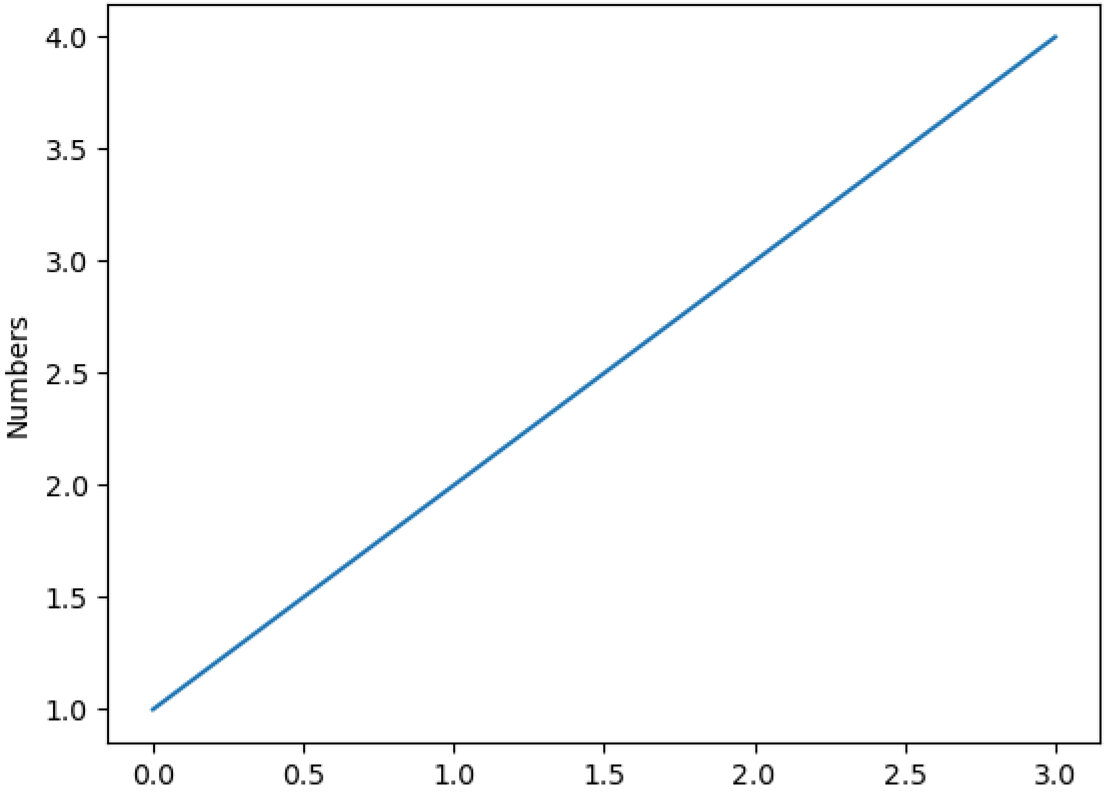
import matplotlib.pyplot as plt

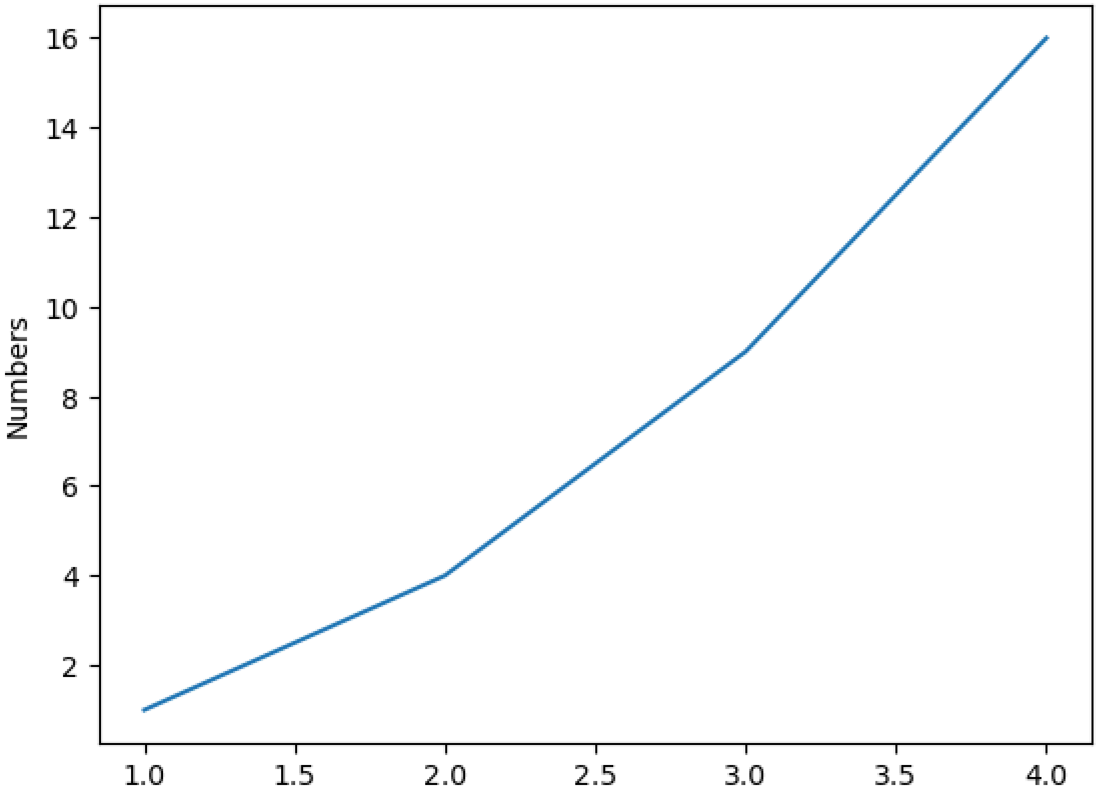
In [2]:

```
plt.plot([1,2,3,4])
plt.ylabel('Numbers')
plt.show()
```

