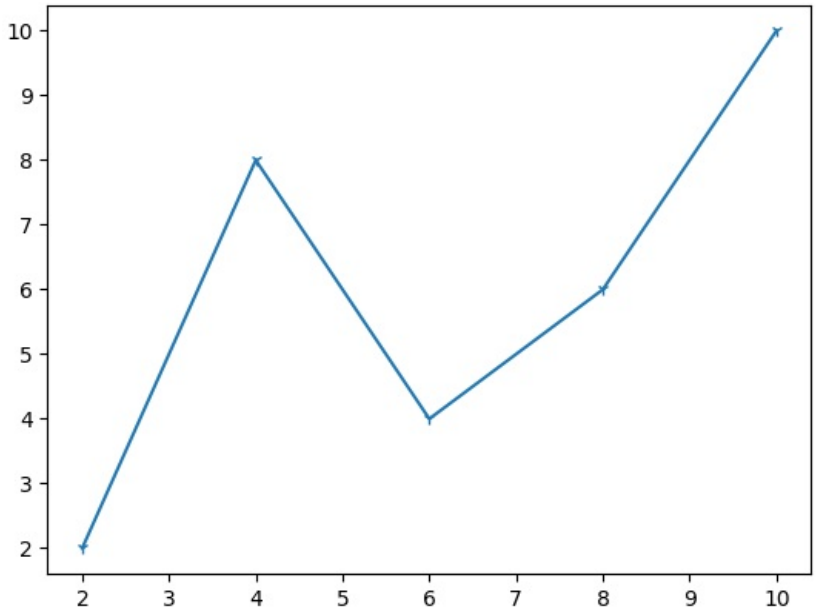
In [28]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='>')
pt.show()
```



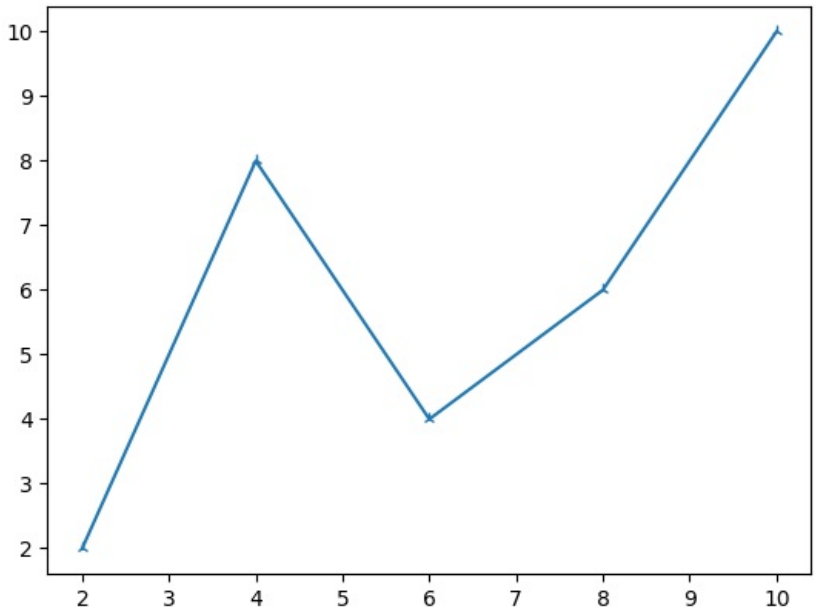
In [29]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='<')
pt.show()
```



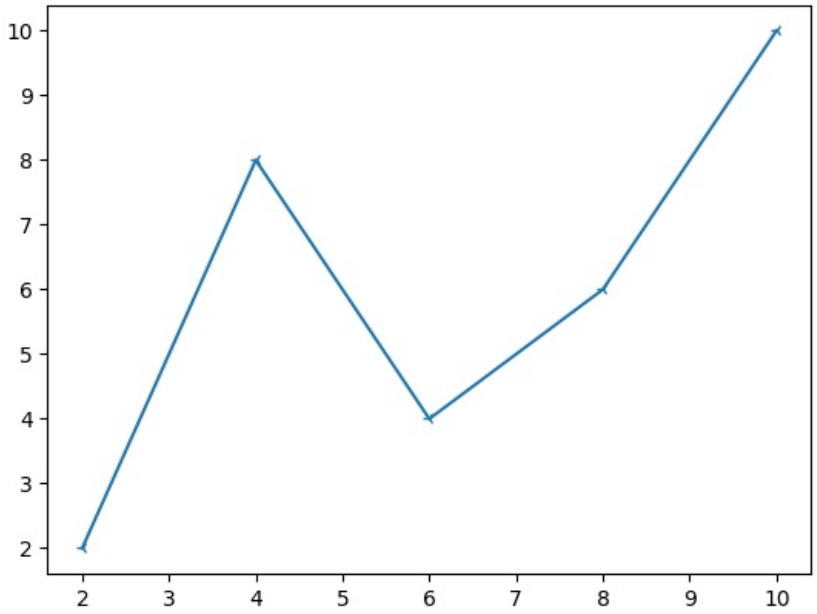
In [30]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='1')
pt.show()
```



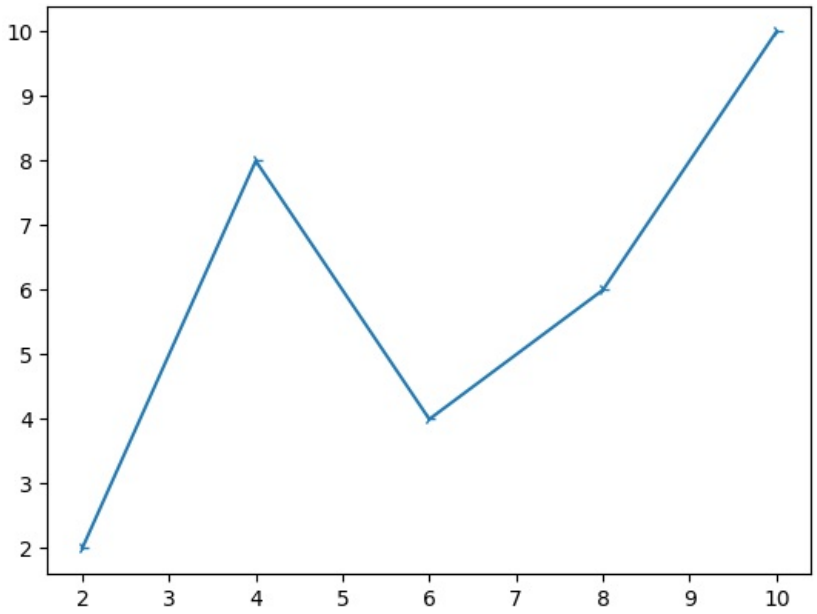
In [31]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='2')
pt.show()
```



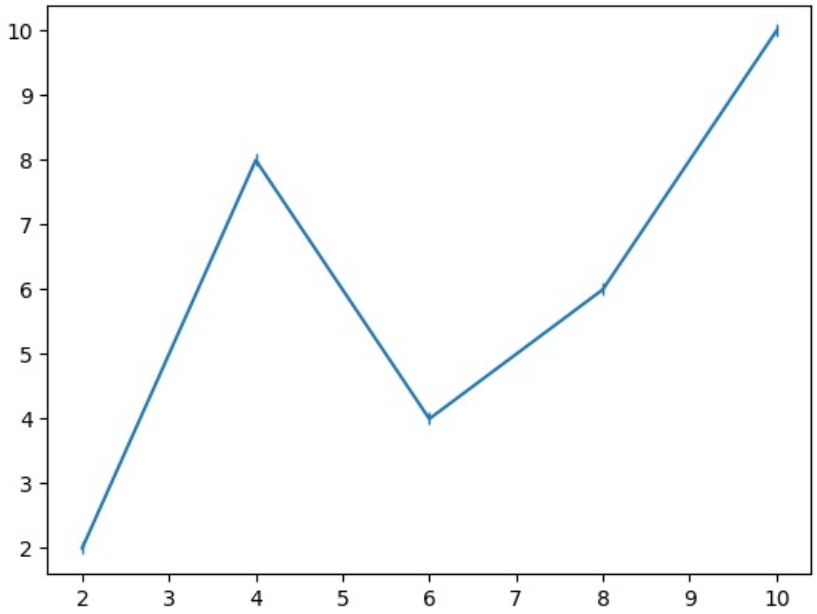
In [32]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='3')
pt.show()
```



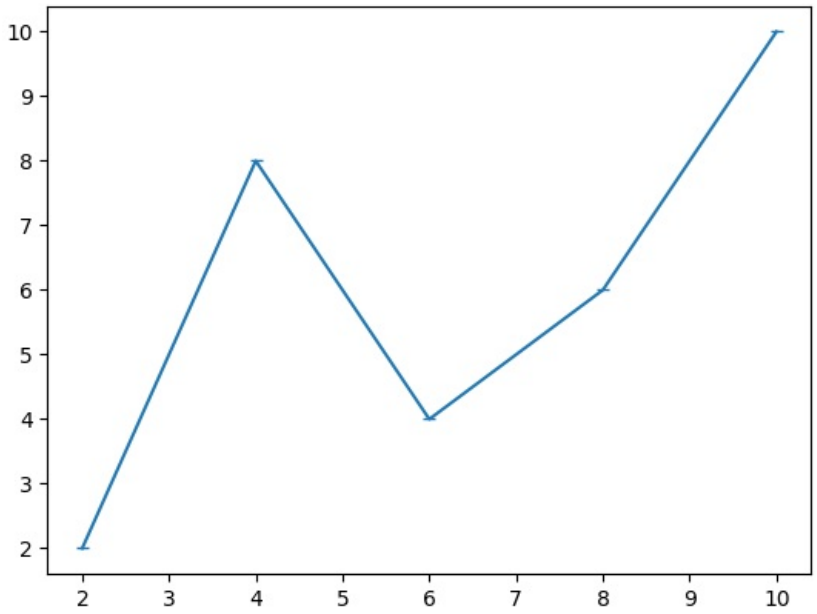
In [33]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='4')
pt.show()
```



In [34]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='|')
pt.show()
```

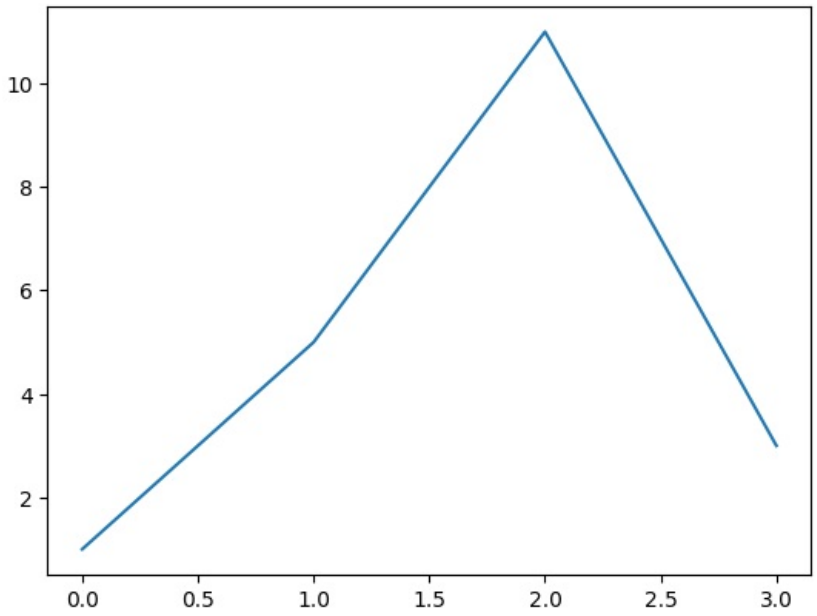


In [36]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([2,8,4,6,10])
pt.plot(x,y,marker='_')
pt.show()
```

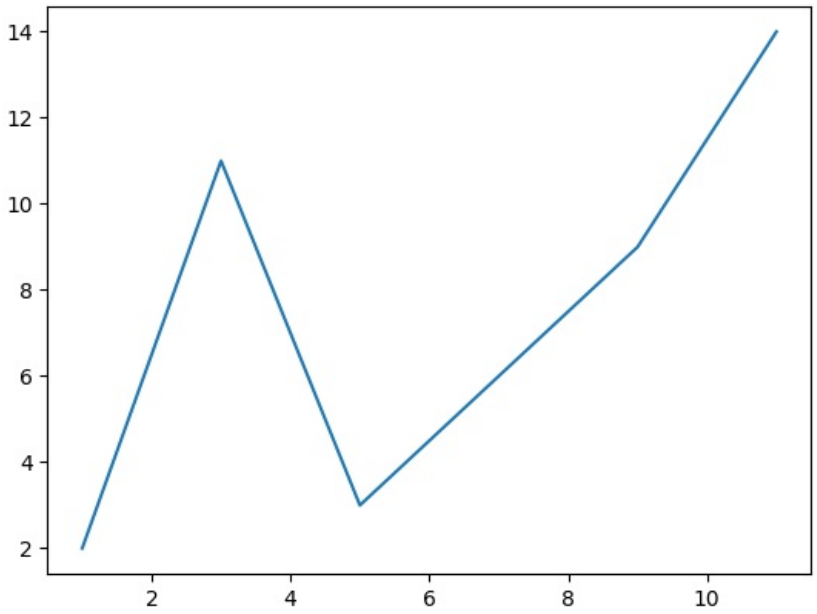


- `Default x axis: When only 1 point of a single axis are given then Default values of x-axis are taken.`
- `Default y axis is not possible.`

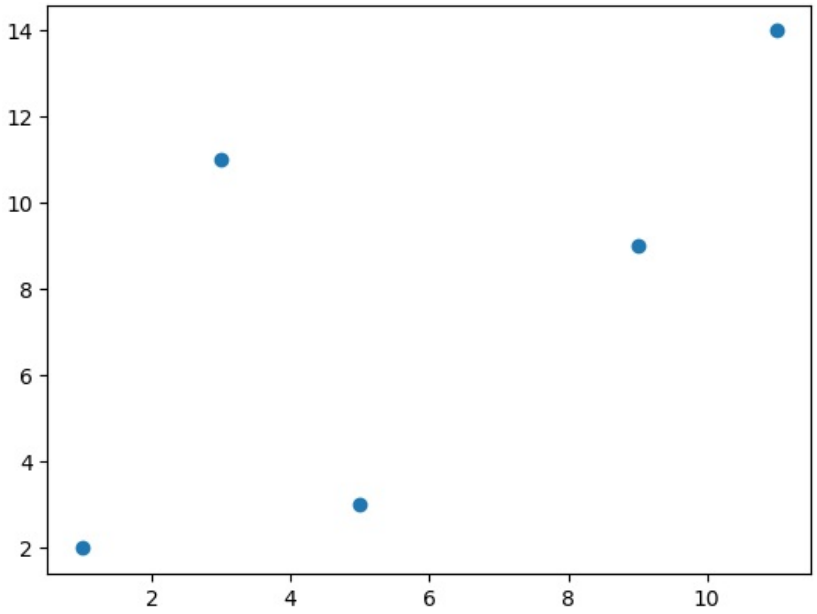
In [37]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,5,11,3])
pt.plot(x)
pt.show()
```



In [39]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y)
pt.show()
```