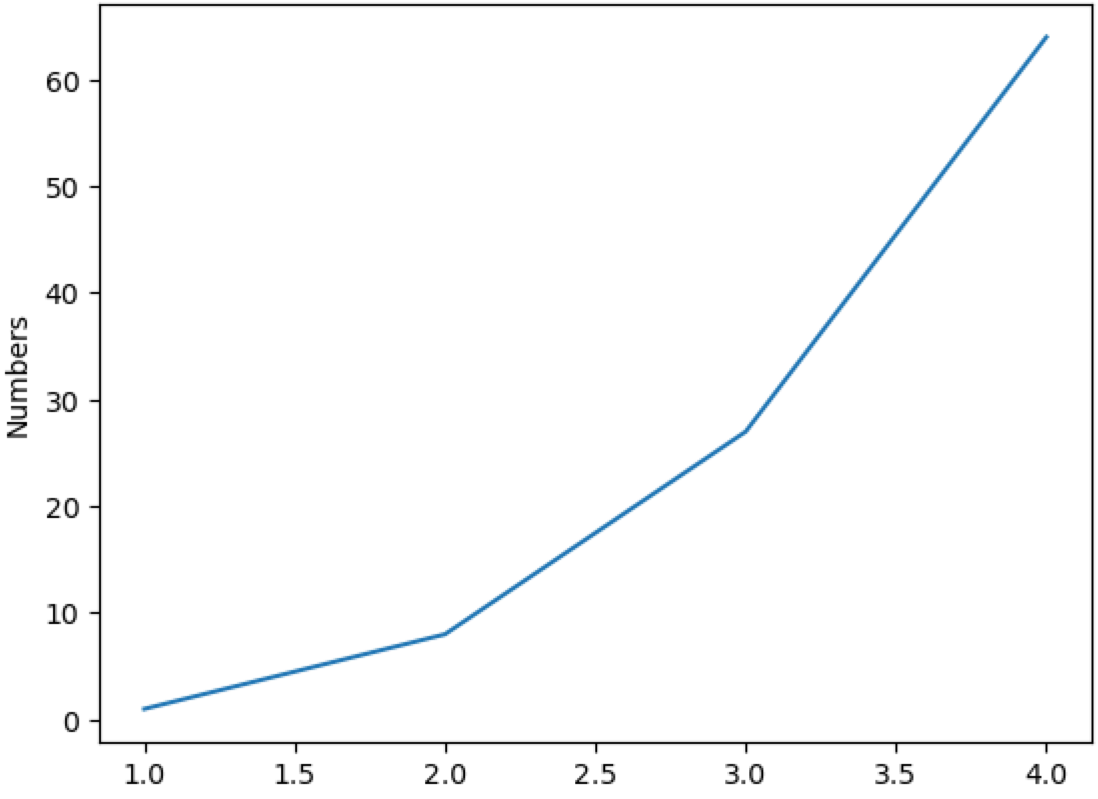


In [3]:

```
plt.plot([1,2,3,4], [1,4,9,16])
plt.ylabel('Numbers')
plt.show()
```



In [4]:
```
plt.plot([1,2,3,4],[1,8,27,64])
plt.ylabel('Numbers')
plt.show()
```



```
In [5]: def greet():
            print('hello')
        greet()

        hello
```

```
In [6]: a = 9
        a

Out[6]:  9
```

```
In [7]: a =9
        A
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[7], line 2
      1 a =9
----> 2 A

NameError: name 'A' is not defined
```

```
In [ ]: A = 9
        A
```

```
In [ ]: a_1 , a_2 , a_3 = 9, 5, 6
        a_3
```

```
In [ ]: a_2=9
```

```
In [ ]: a_2
```

```
In [ ]: del a_3
```

```
In [ ]: a_3
```

```
In [ ]: a_1
```

```
In [ ]: add 5 to a_2
```

```
In [ ]: add 5
```

a_1 + a_2

```
In [ ]: 2 + 3
```

```
In [ ]: a_1 + a_2
```

3 + 9

PYTHON IDENTIFIER = PYTHON VARIABLE

- identifier name> = value

In [ ]:

```
a = 36
a
```

In [ ]:

```
9a = 4
9a
```

In [ ]:

```
a1 = 3
a1
```

In [ ]:

```
a@ = 34
a@
```

In [ ]:

```
 a_1 = 3
a_1
```

In [ ]:

```
_a = 9
_a
```

In [ ]:

```
if = 56
if
```

In [1]:

```
import keyword
keyword.kwlist
```

```
Out[1]: ['False',
         'None',
         'True',
         'and',
         'as',
         'assert',
         'async',
         'await',
         'break',
         'class',
         'continue',
         'def',
         'del',
         'elif',
         'else',
         'except',
         'finally',
         'for',
         'from',
         'global',
         'if',
         'import',
         'in',
         'is',
         'lambda',
         'nonlocal',
         'not',
         'or',
         'pass',
         'raise',
         'return',
         'try',
         'while',
         'with',
         'yield']
```

In [ ]:

```
len(keyword.kwlist)
```

In [ ]:

```
a1 = True
a1
```

In [ ]:

```
a1 = true
a1
```

In [ ]:

```
For = 56
For
```

In [ ]:

```
for =56
for
```

In [ ]:

```
True + True
```

In [ ]:

```
True +False
```

```
In [ ]: False + False
```

```
In [ ]: int(True)
```

```
In [ ]: False / True
```

```
In [ ]: False // True
```

```
In [ ]: True / False
```

```
In [ ]: import numpy as np
```

```
In [ ]: np.zeros(3)
```

```
In [ ]: np.ones((10,10))
```

```
In [ ]: np.ones((4,3))
```

```
In [ ]: np.arange(2,10)
```

```
In [ ]: 2 +3
```

```
In [ ]: int._add_(2,3)
```

```
In [ ]: int.__add__(2,3)
```

```
In [ ]: int.__sub__(2,3)
```

```
In [ ]: int.__mul__(2,3)
```

```
In [ ]: int.__div__(2/3)
```

```
In [ ]: int.__div__(3/2)
```

```
In [ ]: int.__div__(4/2)
```

```
In [ ]: 2+3
```

```
In [ ]: 2+3
```

```
In [ ]: _ + 3
```

PYTHON DATA TYPES

```
In [ ]: A = 67
        A
```

```
In [ ]: type(A)
```

FLOAT DATA TYPE

f = 97.56 f

```
In [ ]: f = 97.56
        f
```

```
In [ ]: f = 97.56
        f
```

```
In [ ]: type(f)
```

```
In [39]: f1 = 1e0
         f1
```

```
Out[39]: 1.0
```

```
In [42]: type(f1)
```

```
Out[42]: float
```

```
In [43]: f2 = 1e1
         f2
```

```
Out[43]: 10.0
```

```
In [44]: f3 = e2
         f3
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[44], line 1
----> 1 f3 = e2
      2 f3

NameError: name 'e2' is not defined
```

```
In [ ]: f3 = 2e2
        f3
```

```
In [ ]: f4=3e4
        f4
```

```
In [ ]: f5 = 3E3
        f5
```

```
In [ ]: f6 = 4a4
        f6
```

In [ ]:

```
pi = 3.14
pi
```

In [ ]:

```
pi1 = 3.17
pi1
```

STRING

In [ ]:

```
s = 'Hello'
s
```

In [ ]:

```
s[5]
```

In [ ]:

```
s[4]
```

In [ ]:

```
print(s[0])
print(s[1])
print(s[2])
print(s[3])
print(s[4])
print(s[0])
```

In [ ]:

```
s1 = "hi"
s1
```

In [ ]:

```
s2 = '''welcome'''
s2
```

In [ ]:

```
s3 = '''hi
            welcome'''
s3
```

In [ ]:

```
nlp = '''data science and gen ai
        has great career in future'''
nlp
```