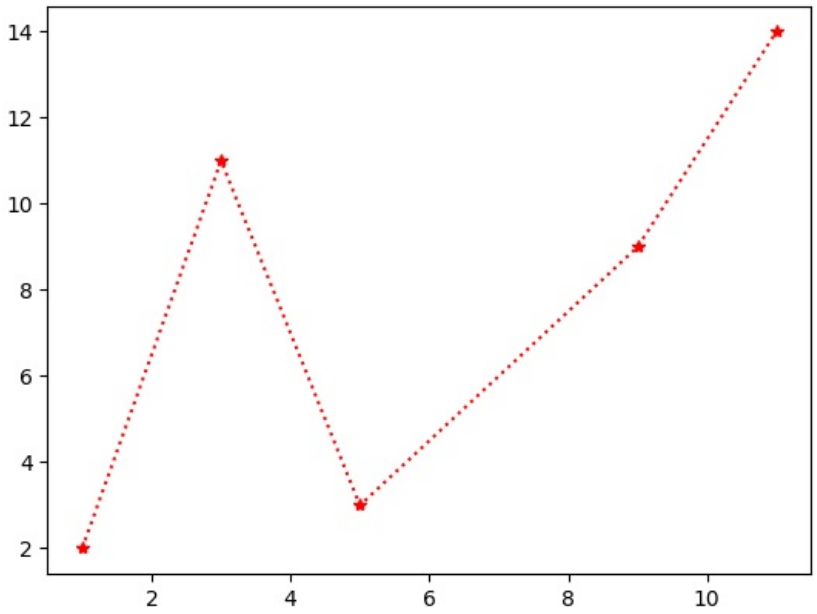


In [40]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'o')
pt.show()
```

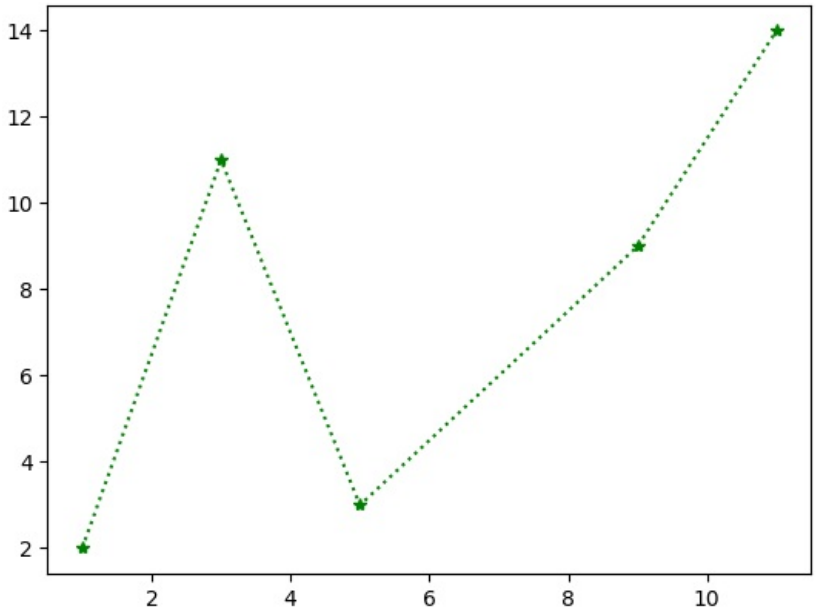


- `Color:`

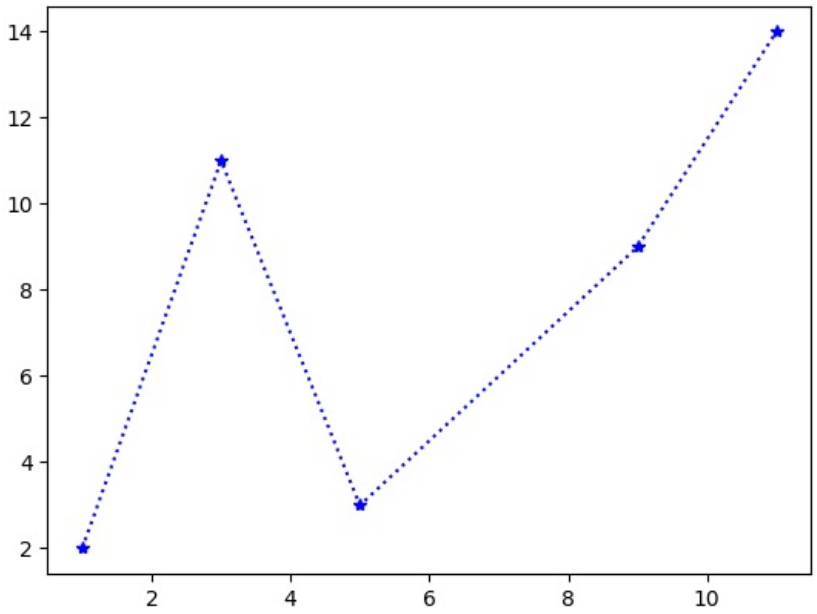
In [41]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'*:r')
pt.show()
```



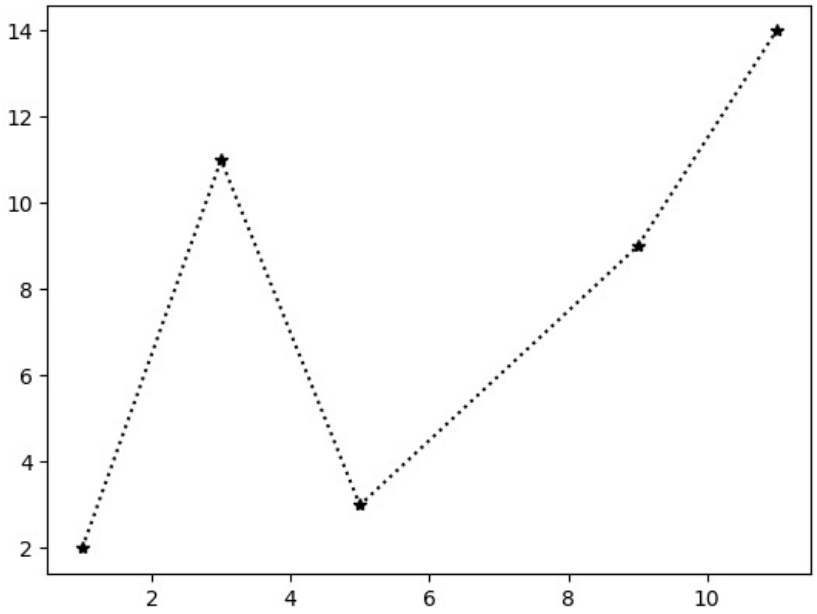
In [43]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'*:g')
pt.show()
```



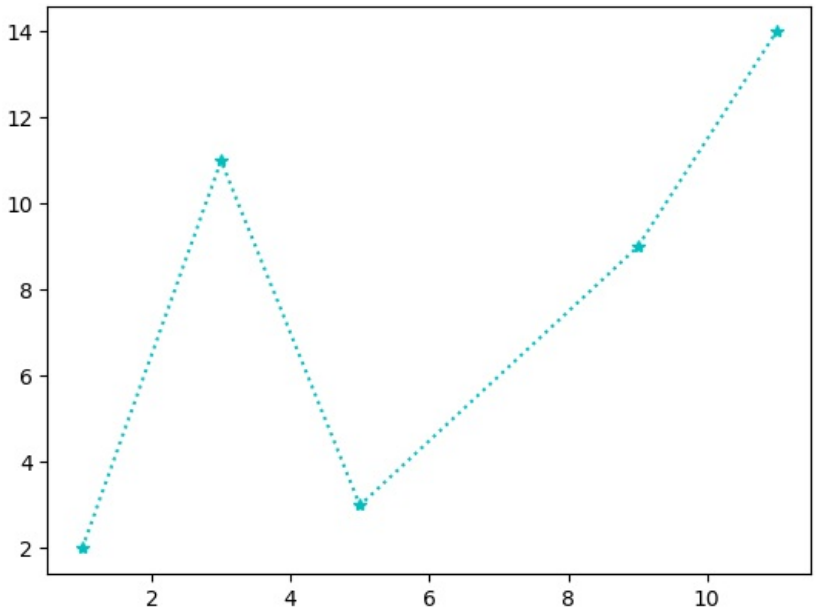
In [44]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'*:b')
pt.show()
```



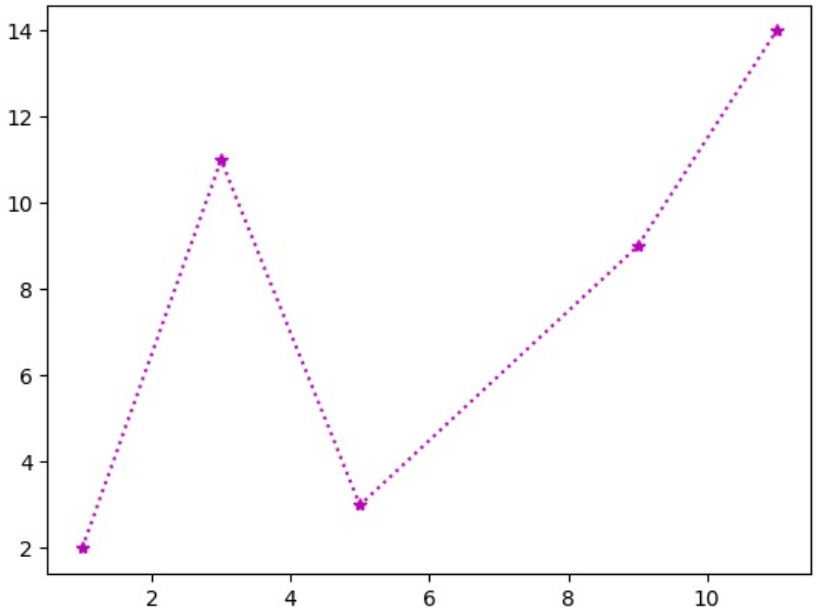
In [45]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'*:k')
pt.show()
```



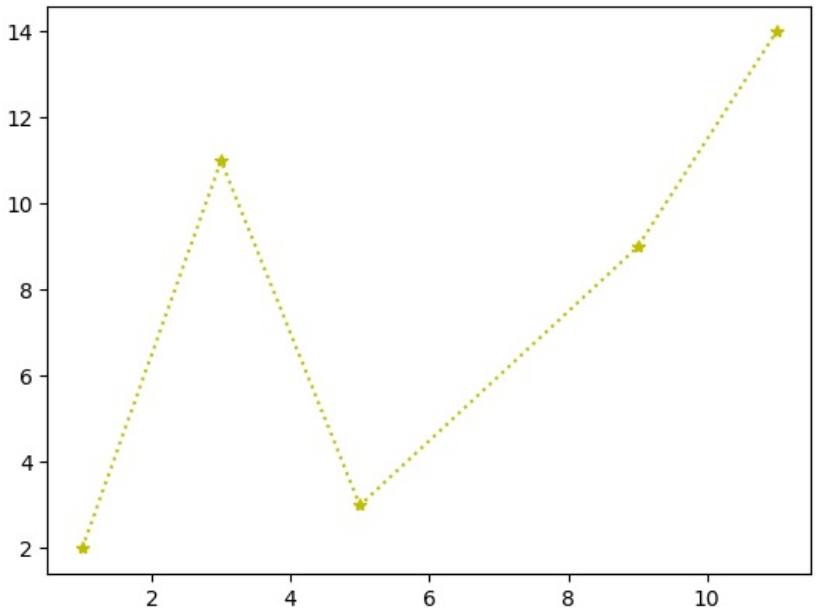
In [47]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'*:c')
pt.show()
```



In [48]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'*:m')
pt.show()
```



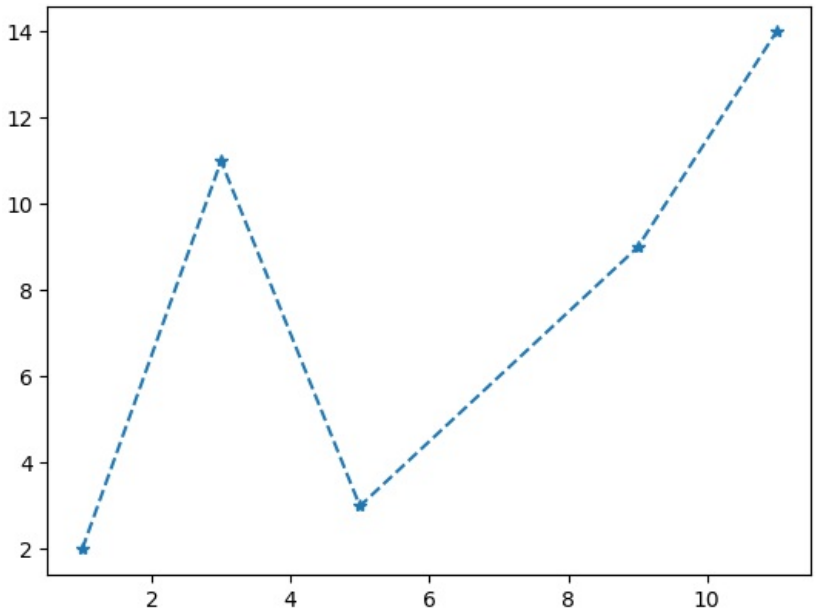
In [49]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'*:y')
pt.show()
```



- `Line Style:`

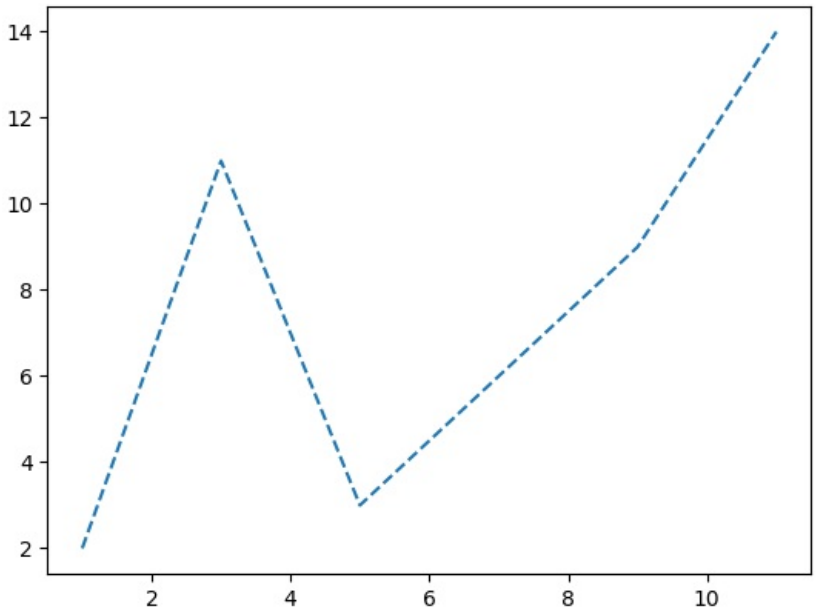
1. `Dashed Line:`

In [53]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'*--')
pt.show()
```



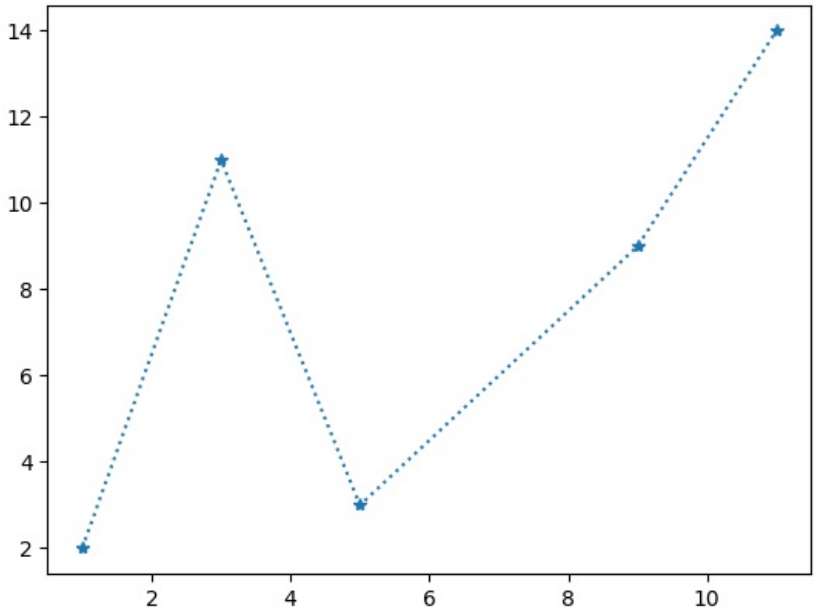
2. `Double Dashed:`

In [61]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'--')
pt.show()
```