In [ ]:

```
a = hello
a
```

COMPLEX

In [ ]:

```
c = 10 + 20j
c
```

In [ ]:

```
type(c)
```

In [ ]:

```
c.real
```

```
In [ ]: c.imaginery
```

```
In [ ]: c.imag
```

```
In [ ]: d= 20 + 30j
        d
```

```
In [ ]: print(c)
        print(d)
```

```
In [ ]: e = c + d
        e
```

```
In [ ]: type(e)
```

bool

```
In [ ]: True
```

```
In [ ]: true
```

```
In [ ]: int(True)
```

```
In [ ]: int(False)
```

```
In [ ]: True + True
```

```
In [ ]: True + False + False
```

```
In [ ]: True + False + True * False
```

```
In [ ]: False / True
```

```
In [ ]: False // True
```

```
In [ ]: True / False
```

Completed datatype

```
In [ ]: int(12.3)
```

```
In [ ]: import numpy as np
```

```
In [ ]: a = np.nan
In [ ]: type(a)
```

```
In [ ]: i4, f4, c4, s4, b4 = 4, 4.4, 4+4j, 'hello', True
```

```
In [ ]: print(type(c4))
```

```
In [ ]: type(b4)
```

Type Casting

```
In [ ]: int(2.3)
```

```
In [ ]: int('10')
```

```
In [ ]: int('ten')
```

```
In [ ]: int(9+15j)
```

```
In [ ]: int(2.3,4.5)
```

```
In [ ]: float(2)
```

```
In [ ]: float('20')
```

```
In [ ]: float(True)
```

```
In [ ]: float(False)
```

```
In [ ]: float(10+20j)
```

```
In [ ]: complex(2)
```

```
In [ ]: complex(5,35)
```

```
In [ ]: complex(5.6)
```

```
In [ ]: complex(6.2,4.9,9.2)
```

```
In [ ]: complex(6.5,5.4)
```

```
In [ ]: complex(True)
```

```
In [ ]: complex(False, True)
```

```
In [ ]: complex(True,False)
In [ ]: complex("90")

In [ ]: complex('89','57')

In [ ]: complex(True)

In [ ]: complex(False)

In [ ]: complex('10',67)

In [ ]: complex(56,'23')

In [45]: complex(45,78)

Out[45]: (45+78j)

In [47]: str(5)

Out[47]: '5'

In [48]: str(4.5)

Out[48]: '4.5'

In [49]: str(False)

Out[49]: 'False'

In [51]: str(int(False))

Out[51]: '0'

In [53]: str(True)

Out[53]: 'True'

In [57]: str(int(True))

Out[57]: '1'

In [59]: str(45+78j)

Out[59]: '(45+78j)'

In [60]: bool(1)

Out[60]: True

In [65]: bool(6.5)
```

```
Out[65]: True
```

```
In [66]: bool()
```

```
Out[66]: False
```

```
In [69]: bool(0)
```

```
Out[69]: False
```

```
In [70]: bool( )
```

```
Out[70]: False
```

```
In [73]: bool(-1)
```

```
Out[73]: True
```

```
In [74]: bool(56+20j)
```

```
Out[74]: True
```

8th Aug

```
    Introduction to Data Structure
        List
```

```
In [83]: l = []
```

```
In [86]: type(l)
```

```
Out[86]: list
```

```
In [87]: l_ = list()
```

```
In [92]: type(l_)
```

```
Out[92]: list
```

```
In [93]: l.append(10)
         l
```

```
Out[93]: [10]
```

```
In [96]: l
```

```
Out[96]: [10]
```

```
In [98]: l.append(23)
         l.append(2.23)
```

```
l.append('hi')
l.append(2+4j)
l.append(True)
l.append(4)
```

```
In [103…  l
```

```
Out[103…  [10, 23, 2.23, 'hi', (2+4j), True, 4]
```

```
In [104…  l.append(10)
          l
```

```
Out[104…  [10, 23, 2.23, 'hi', (2+4j), True, 4, 10]
```

```
In [105…  l.count(10)
```

```
Out[105…  2
```

```
In [106…  l.count(23)
```

```
Out[106…  1
```

```
In [107…  l
```

```
Out[107…  [10, 23, 2.23, 'hi', (2+4j), True, 4, 10]
```

```
In [108…  len(l)
```

```
Out[108…  8
```

```
In [109…  l1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[109], line 1
----> 1 l1

NameError: name 'l1' is not defined
```

```
In [ ]:  l1 = l.copy()
```

```
In [ ]:  l1
```

```
In [ ]:  l
```

```
In [ ]:  l1
```

```
In [ ]:  l.remove(2+4j)
```

```
In [ ]:  l
```

```
In [ ]:  l1.remove(10)
In [ ]: l1
```

```
In [ ]: ;
```

```
In [ ]: l.pop()
```

```
In [ ]: l1.pop()
```

```
In [ ]: l1.pop()
```

```
In [ ]: l
```

```
In [ ]: l1
```

```
In [ ]: l.pop(1)
```

```
In [ ]: l
```

```
In [ ]: l[:]
```

```
In [ ]: l[1:]
```

```
In [ ]: l[:1]
```

```
In [ ]: l
```

```
In [ ]: l[:5]
```

```
In [ ]: l[1:5]
```

```
In [ ]: l
```

```
In [ ]: l[1:-1]
```

```
In [ ]: l
```

```
In [ ]: l1
```

```
In [ ]: l1[1]
```

```
In [ ]: l1[1] = 9
```

```
In [ ]: l1
```

```
In [ ]: l1[2] = 'hello'
```

```
In [ ]: l1
In [ ]: l1[0] = 3.14
        l1
```

```
In [ ]: l
```

```
In [ ]: l.index(10)
```

```
In [ ]: l.index(True)
```

```
In [ ]: l.extend(l1)
```

```
In [ ]: l
```

```
In [ ]: l1
```

```
In [ ]: l1.insert(1,False)
        l1
```

```
In [ ]: l1.clear(-4)
```

```
In [ ]: l1.clear()
        l1
```

```
In [ ]: l1
```

```
In [110…  l2 = [56,34,90,100]
          l2

Out[110…  [56, 34, 90, 100]
```

```
In [112…  l2.reverse()
          l2

Out[112…  [100, 90, 34, 56]
```

```
In [115…  l2.sort()
          l2

Out[115…  [34, 56, 90, 100]
```

```
In [116…  l2.sort(reverse =True )
          l2

Out[116…  [100, 90, 56, 34]
```

```
In [117…  l3 = [3,56,34,43,32]
          l3

Out[117…  [3, 56, 34, 43, 32]
```