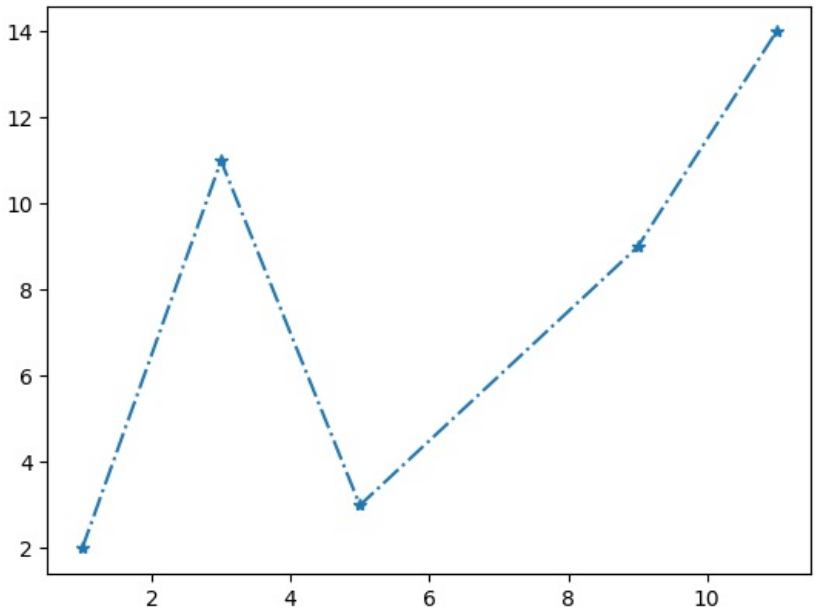
3. `Dotted Line`

In [66]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'*:')
pt.show()
```



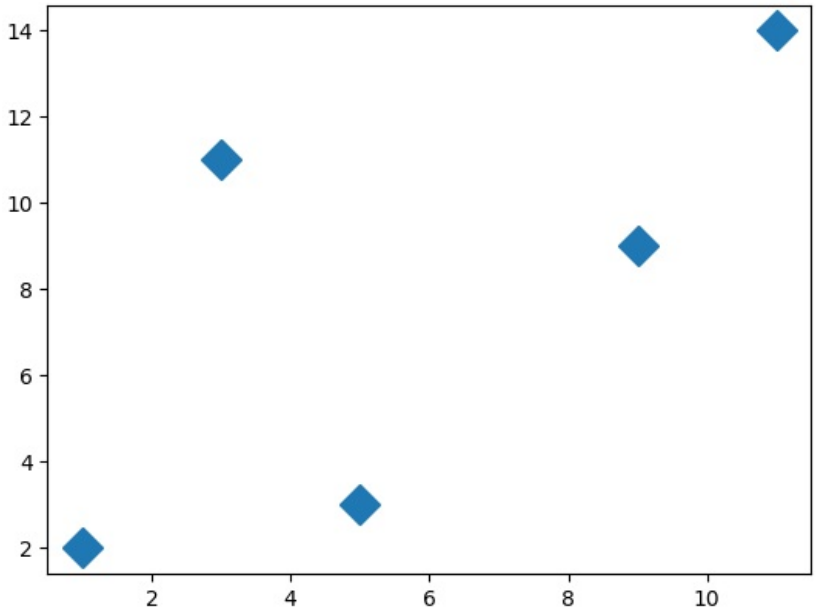
- `Dashed Dotted:`

In [64]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'*-.')
pt.show()
```

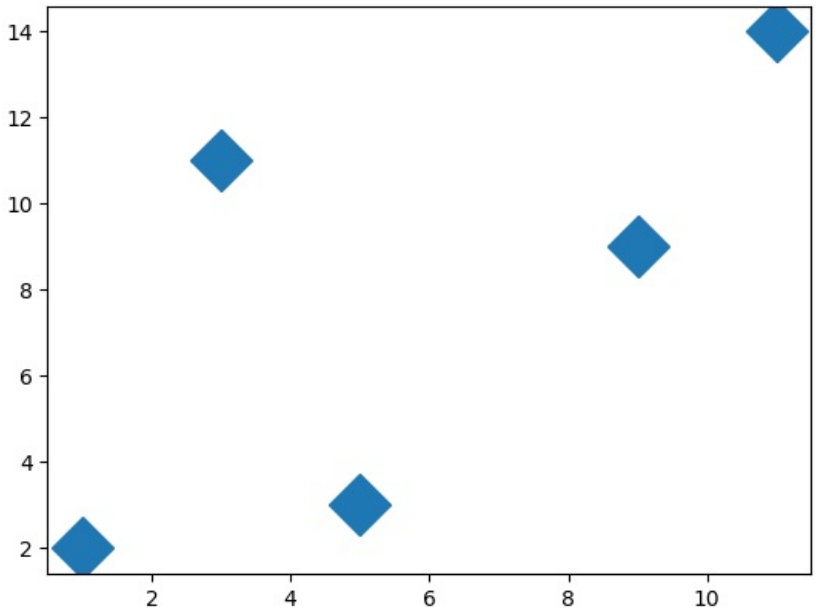


- `Marker Size: (ms)`

In [67]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'D',ms=13.3)
pt.show()
```

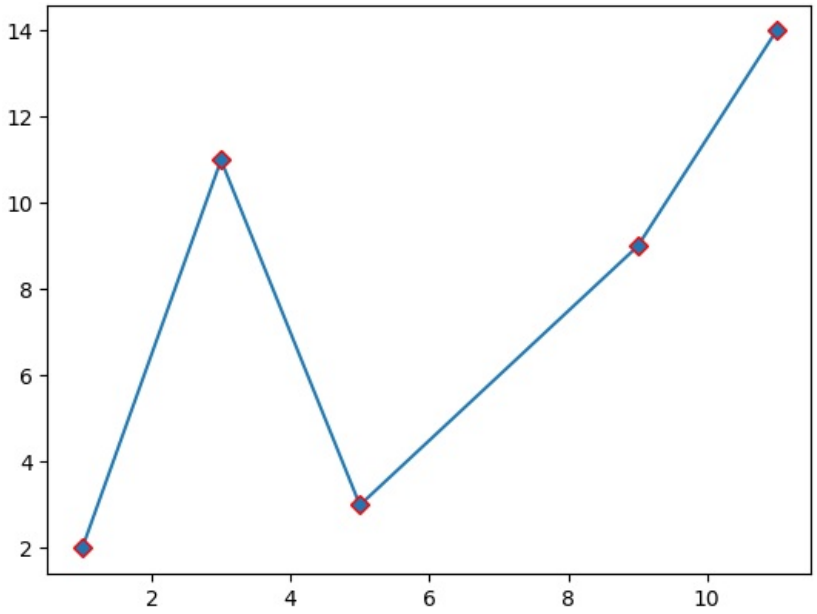


In [68]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'D',ms=20.4)
pt.show()
```

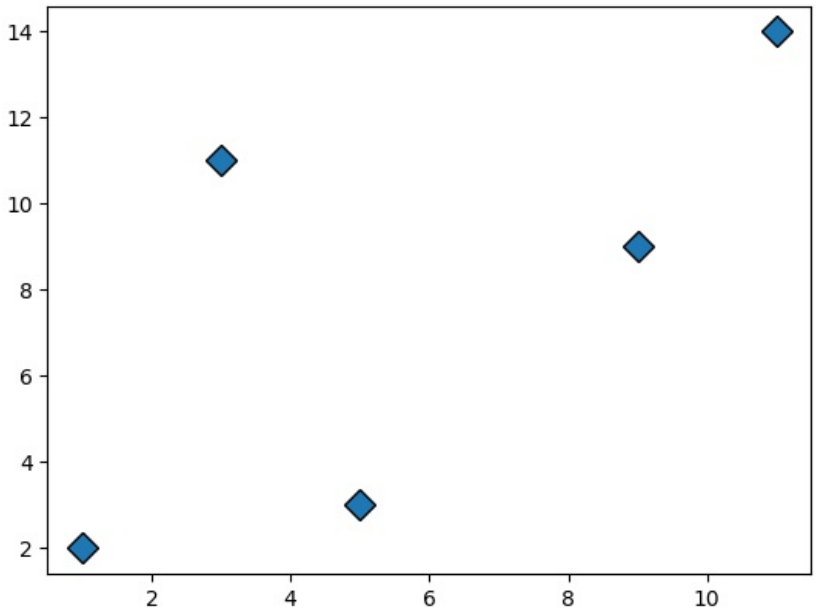


- `Marker Heilighter Color: (mec)`

In [69]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,marker='D',mec='r')
pt.show()
```

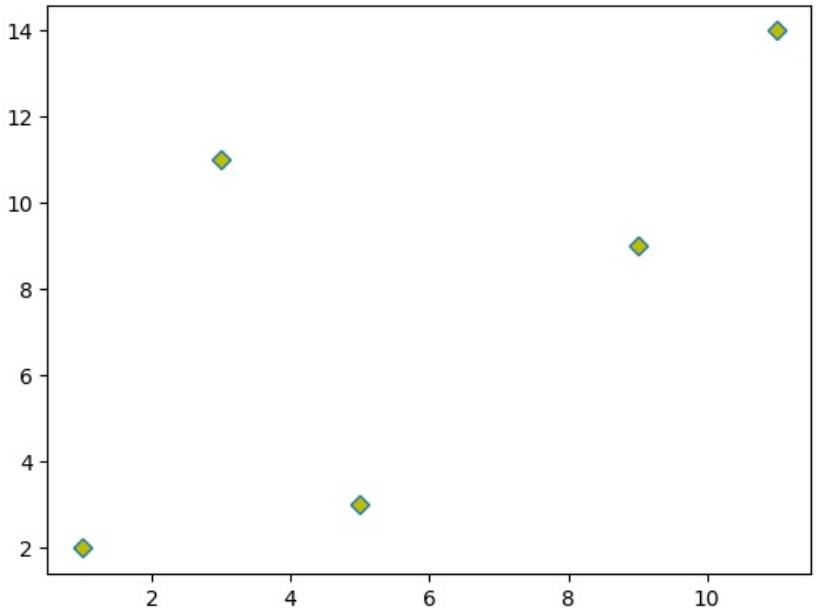


In [77]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'D',ms=10,mec='k')
pt.show()
```

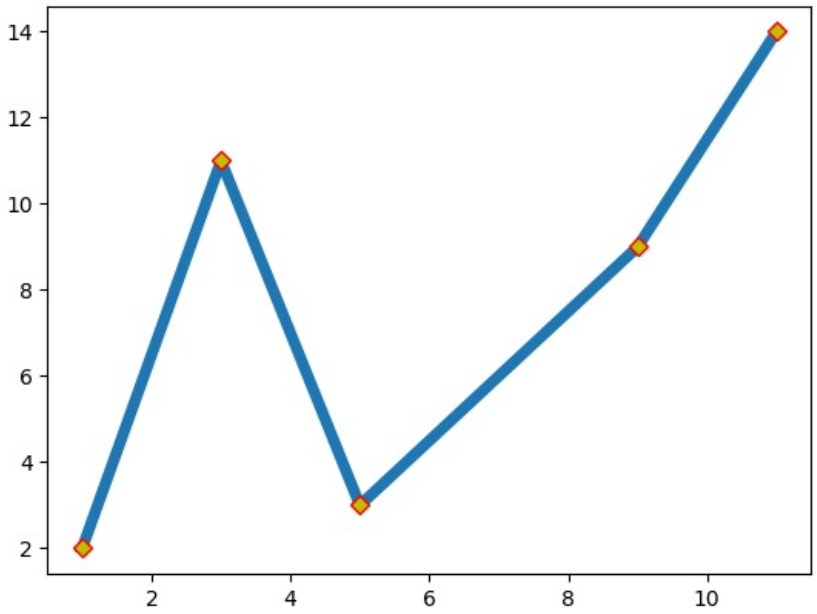


- `Marker Facce Color: (mfc)`

In [76]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,'D',mfc='y')
pt.show()
```

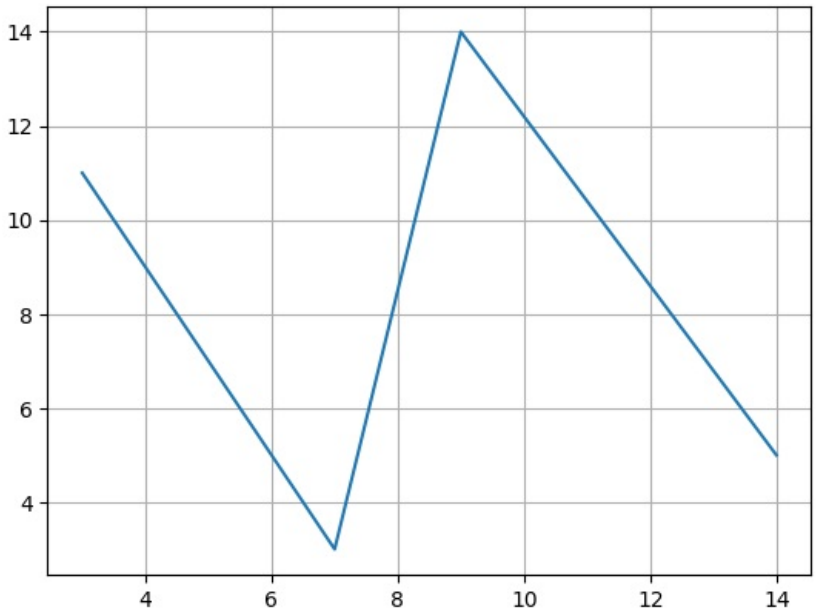


In [85]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5,9,11])
y = np.array([2,11,3,9,14])
pt.plot(x,y,marker='D',mec='r',mfc='y',linewidth=5.3)
pt.show()
```

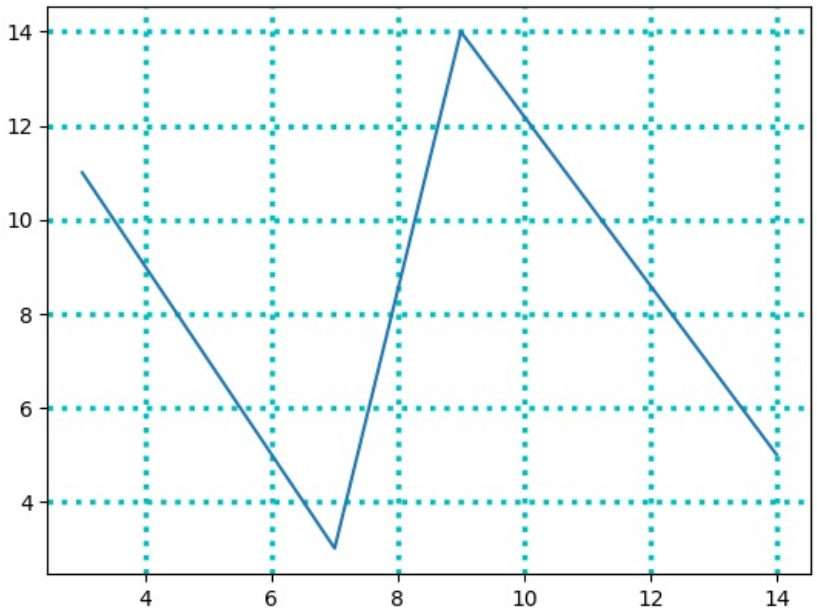


- `Grid: It provides Symmetrical graph as a background.`

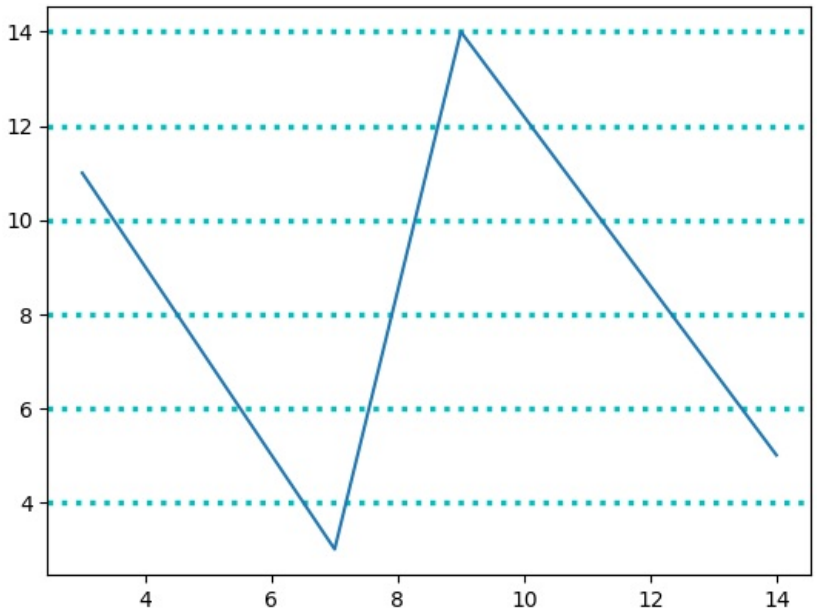
In [86]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([3,7,9,14])
y = np.array([11,3,14,5])
pt.plot(x,y)
pt.grid()
pt.show()
```



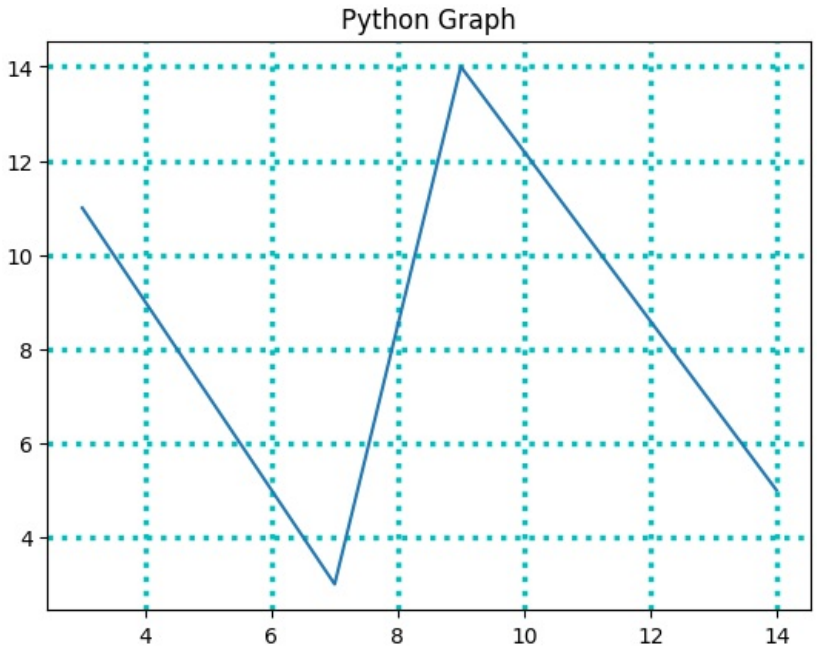
In [96]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([3,7,9,14])
y = np.array([11,3,14,5])
pt.plot(x,y)
pt.grid(c='c',ls=':',linewidth='2.5')
pt.show()
```



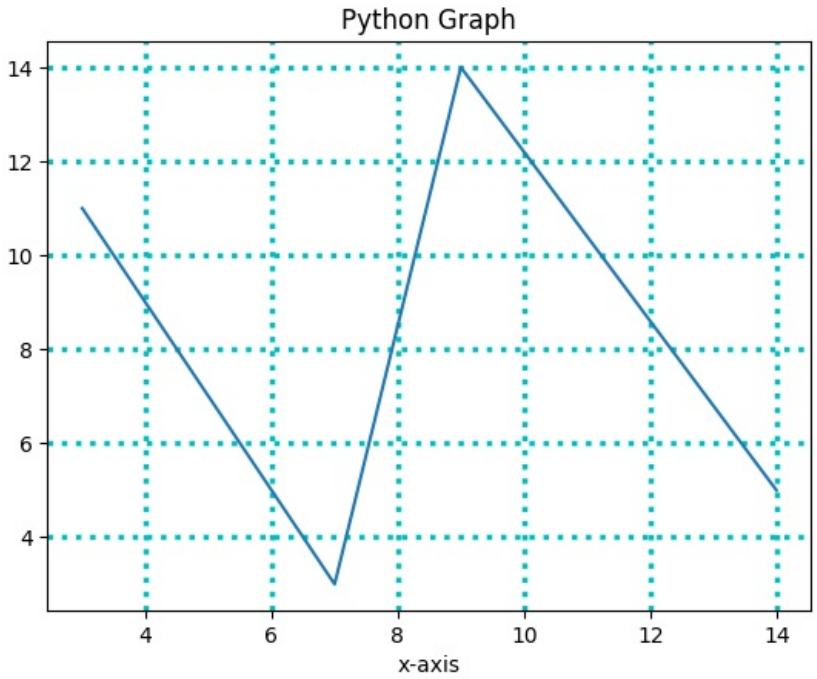
In [97]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([3,7,9,14])
y = np.array([11,3,14,5])
pt.plot(x,y)
pt.grid(c='c',ls=':',linewidth='2.5',axis='y')
pt.show()
```



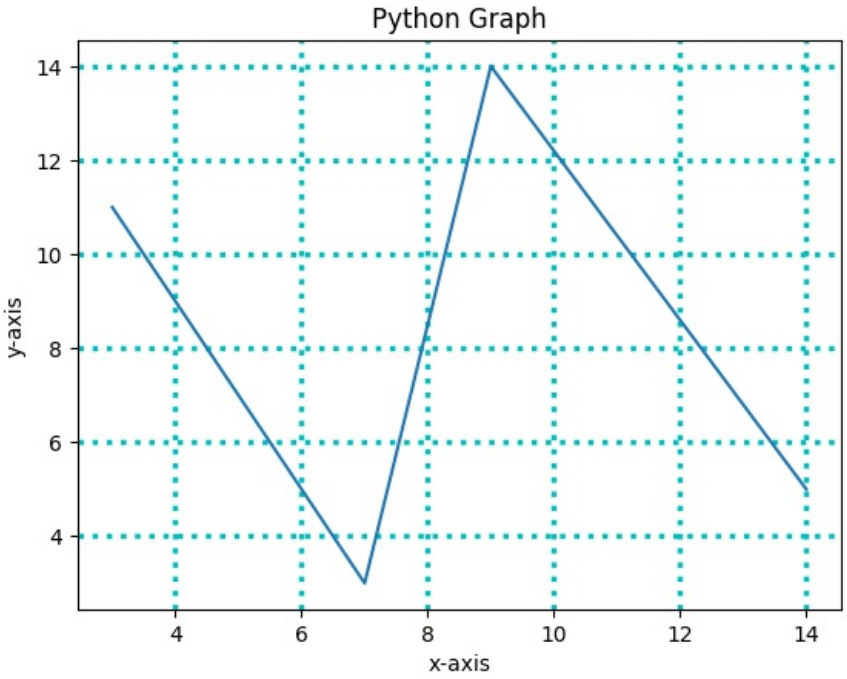
In [98]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([3,7,9,14])
y = np.array([11,3,14,5])
pt.plot(x,y)
pt.grid(c='c',ls=':',linewidth='2.5')
pt.title("Python Graph")
pt.show()
```



In [99]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([3,7,9,14])
y = np.array([11,3,14,5])
pt.plot(x,y)
pt.grid(c='c',ls=':',linewidth='2.5')
pt.title("Python Graph")
pt.xlabel("x-axis")
pt.show()
```

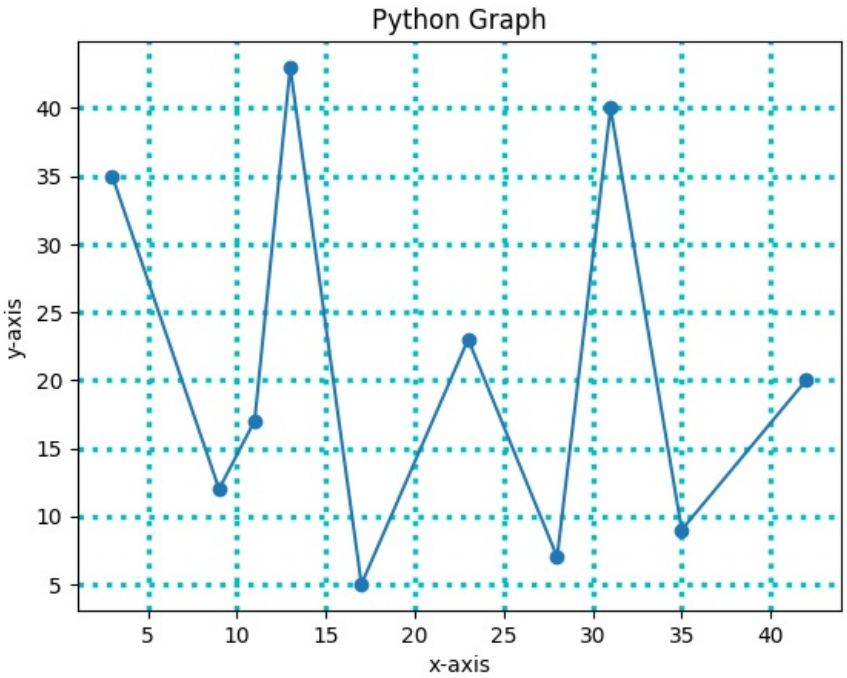


In [100…
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([3,7,9,14])
y = np.array([11,3,14,5])
pt.plot(x,y)
pt.grid(c='c',ls=':',linewidth='2.5')
pt.title("Python Graph")
pt.xlabel("x-axis")
pt.ylabel("y-axis")
pt.show()
```



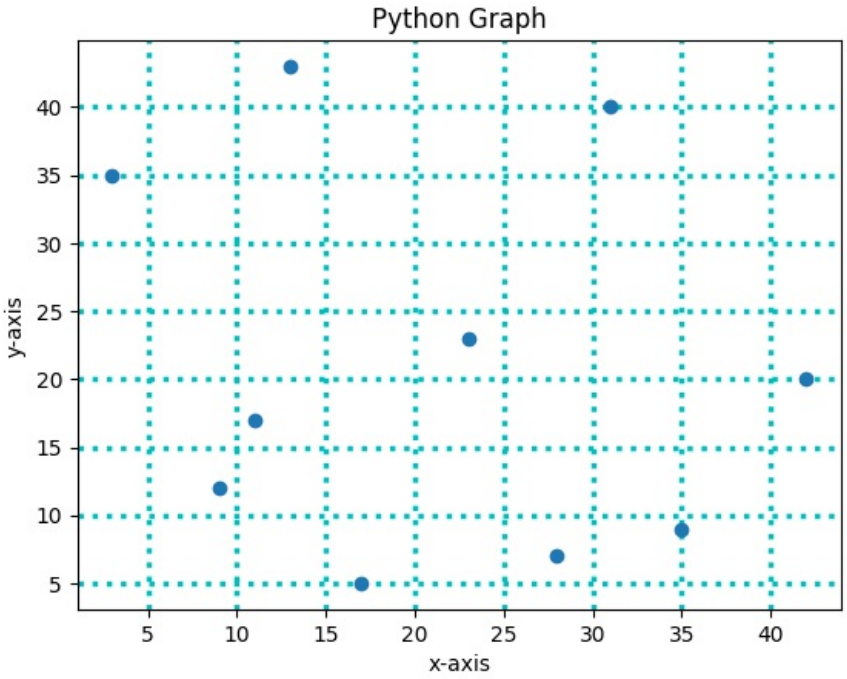
In [103…

```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([3,9,11,13,17,23,28,31,35,42])
y = np.array([35,12,17,43,5,23,7,40,9,20])
pt.title("Python Graph")
pt.xlabel("x-axis")
pt.ylabel("y-axis")
pt.plot(x,y,marker='o')
pt.grid(c='c',ls=':',linewidth='2.5')
pt.show()
```



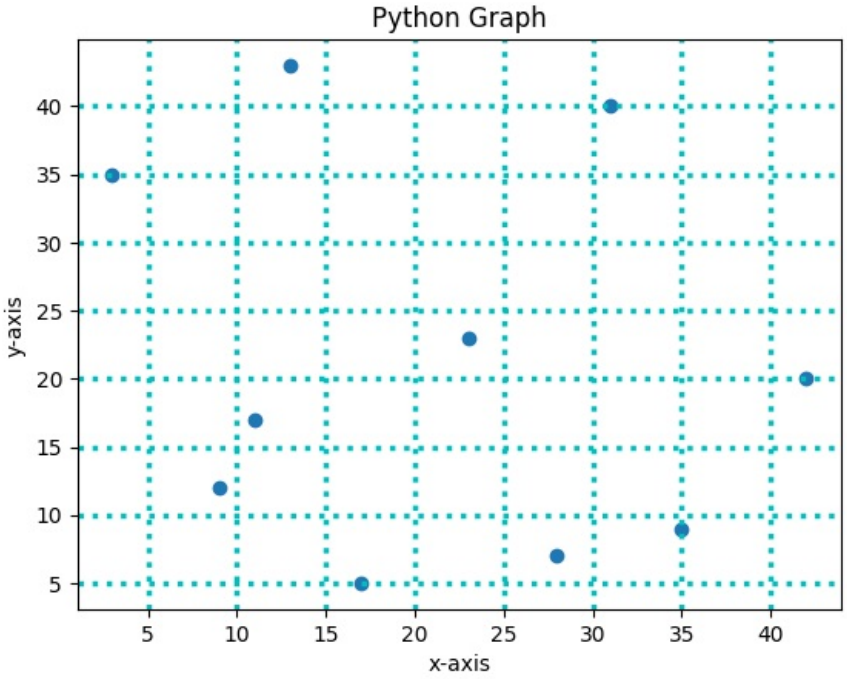
In [104…

```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([3,9,11,13,17,23,28,31,35,42])
y = np.array([35,12,17,43,5,23,7,40,9,20])
pt.title("Python Graph")
pt.xlabel("x-axis")
pt.ylabel("y-axis")
pt.plot(x,y,'o')
pt.grid(c='c',ls=':',linewidth='2.5')
pt.show()
```