In [120…

```
l3.reverse()
l3
```

Out[120…

```
[32, 43, 34, 56, 3]
```

In [121…

```
l2 = [56,34,90,100]
```

In [123…

```
l2.reverse()
l2
```

Out[123…

```
[100, 90, 34, 56]
```

mylist = [88,34,45,43,63,32] sorted(mylist)

In [125…

```
mylist
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[125], line 1
----> 1 mylist

NameError: name 'mylist' is not defined
```

all() and any()

```
all() - Returns false if any value is 0 or 'false'
        For output True of all(), all values should be true there
is no false or 0
any() - Returns True if any value is 1 or 'True'
```

In [ ]:

```
L = [10,34,0,34,45]
L
```

In [ ]:

```
all(L)
```

In [ ]:

```
L1 = [10,34,False,True,54]
all(L1)
```

In [ ]:

```
L2 = [10,34,'false',54]
all(L2)
```

In [ ]:

```
l = [10,34,1,45,56]
all(l)
```

In [ ]:

```
any(L)
```

In [ ]:

```
any(l)
```

List Membership

```
checks if the value or member exists or not
```

In [ ]:

```
10 in l
```

In [126…

```
False in L
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[126], line 1
----> 1 False in L

NameError: name 'L' is not defined
```

In [ ]:

```
True in L1
```

In [128…

```
100 in l
```

Out[128…

```
False
```

In [133…

```
if 100 in l:
    print('100 is exist')
else:
    print('100 is not exist')
```

```
100 is not exist
```

In [191…

```
l1 = ['a','m','o','z']
```

In [199…

```
l2 = [2,4.5,'nit',True,2+5j]
```

In [197…

```
l3 = [200,100,76, 9, 4,3]
```

In [201…

```
print(l1)
print(l2)
print(l3)
```

```
['a', 'm', 'o', 'z']
[2, 4.5, 'nit', True, (2+5j)]
[200, 100, 76, 9, 4, 3]
```

In [203…

```
for i in l2:
    print(i)
```

```
2
4.5
nit
True
(2+5j)
```

In [205…

```
for i in enumerate(l2):
    print(i)
```

```
(0, 2)
(1, 4.5)
(2, 'nit')
(3, True)
(4, (2+5j))
```