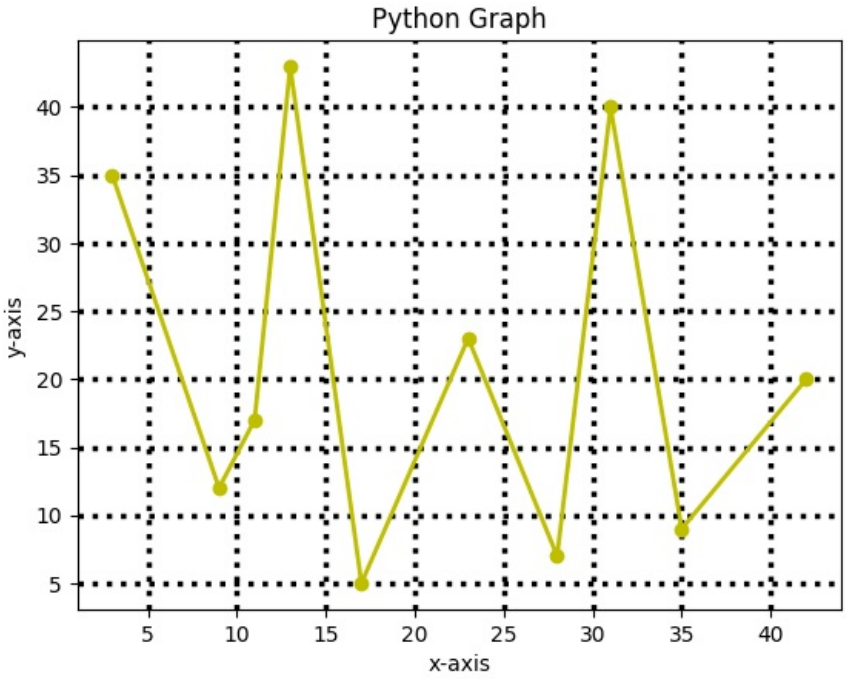


- `Scatter:`

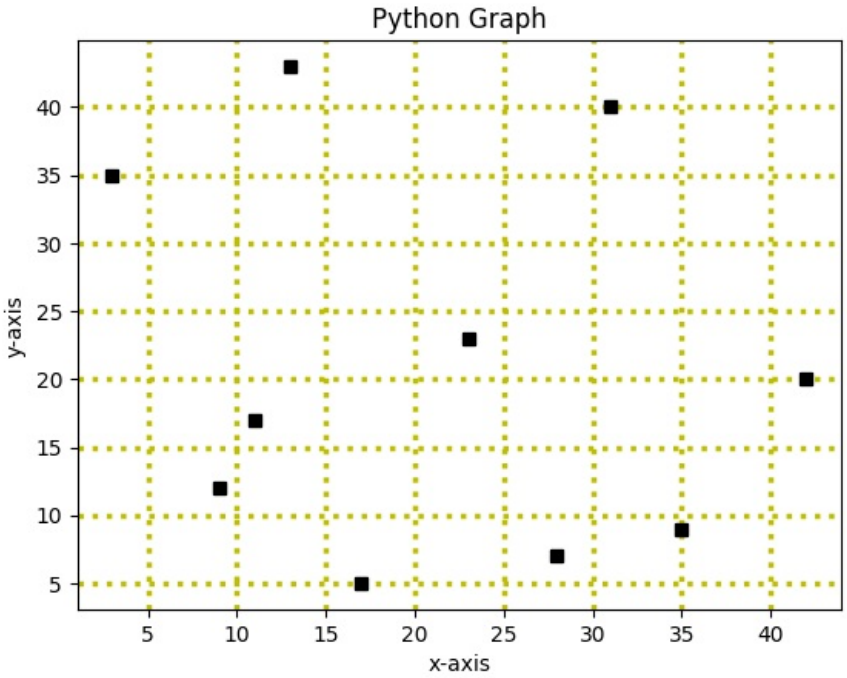
In [112…
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([3,9,11,13,17,23,28,31,35,42])
y = np.array([35,12,17,43,5,23,7,40,9,20])
pt.title("Python Graph")
pt.xlabel("x-axis")
pt.ylabel("y-axis")
pt.scatter(x,y,marker='o')
pt.grid(c='c',ls=':',linewidth='2.5')
pt.show()
```



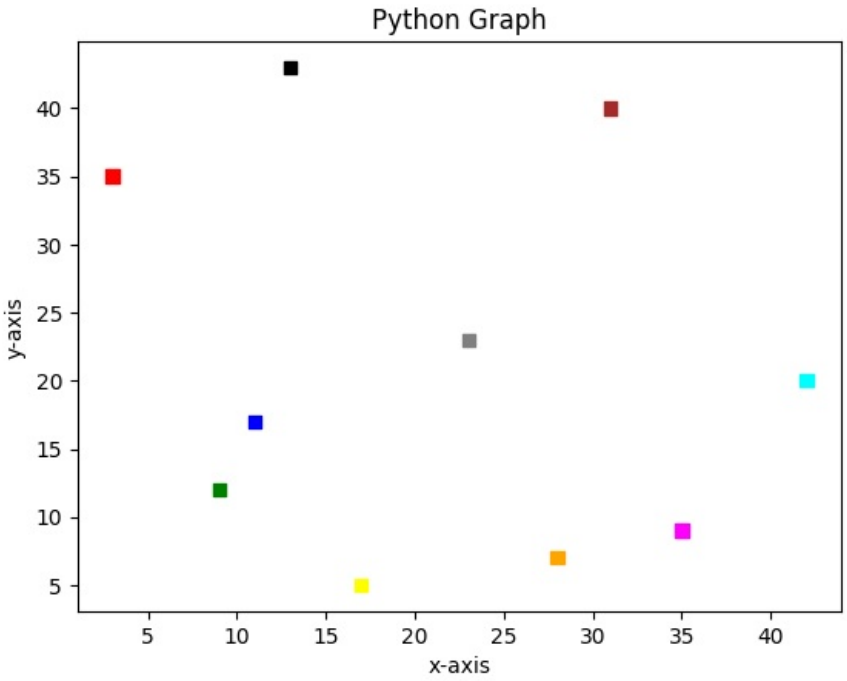
In [126…
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([3,9,11,13,17,23,28,31,35,42])
y = np.array([35,12,17,43,5,23,7,40,9,20])
colors = np.array(['red','green','blue','black','yellow','gray','orange','brown','magenta','cyan'])
pt.title("Python Graph")
pt.xlabel("x-axis")
pt.ylabel("y-axis")
pt.plot(x,y,marker='o',c='y',linewidth=2)
pt.grid(c='k',ls=':',linewidth='2.5')
pt.show()
```



In [129…
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([3,9,11,13,17,23,28,31,35,42])
y = np.array([35,12,17,43,5,23,7,40,9,20])
colors = np.array(['red','green','blue','black','yellow','gray','orange','brown','magenta','cyan'])
pt.title("Python Graph")
pt.xlabel("x-axis")
pt.ylabel("y-axis")
pt.plot(x,y,'s',c='k',linewidth=2)
pt.grid(c='y',ls=':',linewidth='2.5')
pt.show()
```



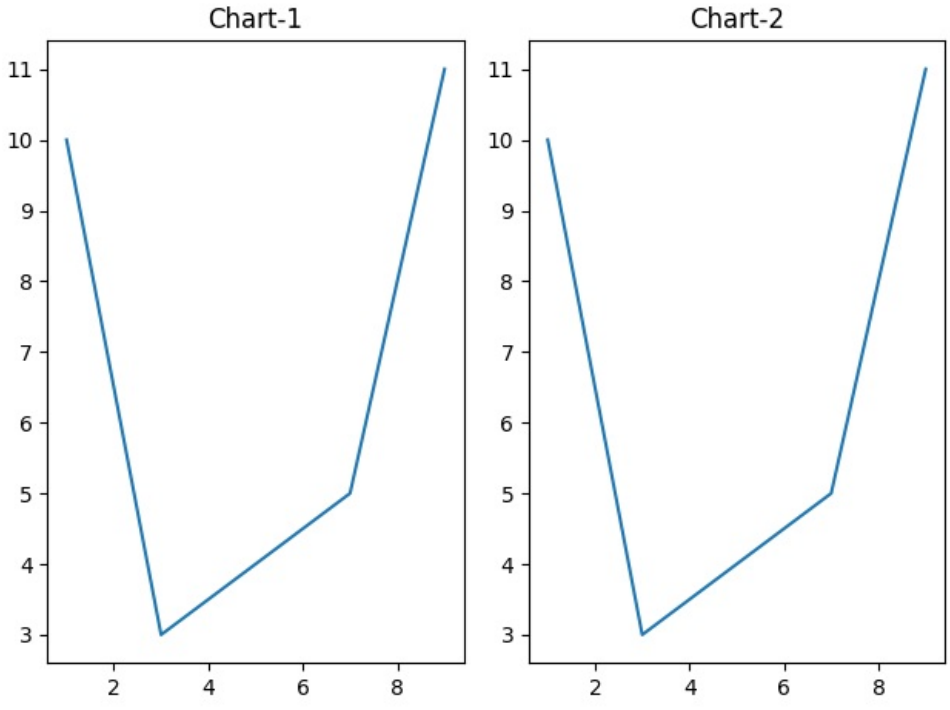
In [119…
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([3,9,11,13,17,23,28,31,35,42])
y = np.array([35,12,17,43,5,23,7,40,9,20])
colors = np.array(['red','green','blue','black','yellow','gray','orange','brown','magenta','cyan'])
pt.title("Python Graph")
pt.xlabel("x-axis")
pt.ylabel("y-axis")
# pt.plot(x,y,marker='o',c='y',linewidth='2')
pt.scatter(x,y,marker='D',)
# pt.grid(c='k',ls=':',linewidth='2.5')
pt.show()
```



Matplotlib (subplot) V25.02.13

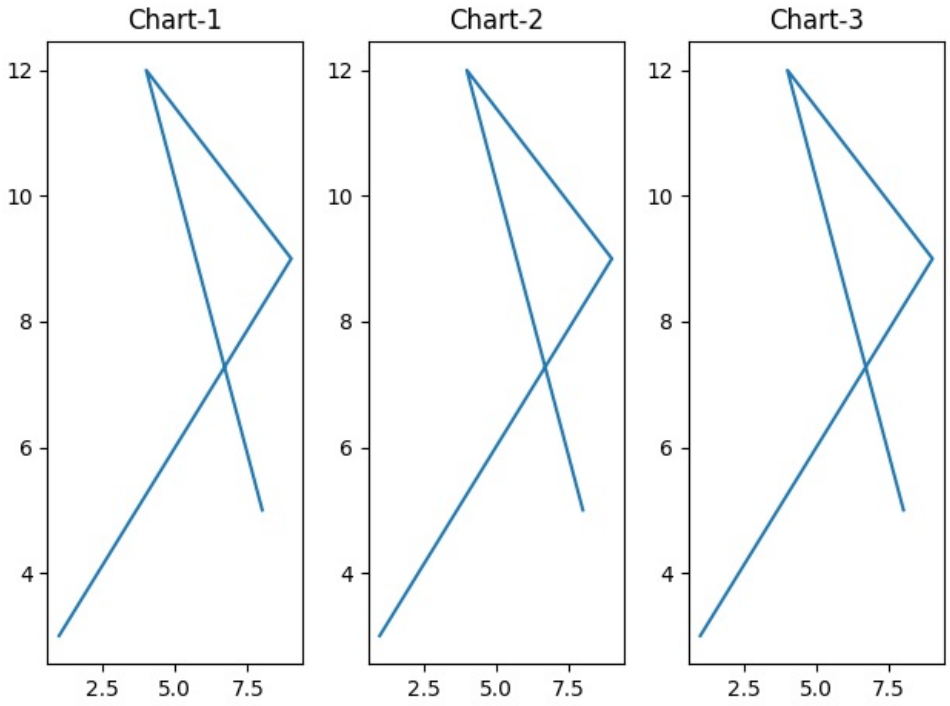
- `subplot: syntax: pt.subplot(row,column,position)`

In [30]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,7,9])
y = np.array([10,3,5,11])
pt.subplot(1,2,1)
pt.title("Graph-1")
pt.plot(x,y)
a = np.array([1,3,7,9])
b = np.array([10,3,5,11])
pt.subplot(1,2,2)
pt.plot(a,b)
pt.title("Graph-2")
pt.tight_layout()
pt.show()
```



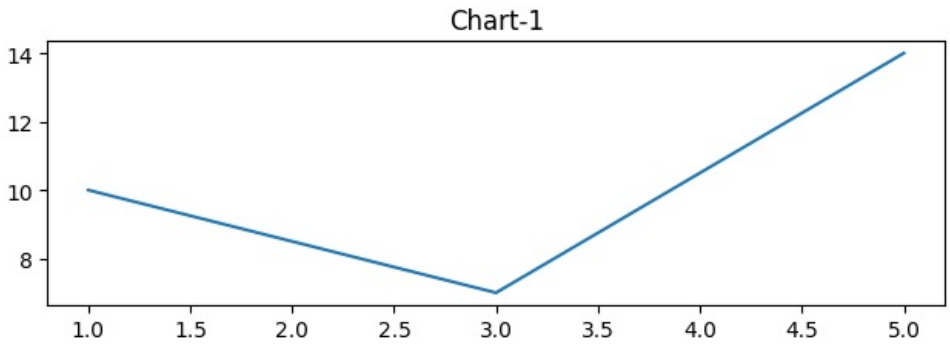
In [33]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,9,4,8])
y = np.array([3,9,12,5])
pt.subplot(1,3,1)
pt.plot(x,y)
```

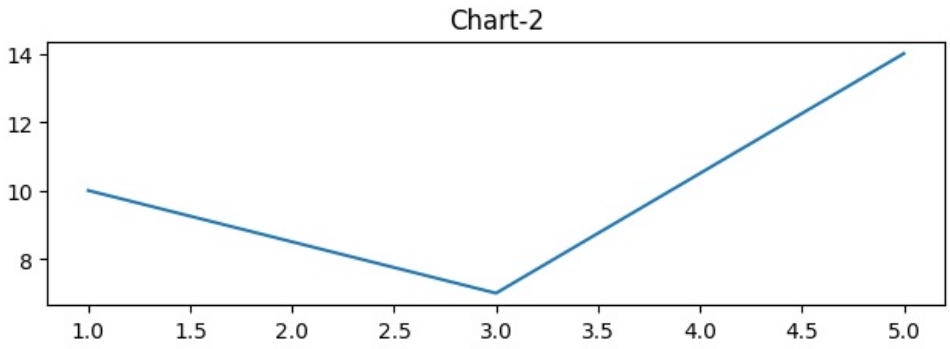
```
pt.title("Graph-1")
a = np.array([1,9,4,8])
b = np.array([3,9,12,5])
pt.subplot(1,3,2)
pt.plot(a,b)
pt.title("Graph-2")
p = np.array([1,9,4,8])
q = np.array([3,9,12,5])
pt.subplot(1,3,3)
pt.plot(p,q)
pt.title("Graph-3")
pt.tight_layout()
pt.show()
```



In [34]:

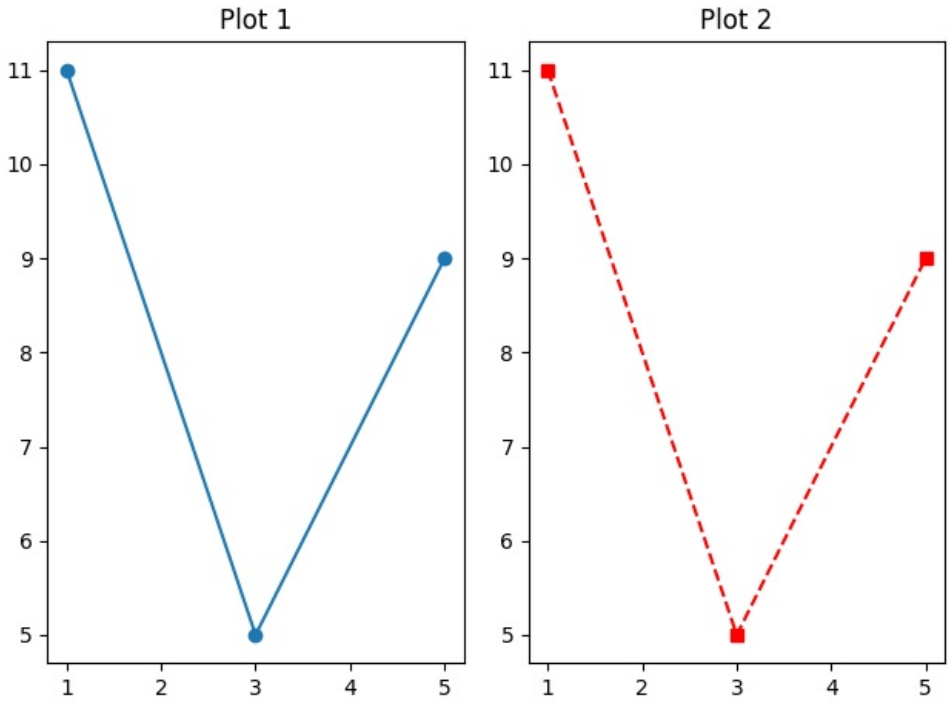
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5])
y = np.array([10,7,14])
pt.subplot(2,1,1)
pt.plot(x,y)
pt.title("Graph-1")
a = np.array([1,3,5])
b = np.array([10,7,14])
pt.subplot(2,1,2)
pt.plot(a,b)
pt.title("Graph-2")
pt.tight_layout()
pt.show()
```



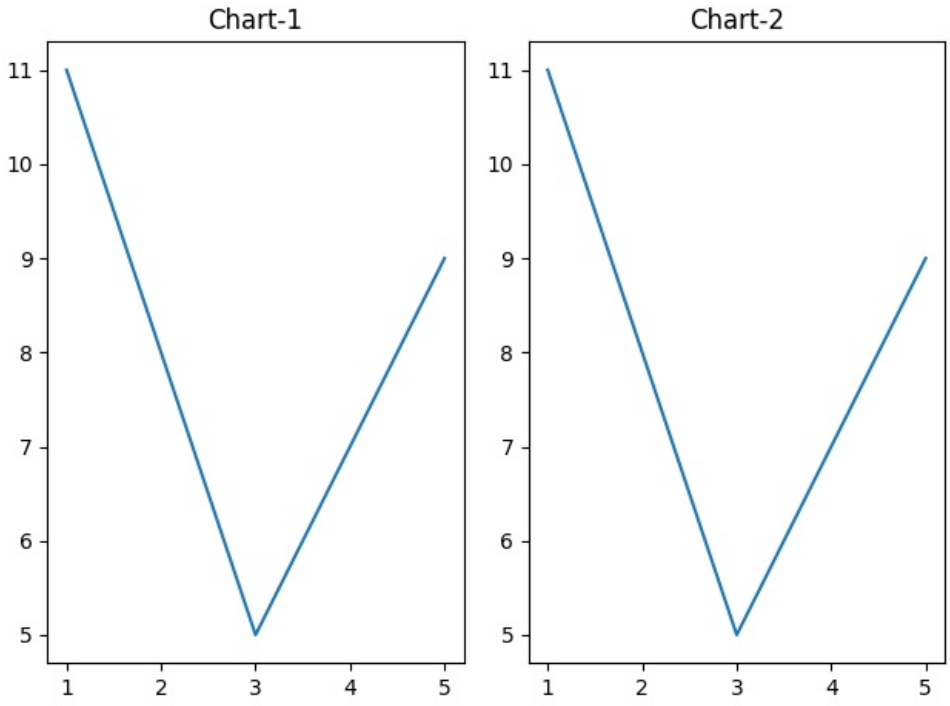


- `Demo` `Experiment`

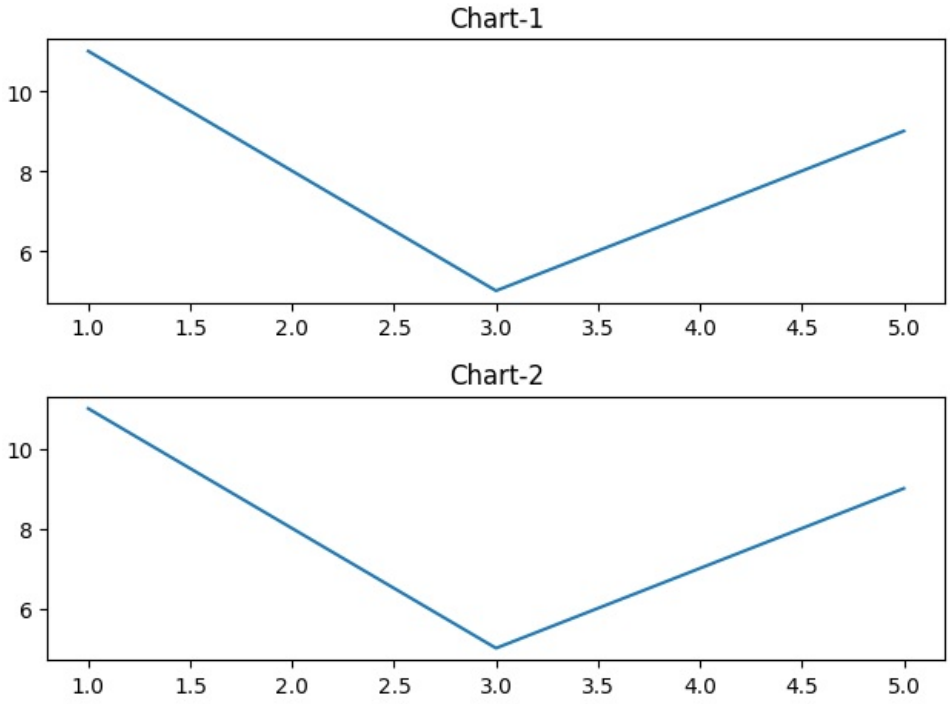
In [35]:
```
import matplotlib.pyplot as pt
import numpy as np

x = np.array([1, 3, 5])
y = np.array([11, 5, 9])

# First subplot (1 row, 2 columns, 1st plot)
pt.subplot(1, 2, 1)
pt.plot(x, y, marker='o', linestyle='-')
pt.title("Plot 1")

a = np.array([1, 3, 5])
b = np.array([11, 5, 9])

# Second subplot (1 row, 2 columns, 2nd plot)
pt.subplot(1, 2, 2)
pt.plot(a, b, marker='s', linestyle='--', color='red')
pt.title("Plot 2")

# Show all plots together
pt.tight_layout()  # Adjust layout for better spacing
pt.show()
```



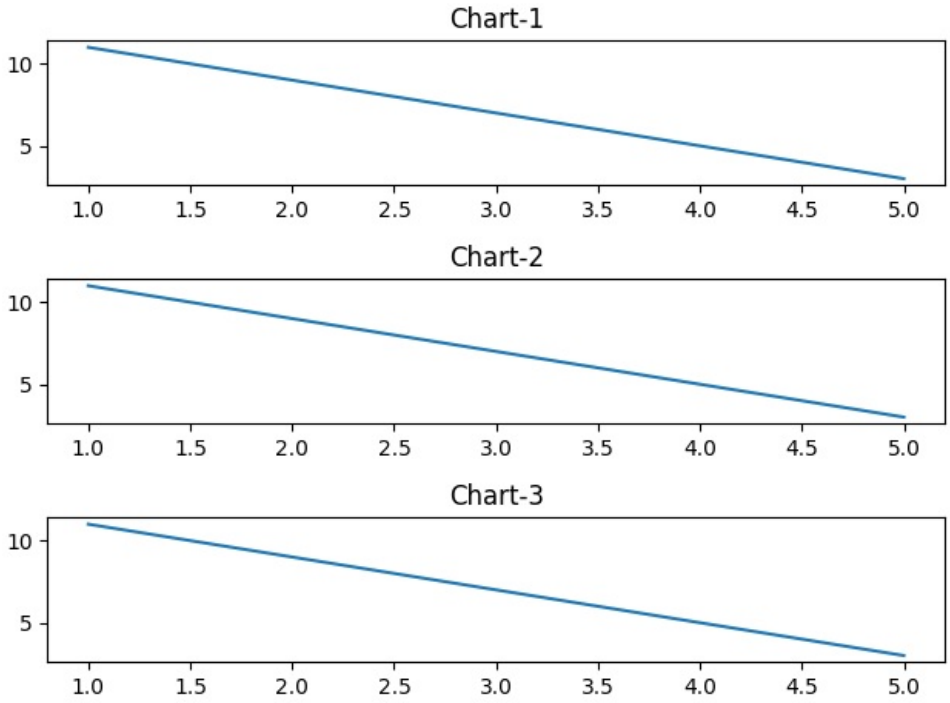
In [43]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5])
y = np.array([11,5,9])
pt.subplot(1,2,1)
pt.plot(x,y)
pt.title("Graph-1")
a = np.array([1,3,5])
b = np.array([11,5,9])
pt.subplot(1,2,2)
pt.plot(x,y)
pt.title("Graph-2")
pt.tight_layout()
pt.show()
```



In [44]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5])
y = np.array([11,5,9])
pt.subplot(2,1,1)
pt.plot(x,y)
pt.title("Graph-1")
a = np.array([1,3,5])
b = np.array([11,5,9])
pt.subplot(2,1,2)
pt.plot(x,y)
pt.title("Graph-2")
pt.tight_layout()
pt.show()
```

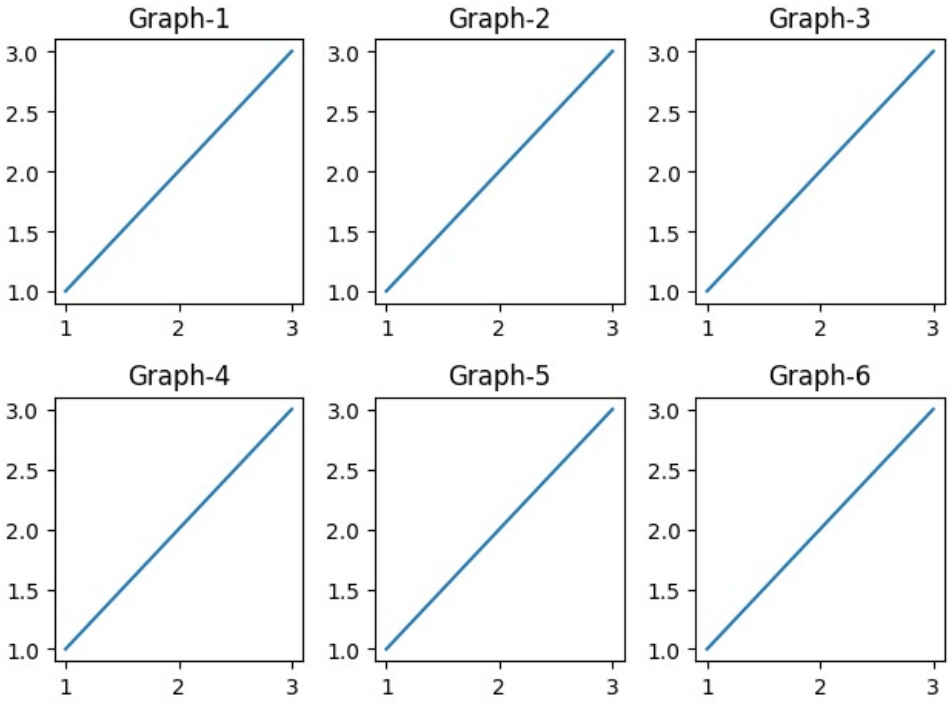


In [47]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,3,5])
y = np.array([11,7,3])
pt.subplot(3,1,1)
pt.plot(x,y)
pt.title("Graph-1")
a = np.array([1,3,5])
b = np.array([11,7,3])
pt.subplot(3,1,2)
pt.plot(x,y)
pt.title("Graph-2")
p = np.array([1,3,5])
q = np.array([11,7,3])
pt.subplot(3,1,3)
pt.plot(x,y)
pt.title("Graph-3")
pt.tight_layout()
pt.show()
```



In [56]:
```
import matplotlib.pyplot as pt
import numpy as np
a = np.array([1,2,3])
b = np.array([1,2,3])
pt.subplot(2,3,1)
pt.plot(a,b)
```

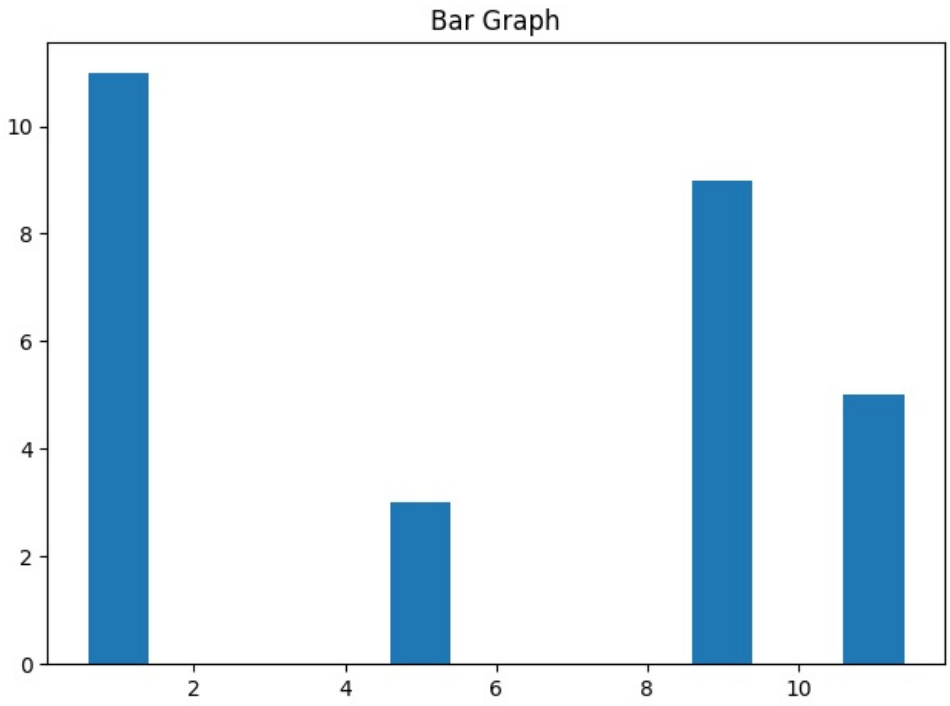
```
pt.title("Graph-1")
c = np.array([1,2,3])
d = np.array([1,2,3])
pt.subplot(2,3,2)
pt.plot(c,d)
pt.title("Graph-2")
e = np.array([1,2,3])
f = np.array([1,2,3])
pt.subplot(2,3,3)
pt.plot(e,f)
pt.title("Graph-3")
g = np.array([1,2,3])
h = np.array([1,2,3])
pt.subplot(2,3,4)
pt.plot(g,h)
pt.title("Graph-4")
i = np.array([1,2,3])
j = np.array([1,2,3])
pt.subplot(2,3,5)
pt.plot(i,j)
pt.title("Graph-5")
k = np.array([1,2,3])
l = np.array([1,2,3])
pt.subplot(2,3,6)
pt.plot(k,l)
pt.title("Graph-6")
pt.tight_layout()
pt.show()
```



Bar Graph

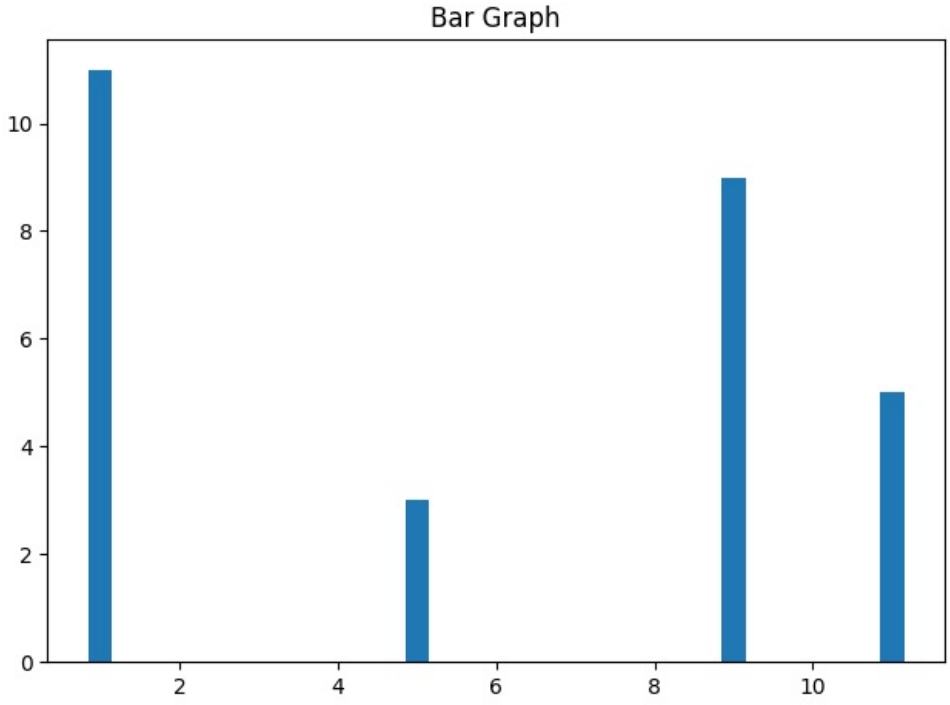
- Bar Graph: syntax: pt.bar(x,y)
- Verticle: Bar: Accepts width
- Horizontal: Barh: Accepts Height
- Default height and width of bar is 0.8.
- Bar Graph: Separate Bars
- Histogram Graph: Joint Bars. Use S.D.