In [213…

```
l4 = l2 + l3
l4
```

Out[213…

```
[2, 4.5, 'nit', True, (2+5j), 200, 100, 76, 9, 4, 3]
```

In [223…

```
all(l1)
```

Out[223…

```
True
```

In [225…

```
l2.append(0)
```

In [227…

```
all(l2)
```

Out[227…

```
False
```

In [231…

```
any(l2)
```

Out[231…

```
True
```

list completed

Tuple

In [239…

```
t = ()
t
```

Out[239…

```
()
```

In [241…

```
type(t)
```

Out[241…

```
tuple
```

In [243…

```
t1 = (1, 2, 3)
t1
```

Out[243…

```
(1, 2, 3)
```

In [245…

```
t1.count(2)
```

Out[245…

```
1
```

```
In [247…  t2 = (10,20, 40 ,20)
          t2

Out[247…  (10, 20, 40, 20)

In [249…  t2.count(20)

Out[249…  2

In [251…  t2.index(10)

Out[251…  0

In [253…  t2.index(20)

Out[253…  1

In [255…  t1

Out[255…  (1, 2, 3)

In [257…  t1[0] = 100
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[257], line 1
----> 1 t1[0] = 100

TypeError: 'tuple' object does not support item assignment
```

```
In [259…  l1

Out[259…  ['a', 'm', 'o', 'z']

In [263…  l1[0] = 100
          l1

Out[263…  [100, 'm', 'o', 'z']

In [265…  t1.clear()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[265], line 1
----> 1 t1.clear()

AttributeError: 'tuple' object has no attribute 'clear'
```

```
In [271…  sbi = ('name', 2.34, 'cizp4532u', '4th nov 2001', 6798)
          sbi

Out[271…  ('name', 2.34, 'cizp4532u', '4th nov 2001', 6798)
```

```
In [269…  sbi.remove(234)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[269], line 1
----> 1 sbi.remove(234)

AttributeError: 'tuple' object has no attribute 'remove'
```

```
In [273…  t2 = sbi
```

```
In [275…  t2
```

```
Out[275…  ('name', 2.34, 'cizp4532u', '4th nov 2001', 6798)
```

```
In [277…  t3 = ([1,2,3], 45,'hello')
          t3
```

```
Out[277…  ([1, 2, 3], 45, 'hello')
```

```
In [283…  t4 = sbi.copy()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[283], line 1
----> 1 t4 = sbi.copy()

AttributeError: 'tuple' object has no attribute 'copy'
```

12th Aug

SET

```
In [3]:  s = {}
```

```
In [5]:  type(s)
```

```
Out[5]:  dict
```

```
In [9]:  s = set()
```

```
In [11]:  type(s)
```

```
Out[11]:  set
```

```
In [13]:  s.add(10)
```

```
In [15]:  s
```

```
Out[15]: {10}
```

```
In [17]: s.add(20,30)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[17], line 1
----> 1 s.add(20,30)

TypeError: set.add() takes exactly one argument (2 given)
```

```
In [21]: s.add(20)
         s.add(30)
         s.add(40)
         s.add(50)
```

```
In [23]: s
```

```
Out[23]: {10, 20, 30, 40, 50}
```

```
In [25]: len(s)
```

```
Out[25]: 5
```

```
In [29]: s[:]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[29], line 1
----> 1 s[:]

TypeError: 'set' object is not subscriptable
```

```
In [31]: s[3:]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[31], line 1
----> 1 s[3:]

TypeError: 'set' object is not subscriptable
```

```
In [33]: s.add(10)
```

```
In [35]: s
```

```
Out[35]: {10, 20, 30, 40, 50}
```

```
In [39]: s1 = set()
         s1
```

```
Out[39]: set()
```

```
In [41]: s1.add(2)
         s1.add(3.14)
         s1.add('hello')
         s1.add(1+2j)
         s1.add(True)
```

```
In [43]: s1
```

```
Out[43]: {(1+2j), 2, 3.14, True, 'hello'}
```

```
In [45]: s
```

```
Out[45]: {10, 20, 30, 40, 50}
```

```
In [47]: s2 = s.copy()
```

```
In [49]: s2
```

```
Out[49]: {10, 20, 30, 40, 50}
```

```
In [53]: s3 = set()
         s3
```

```
Out[53]: set()
```

```
In [55]: s3.add(100)
         s3.add(2)
         s3.add(15)
         s3.add(95)
         s3
```

```
Out[55]: {2, 15, 95, 100}
```

```
In [57]: len(s3)
```

```
Out[57]: 4
```

```
In [61]: s3.clear()
```

```
In [63]: len(s3)
```

```
Out[63]: 0
```

```
In [65]: s3
```

```
Out[65]: set()
```

```
In [67]: s2
```

```
Out[67]: {10, 20, 30, 40, 50}
```

In [69]:
```
s2.pop()
```

Out[69]:
```
50
```

In [71]:
```
s2
```

Out[71]:
```
{10, 20, 30, 40}
```

In [73]:
```
s1
```

Out[73]:
```
{(1+2j), 2, 3.14, True, 'hello'}
```

In [75]:
```
s1.pop()
```

Out[75]:
```
True
```

In [77]:
```
s1.remove((1+2j))
```

In [79]:
```
s1
```

Out[79]:
```
{2, 3.14, 'hello'}
```

In [81]:
```
s1.pop(2)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[81], line 1
----> 1 s1.pop(2)

TypeError: set.pop() takes no arguments (1 given)
```

In [83]:
```
s2
```

Out[83]:
```
{10, 20, 30, 40}
```

In [85]:
```
s2.remove(100)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[85], line 1
----> 1 s2.remove(100)

KeyError: 100
```

In [87]:
```
100 in s2
```

Out[87]:
```
False
```

In [89]:
```
10 in s2
```

Out[89]:
```
True
```

In [91]:

```
s2.discard(100)
```

In [93]:

```
s2
```

Out[93]:

```
{10, 20, 30, 40}
```

In [95]:

```
s2.discard(30)
s2
```

Out[95]:

```
{10, 20, 40}
```

In [101…

```
for i in s2:
    print(i)
```

```
20
40
10
```

Set operations

In [110…

```
A = {1,2,3,4,5}
B = {4,5,6,7,8}
C = {8,9,10}
```

In [112…

```
A.union(B)
```

Out[112…

```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [114…

```
B.union(C)
```

Out[114…

```
{4, 5, 6, 7, 8, 9, 10}
```

In [116…

```
A | B
```

Out[116…

```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [118…

```
A | B | C
```

Out[118…

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

In [122…

```
print(A)
print(B)
print(C)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

In [124…

```
A.intersection(B)
```

Out[124…

```
{4, 5}
```

In [126…

```
A.intersection(C)
```

Out[126…

```
set()
```

In [128…

```
B.intersection(C)
```

Out[128…

```
{8}
```

In [130…

```
B & C
```

Out[130…

```
{8}
```

In [132…

```
A & B
```

Out[132…

```
{4, 5}
```

In [134…

```
A.difference(B)
```

Out[134…

```
{1, 2, 3}
```

In [136…

```
B.difference(A)
```

Out[136…

```
{6, 7, 8}
```

In [138…

```
A.difference(C)
```

Out[138…

```
{1, 2, 3, 4, 5}
```

In [140…

```
C.difference(A)
```

Out[140…

```
{8, 9, 10}
```

In [142…

```
A - B
```

Out[142…

```
{1, 2, 3}
```

In [144…

```
B-C
```

Out[144…

```
{4, 5, 6, 7}
```

In [146…

```
C - B
```

Out[146…

```
{9, 10}
```

In [148…

```
print(A)
print(B)
print(C)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

In [150…

```
A.symmetric_difference(B)
```

Out[150…

```
{1, 2, 3, 6, 7, 8}
```

In [152…

```
A.symmetric_difference(C)
```

Out[152…

```
{1, 2, 3, 4, 5, 8, 9, 10}
```

In [154…

```
B.symmetric_difference(C)
```

Out[154…

```
{4, 5, 6, 7, 9, 10}
```

In [156…

```
B.symmetric_difference(A)
```

Out[156…

```
{1, 2, 3, 6, 7, 8}
```

13th AUG

In [2]:

```
A1 = {1,2,3,4,5,6,7,8,9}
B1 = {3,4,5,6,7,8}
C1 = {10,20,30,40}
```

In [4]:

```
A1.issubset(B1)
```

Out[4]:

```
False
```

In [6]:

```
B1.issubset(A1)
```

Out[6]:

```
True
```

In [8]:

```
C1.isdisjoint(B1)
```

Out[8]:

```
True
```

In [10]:

```
A1.issuperset(B1)
```

Out[10]:

```
True
```

In [12]:

```
A1.issuperset(C1)
```

Out[12]:

```
False
```

In [14]:

```
list(enumerate(A1))
```

Out[14]:

```
[(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9)]
```

In [16]:

```
for i in enumerate(A1):
    print(i)
```

```
(0, 1)
(1, 2)
(2, 3)
(3, 4)
(4, 5)
(5, 6)
(6, 7)
(7, 8)
(8, 9)
```

SET datastructure completed

Dict

```
In [20]: d = dict()
         type(d)

Out[20]: dict

In [22]: d

Out[22]: {}

In [24]: d1 = {1: 'one', 2: 'two', 3: 'three', 4:'four', 5:'five'}
         d1

Out[24]: {1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}

In [26]: d1.items()

Out[26]: dict_items([(1, 'one'), (2, 'two'), (3, 'three'), (4, 'four'), (5, 'five')])

In [28]: len(d1)

Out[28]: 5

In [30]: d1.keys()

Out[30]: dict_keys([1, 2, 3, 4, 5])

In [32]: d1.values()

Out[32]: dict_values(['one', 'two', 'three', 'four', 'five'])

In [34]: d1['six'] = 6
         d1

Out[34]: {1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five', 'six': 6}
```

In [36]:

```
d1[6] = 'six'
d1
```

Out[36]:

```
{1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five', 'six': 6, 6: 'six'}
```

In [38]:

```
d1[6]
```

Out[38]:

```
'six'
```

In [40]:

```
d1['two']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[40], line 1
----> 1 d1['two']

KeyError: 'two'
```

In [42]:

```
d1.get(1)
```

Out[42]:

```
'one'
```

In [46]:

```
d1[:]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[46], line 1
----> 1 d1[:]

KeyError: slice(None, None, None)
```