In [73]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,5,9,11])
y = np.array([11,3,9,5])
pt.bar(x,y)
pt.title("Bar Graph")
pt.tight_layout()
pt.show()
```



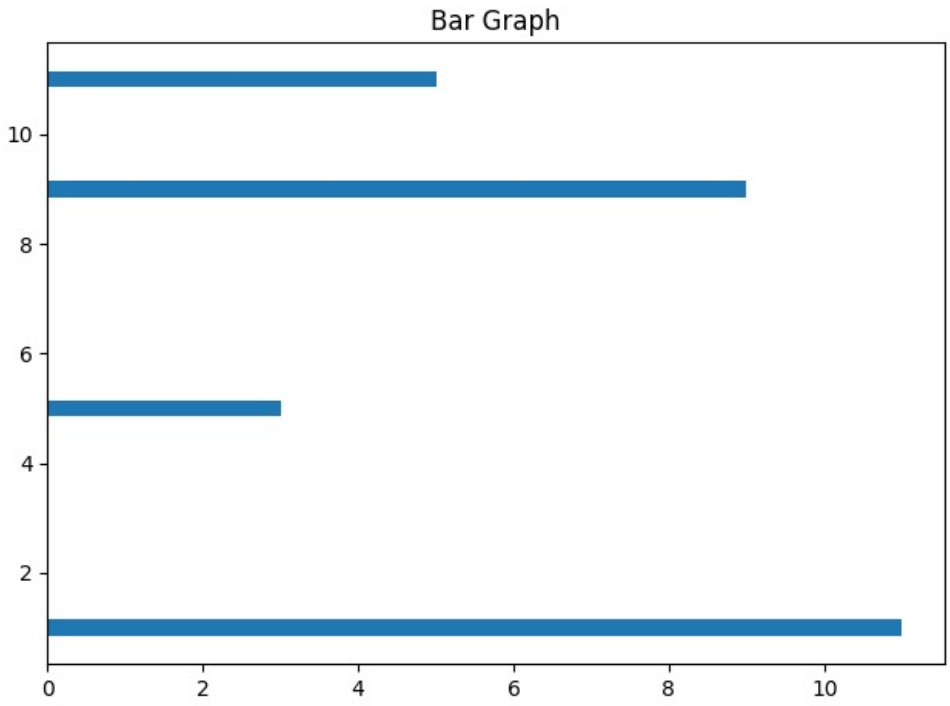
- `bar : width`

In [74]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,5,9,11])
y = np.array([11,3,9,5])
pt.bar(x,y,width=0.3)
pt.title("Bar Graph")
pt.tight_layout()
pt.show()
```

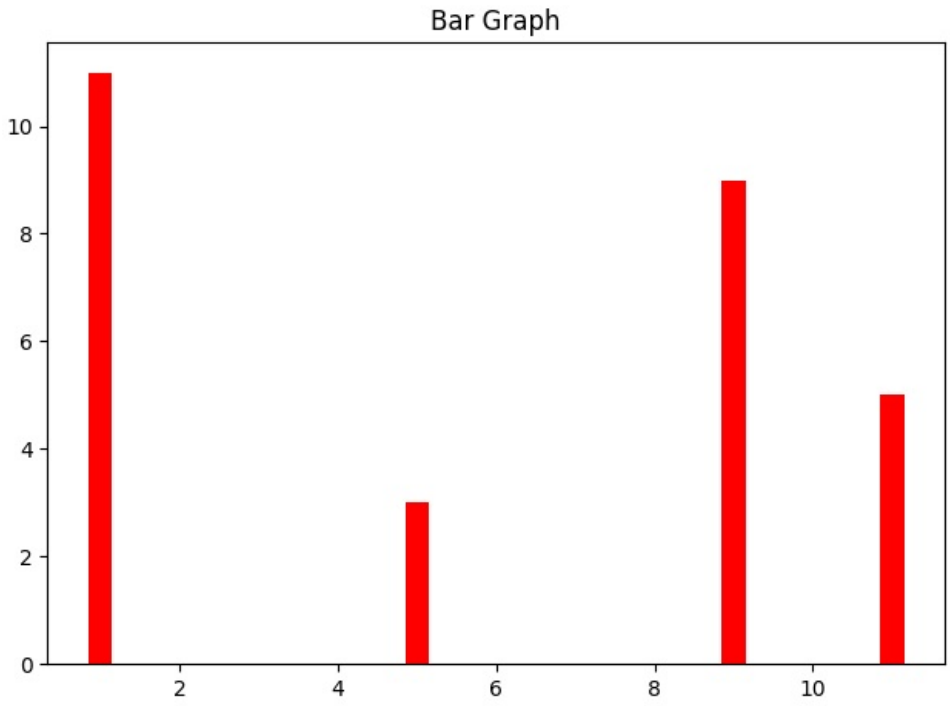


- `barh : height`

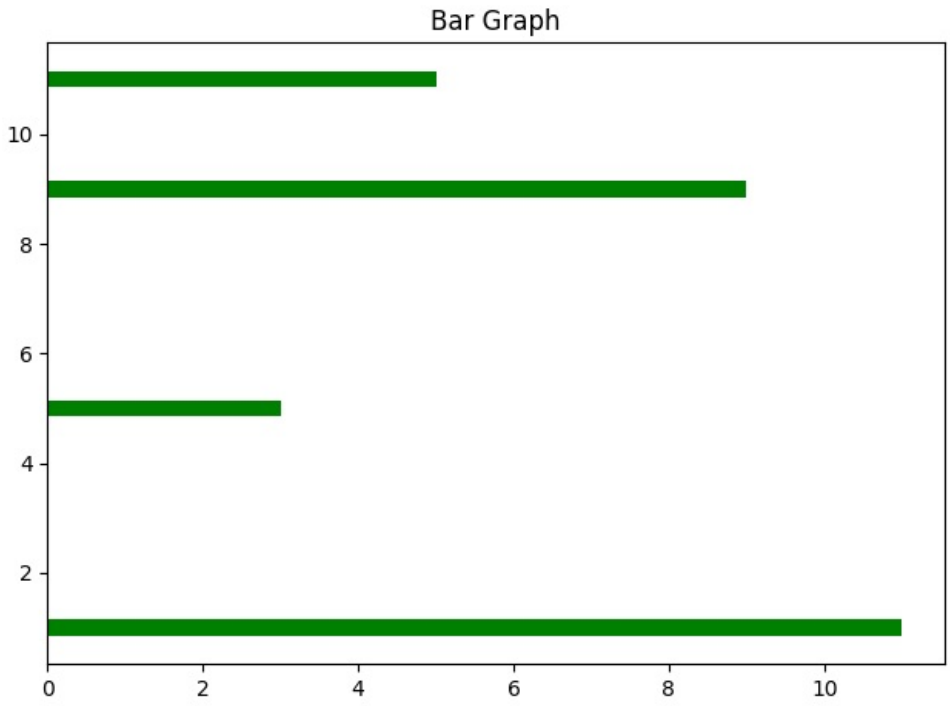
In [71]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,5,9,11])
y = np.array([11,3,9,5])
pt.barh(x,y,height=0.3)
pt.title("Bar Graph")
pt.tight_layout()
pt.show()
```



In [72]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,5,9,11])
y = np.array([11,3,9,5])
pt.bar(x,y,width=0.3, color='red')
pt.title("Bar Graph")
pt.tight_layout()
pt.show()
```



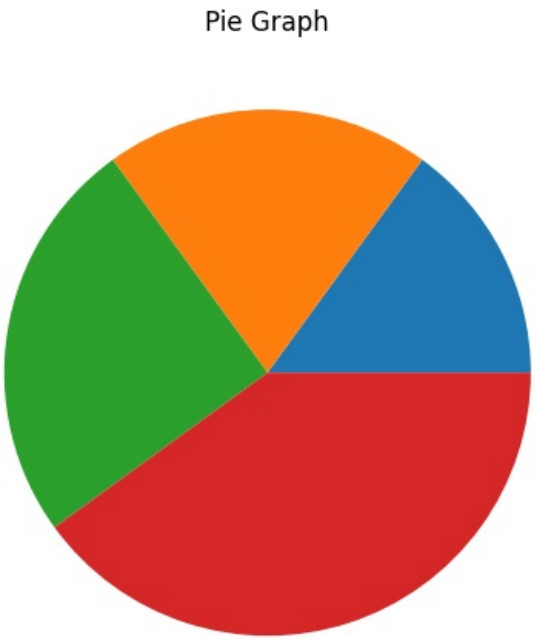
In [66]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([1,5,9,11])
y = np.array([11,3,9,5])
pt.barh(x,y,height=0.3,color='green')
pt.title("Bar Graph")
pt.tight_layout()
pt.show()
```



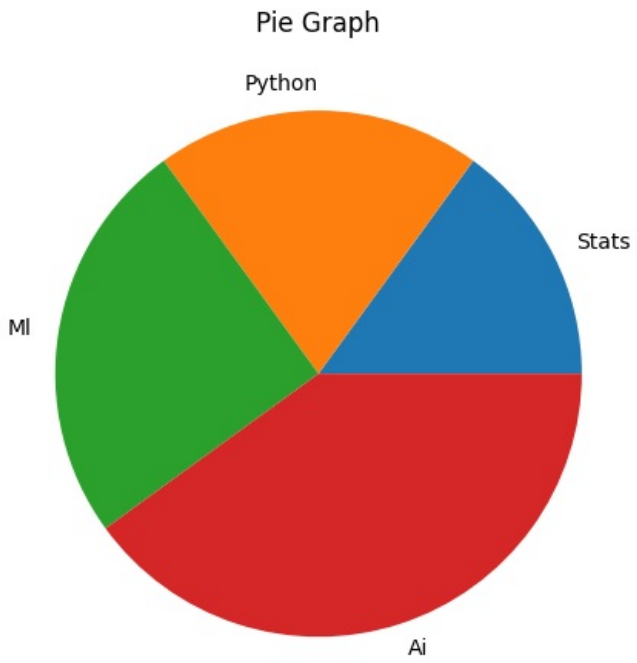
Pie Graph:

- `0 degree`
- `Anticlockwise`
- `Position = Specific Value / Total Addition of all Values`
- `Default colors: blue, orange, green, red.`

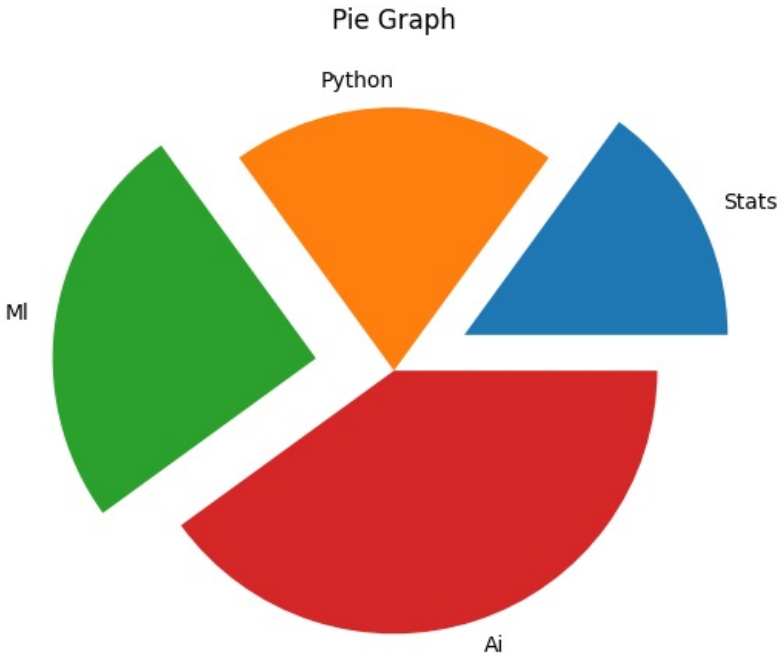
In [76]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([15,20,25,40])
pt.pie(x)
pt.title("Pie Graph")
pt.tight_layout()
pt.show()
```



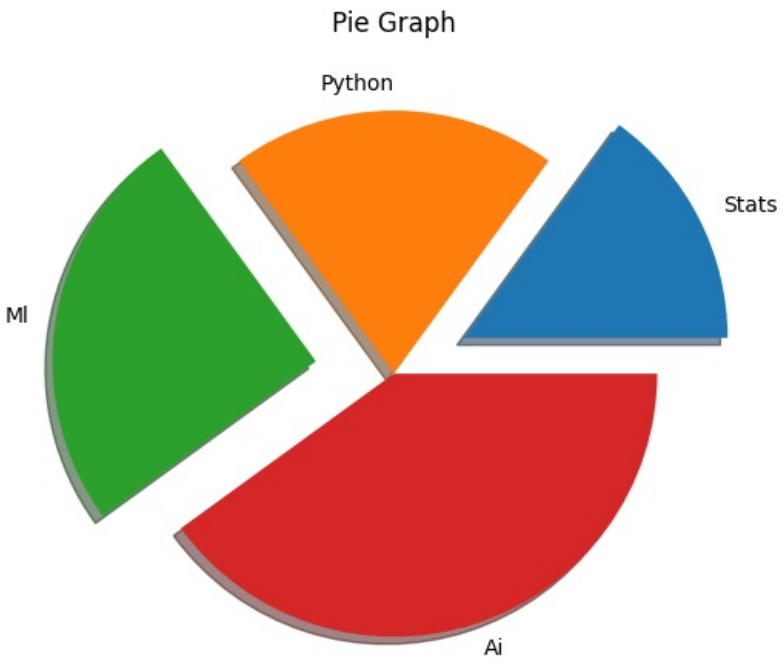
In [77]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([15,20,25,40])
users = np.array(["Stats","Python","Ml","Ai"])
pt.pie(x,labels=users)
pt.title("Pie Graph")
pt.tight_layout()
pt.show()
```



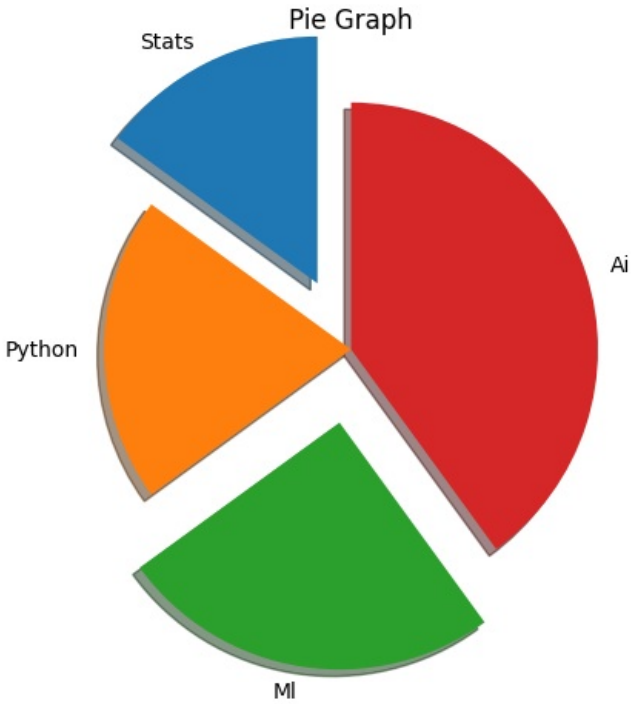
In [80]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([15,20,25,40])
users = np.array(["Stats","Python","Ml","Ai"])
ex = np.array([0.3,0,0.3,0])
pt.pie(x,labels=users,explode=ex)
pt.title("Pie Graph")
pt.tight_layout()
pt.show()
```



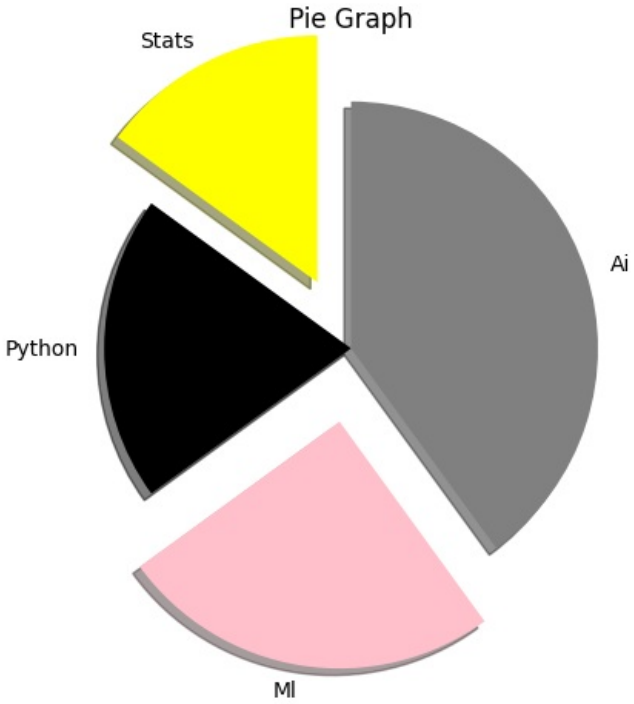
In [81]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([15,20,25,40])
users = np.array(["Stats","Python","Ml","Ai"])
ex = np.array([0.3,0,0.3,0])
pt.pie(x,labels=users,explode=ex, shadow=True)
pt.title("Pie Graph")
pt.tight_layout()
pt.show()
```



In [82]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([15,20,25,40])
users = np.array(["Stats","Python","Ml","Ai"])
ex = np.array([0.3,0,0.3,0])
pt.pie(x, labels=users, explode=ex, shadow=True, startangle=90)
pt.title("Pie Graph")
pt.tight_layout()
pt.show()
```



In [84]:
```
import matplotlib.pyplot as pt
import numpy as np
x = np.array([15,20,25,40])
users = np.array(["Stats","Python","Ml","Ai"])
ex = np.array([0.3,0,0.3,0])
colors = np.array(["yellow","black","pink","grey"])
pt.pie(x, labels=users, explode=ex, shadow=True, startangle=90, colors=colors)
pt.title("Pie Graph")
pt.tight_layout()
pt.show()
```



Panda

- Python Library(Packase) for Data Operations.
- import panda as pd
- mostly supports csv files
- DataFrames: Return 1st and last 5 Records of data
- Path: Download -> move to desktop -> Right click -> Properties -> Copy path
- pd.read_csv("Path") (convert (\)Backslash -> (/)Forwordslash and add file name with .csv) eg,. C:/Users/majag/Desktop/data.csv
- .to_string(): Return all data
- .info(): Information of File
- .plot

In [4]:
```
pip install pandas
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting pandas
  Downloading pandas-2.2.3-cp312-cp312-win_amd64.whl.metadata (19 kB)
Requirement already satisfied: numpy>=1.26.0 in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from pandas) (2.9.0.post0)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2025.1-py2.py3-none-any.whl.metadata (22 kB)
Requirement already satisfied: tzdata>=2022.7 in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from pandas) (2024.1)
Requirement already satisfied: six>=1.5 in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Downloading pandas-2.2.3-cp312-cp312-win_amd64.whl (11.5 MB)
   ---------------------------------------- 0.0/11.5 MB ? eta -:--:--
   ---------------------------------------- 0.0/11.5 MB ? eta -:--:--
    --------------------------------------- 0.3/11.5 MB ? eta -:--:--
   -- ------------------------------------- 0.8/11.5 MB 1.6 MB/s eta 0:00:07
   --- ------------------------------------ 1.0/11.5 MB 1.6 MB/s eta 0:00:07
   ---- ----------------------------------- 1.3/11.5 MB 1.6 MB/s eta 0:00:07
   ------ --------------------------------- 1.8/11.5 MB 1.6 MB/s eta 0:00:06
   ------ --------------------------------- 1.8/11.5 MB 1.6 MB/s eta 0:00:06
   -------- ------------------------------- 2.4/11.5 MB 1.5 MB/s eta 0:00:07
   --------- ------------------------------ 2.6/11.5 MB 1.5 MB/s eta 0:00:06
   ---------- ----------------------------- 3.1/11.5 MB 1.6 MB/s eta 0:00:06
   ----------- ---------------------------- 3.4/11.5 MB 1.6 MB/s eta 0:00:06
   ------------ --------------------------- 3.7/11.5 MB 1.6 MB/s eta 0:00:05
   ------------ --------------------------- 3.7/11.5 MB 1.6 MB/s eta 0:00:05
   -------------- ------------------------- 4.2/11.5 MB 1.5 MB/s eta 0:00:05
   --------------- ------------------------ 4.5/11.5 MB 1.5 MB/s eta 0:00:05
   ---------------- ----------------------- 4.7/11.5 MB 1.5 MB/s eta 0:00:05
   ----------------- ---------------------- 5.0/11.5 MB 1.4 MB/s eta 0:00:05
   ------------------ --------------------- 5.5/11.5 MB 1.5 MB/s eta 0:00:05
   ------------------- -------------------- 5.8/11.5 MB 1.5 MB/s eta 0:00:04
   --------------------- ------------------ 6.3/11.5 MB 1.5 MB/s eta 0:00:04
   ---------------------- ----------------- 6.8/11.5 MB 1.6 MB/s eta 0:00:03
   ------------------------ --------------- 7.3/11.5 MB 1.6 MB/s eta 0:00:03
   ------------------------- -------------- 7.6/11.5 MB 1.6 MB/s eta 0:00:03
   --------------------------- ------------ 8.1/11.5 MB 1.6 MB/s eta 0:00:03
   ---------------------------- ----------- 8.4/11.5 MB 1.6 MB/s eta 0:00:02
   ----------------------------- ---------- 8.7/11.5 MB 1.6 MB/s eta 0:00:02
   ------------------------------ --------- 9.2/11.5 MB 1.6 MB/s eta 0:00:02
   ------------------------------- -------- 9.4/11.5 MB 1.6 MB/s eta 0:00:02
   --------------------------------- ------ 10.0/11.5 MB 1.6 MB/s eta 0:00:01
   --------------------------------- ------ 10.0/11.5 MB 1.6 MB/s eta 0:00:01
   ---------------------------------- ----- 10.5/11.5 MB 1.6 MB/s eta 0:00:01
   ----------------------------------- ---- 11.0/11.5 MB 1.6 MB/s eta 0:00:01
   ----------------------------------- ---- 11.3/11.5 MB 1.6 MB/s eta 0:00:01
   ---------------------------------------- 11.5/11.5 MB 1.6 MB/s eta 0:00:00
Downloading pytz-2025.1-py2.py3-none-any.whl (507 kB)
Installing collected packages: pytz, pandas
Successfully installed pandas-2.2.3 pytz-2025.1
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
pip install --upgrade pip
```

```
Defaulting to user installation because normal site-packages is not writeableNote: you may need to restart the kernel to use updated packages.

Requirement already satisfied: pip in c:\users\majag\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (24.2)
Collecting pip
  Downloading pip-25.0.1-py3-none-any.whl.metadata (3.7 kB)
Downloading pip-25.0.1-py3-none-any.whl (1.8 MB)
   ---------------------------------------- 0.0/1.8 MB ? eta -:--:--
   ---------------------------------------- 0.0/1.8 MB ? eta -:--:--
   ----- ---------------------------------- 0.3/1.8 MB ? eta -:--:--
   ---------------- ----------------------- 0.8/1.8 MB 2.0 MB/s eta 0:00:01
   ---------------------- ----------------- 1.0/1.8 MB 1.9 MB/s eta 0:00:01
   ---------------------------------- ----- 1.6/1.8 MB 2.0 MB/s eta 0:00:01
   ---------------------------------------- 1.8/1.8 MB 1.8 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 24.2
    Uninstalling pip-24.2:
      Successfully uninstalled pip-24.2
Successfully installed pip-25.0.1
```

In [10]:
```
import pandas as pd
# path1 = C:\Users\majag\Desktop\GitHub ⚒\Python ⚒
# path2 = C:\Users\majag\Desktop
data = pd.read_csv("C:/Users/majag/Desktop/data.csv")
view = pd.DataFrame(data)
```

```
print(view)
```

```
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
..        ...    ...       ...       ...
164        60    105       140     290.8
165        60    110       145     300.0
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4

[169 rows x 4 columns]
```

In [11]:
```
import pandas as pd
# path1 = C:\Users\majag\Desktop\GitHub ⚒\Python ⚒
# path2 = C:\Users\majag\Desktop
data = pd.read_csv("C:/Users/majag/Desktop/data.csv")
view = pd.DataFrame(data)
print(view.to_string())
```

```
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
5          60    102       127     300.0
6          60    110       136     374.0
7          45    104       134     253.3
8          30    109       133     195.1
9          60     98       124     269.0
10         60    103       147     329.3
11         60    100       120     250.7
12         60    106       128     345.3
13         60    104       132     379.3
14         60     98       123     275.0
15         60     98       120     215.2
16         60    100       120     300.0
17         45     90       112       NaN
18         60    103       123     323.0
19         45     97       125     243.0
20         60    108       131     364.2
21         45    100       119     282.0
22         60    130       101     300.0
23         45    105       132     246.0
24         60    102       126     334.5
25         60    100       120     250.0
26         60     92       118     241.0
27         60    103       132       NaN
28         60    100       132     280.0
29         60    102       129     380.3
30         60     92       115     243.0
31         45     90       112     180.1
32         60    101       124     299.0
33         60     93       113     223.0
34         60    107       136     361.0
35         60    114       140     415.0
36         60    102       127     300.0
37         60    100       120     300.0
38         60    100       120     300.0
39         45    104       129     266.0
40         45     90       112     180.1
41         60     98       126     286.0
42         60    100       122     329.4
43         60    111       138     400.0
44         60    111       131     397.0
45         60     99       119     273.0
46         60    109       153     387.6
47         45    111       136     300.0
48         45    108       129     298.0
49         60    111       139     397.6
50         60    107       136     380.2
51         80    123       146     643.1
52         60    106       130     263.0
53         60    118       151     486.0
54         30    136       175     238.0
55         60    121       146     450.7
56         60    118       121     413.0
57         45    115       144     305.0
58         20    153       172     226.4
```

59	45	123	152	321.0
60	210	108	160	1376.0
61	160	110	137	1034.4
62	160	109	135	853.0
63	45	118	141	341.0
64	20	110	130	131.4
65	180	90	130	800.4
66	150	105	135	873.4
67	150	107	130	816.0
68	20	106	136	110.4
69	300	108	143	1500.2
70	150	97	129	1115.0
71	60	109	153	387.6
72	90	100	127	700.0
73	150	97	127	953.2
74	45	114	146	304.0
75	90	98	125	563.2
76	45	105	134	251.0
77	45	110	141	300.0
78	120	100	130	500.4
79	270	100	131	1729.0
80	30	159	182	319.2
81	45	149	169	344.0
82	30	103	139	151.1
83	120	100	130	500.0
84	45	100	120	225.3
85	30	151	170	300.0
86	45	102	136	234.0
87	120	100	157	1000.1
88	45	129	103	242.0
89	20	83	107	50.3
90	180	101	127	600.1
91	45	107	137	NaN
92	30	90	107	105.3
93	15	80	100	50.5
94	20	150	171	127.4
95	20	151	168	229.4
96	30	95	128	128.2
97	25	152	168	244.2
98	30	109	131	188.2
99	90	93	124	604.1
100	20	95	112	77.7
101	90	90	110	500.0
102	90	90	100	500.0
103	90	90	100	500.4
104	30	92	108	92.7
105	30	93	128	124.0
106	180	90	120	800.3
107	30	90	120	86.2
108	90	90	120	500.3
109	210	137	184	1860.4
110	60	102	124	325.2
111	45	107	124	275.0
112	15	124	139	124.2
113	45	100	120	225.3
114	60	108	131	367.6
115	60	108	151	351.7
116	60	116	141	443.0
117	60	97	122	277.4
118	60	105	125	NaN
119	60	103	124	332.7
120	30	112	137	193.9
121	45	100	120	100.7
122	60	119	169	336.7
123	60	107	127	344.9
124	60	111	151	368.5
125	60	98	122	271.0
126	60	97	124	275.3
127	60	109	127	382.0
128	90	99	125	466.4
129	60	114	151	384.0
130	60	104	134	342.5
131	60	107	138	357.5
132	60	103	133	335.0
133	60	106	132	327.5
134	60	103	136	339.0
135	20	136	156	189.0
136	45	117	143	317.7
137	45	115	137	318.0
138	45	113	138	308.0
139	20	141	162	222.4
140	60	108	135	390.0
141	60	97	127	NaN

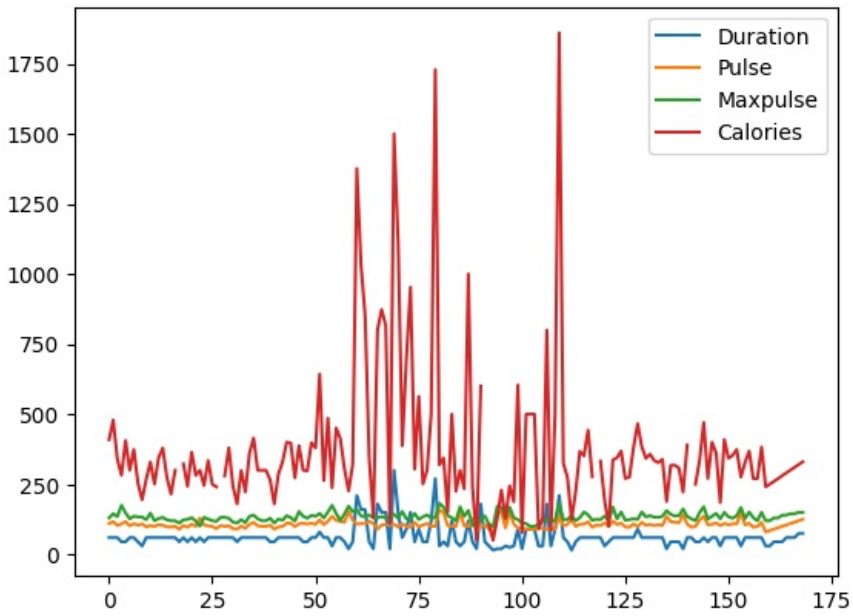
```
142        45    100       120     250.4
143        45    122       149     335.4
144        60    136       170     470.2
145        45    106       126     270.8
146        60    107       136     400.0
147        60    112       146     361.9
148        30    103       127     185.0
149        60    110       150     409.4
150        60    106       134     343.0
151        60    109       129     353.2
152        60    109       138     374.0
153        30    150       167     275.8
154        60    105       128     328.0
155        60    111       151     368.5
156        60     97       131     270.4
157        60    100       120     270.4
158        60    114       150     382.8
159        30     80       120     240.9
160        30     85       120     250.4
161        45     90       130     260.4
162        45     95       130     270.0
163        45    100       140     280.9
164        60    105       140     290.8
165        60    110       145     300.0
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4
```

In [12]:
```
import pandas as pd
# path1 = C:\Users\majag\Desktop\GitHub ⚒\Python ⚒
# path2 = C:\Users\majag\Desktop
data = pd.read_csv("C:/Users/majag/Desktop/data.csv")
view = pd.DataFrame(data)
print(view.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169 entries, 0 to 168
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Duration  169 non-null    int64  
 1   Pulse     169 non-null    int64  
 2   Maxpulse  169 non-null    int64  
 3   Calories  164 non-null    float64
dtypes: float64(1), int64(3)
memory usage: 5.4 KB
None
```

In [13]:
```
import pandas as pd
# path1 = C:\Users\majag\Desktop\GitHub ⚒\Python ⚒
# path2 = C:\Users\majag\Desktop
data = pd.read_csv("C:/Users/majag/Desktop/data.csv")
view = pd.DataFrame(data)
print(view.plot())
```

```
Axes(0.125,0.11;0.775x0.77)
```



math library

In [27]:
```python
import math
print(math.pi)
print(math.e)
print(math.sqrt(25))
print(math.sqrt(17))
print(math.isqrt(17))
print(math.fabs(17.9))
# Absolute value: Returning Positive Value.
print(math.ceil(31.9))
print(math.ceil(-31.9))
print(math.floor(31.9))
print(math.floor(-31.9))
```

```
3.141592653589793
2.718281828459045
5.0
4.123105625617661
4
17.9
32
-31
31
-32
```

- Scripting language: Everithing consideras a String. (Numbers also).
- By default in Input function everything consider as String. explicitly we have to convetit into numbers or any other value.
- Python is scripting language that's why it is use for web devlopment.