In [48]:

```
d1[1:]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[48], line 1
----> 1 d1[1:]

KeyError: slice(1, None, None)
```

In [50]:

```
d1
```

Out[50]:

```
{1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five', 'six': 6, 6: 'six'}
```

In [52]:

```
d1.pop()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[52], line 1
----> 1 d1.pop()

TypeError: pop expected at least 1 argument, got 0
```

In [54]:

```
d1.pop('six')
```

```
Out[54]: 6

In [56]: d1

Out[56]: {1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five', 6: 'six'}

In [58]: d1.popitem()

Out[58]: (6, 'six')

In [60]: d1

Out[60]: {1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}

In [64]: d2 = d1.copy()
         d2

Out[64]: {1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}

In [66]: id(d1)

Out[66]: 794670938368

In [68]: id(d2)

Out[68]: 794757454080
```

Numpy begins

```
In [1]: import numpy as np

In [3]: np.__version__

Out[3]: '1.26.4'

In [4]: my_list = [0,1,2,3,4,5]
        my_list

Out[4]: [0, 1, 2, 3, 4, 5]

In [5]: type(my_list)

Out[5]: list

In [6]: arr = np.array(my_list)

In [7]: arr

Out[7]: array([0, 1, 2, 3, 4, 5])
```

```
In [8]:  type(arr)

Out[8]:  numpy.ndarray

In [9]:  np.arange(15)

Out[9]:  array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14])

In [10]: np.arange(3.0)

Out[10]: array([0., 1., 2.])

In [11]: np.arange(10)

Out[11]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

In [12]: np.arange(1,5)

Out[12]: array([1, 2, 3, 4])

In [13]: np.arange(10,20)

Out[13]: array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])

In [14]: np.arange(20,10)

Out[14]: array([], dtype=int32)

In [15]: np.arange(-20,10)

Out[15]: array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11, -10,  -9,  -8,
                 -7,  -6,  -5,  -4,  -3,  -2,  -1,   0,   1,   2,   3,   4,   5,
                  6,   7,   8,   9])

In [16]: np.arange(-20,-10)

Out[16]: array([-20, -19, -18, -17, -16, -15, -14, -13, -12, -11])

In [17]: np.arange(10,10)

Out[17]: array([], dtype=int32)

In [18]: np.arange(10,30,5)

Out[18]: array([10, 15, 20, 25])

In [19]: np.zeros(5) #parameter tuning

Out[19]: array([0., 0., 0., 0., 0.])

In [20]: np.zeros(5, dtype = int) #hyper parameter tuning
```

```
Out[20]: array([0, 0, 0, 0, 0])
```

```
In [21]: np.zeros((10,10,2) , dtype = int)
```

```
Out[21]: array([[[0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0]],

                [[0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0]],

                [[0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0]],

                [[0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0]],

                [[0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0],
                 [0, 0]],

                [[0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0]],

       [[0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0]],

       [[0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0]],

       [[0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0]],

       [[0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0],
        [0, 0]]])
```

```
In [22]: np.zeros((2,2,1) , dtype = int)
```

```
Out[22]:  array([[[0],
                  [0]],

                 [[0],
                  [0]]])
```

In [23]:

```
zero = np.zeros([2,2])
type(zeros)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[23], line 2
      1 zero = np.zeros([2,2])
----> 2 type(zeros)

NameError: name 'zeros' is not defined
```

In [26]:

```
range(5)
```

```
Out[26]:  range(0, 5)
```

In [28]:

```
r = range(5)
r
```

```
Out[28]:  range(0, 5)
```

In [30]:

```
for i in r:
    print(i)
```

```
0
1
2
3
4
```

In [32]:

```
list(range(5))
```

```
Out[32]:  [0, 1, 2, 3, 4]
```

In [34]:

```
range(1,10)
```

```
Out[34]:  range(1, 10)
```

In [36]:

```
list(range(1,10))
```

```
Out[36]:  [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [38]:

```
list(range(1,10,3))
```

```
Out[38]:  [1, 4, 7]
```

In [40]:

```
y = list(range(12))
y
```

```
Out[40]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

```
In [42]: rand(3,2)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[42], line 1
----> 1 rand(3,2)

NameError: name 'rand' is not defined
```

```
In [44]: np.random.rand(4)
```

```
Out[44]: array([0.91454118, 0.0969258 , 0.36889894, 0.35271603])
```

```
In [46]: np.rand(4)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[46], line 1
----> 1 np.rand(4)

File ~\anaconda3\Lib\site-packages\numpy\__init__.py:333, in __getattr__(attr)
    330     "Removed in NumPy 1.25.0"
    331     raise RuntimeError("Tester was removed in NumPy 1.25.")
--> 333 raise AttributeError("module {!r} has no attribute "
    334                      "{!r}".format(__name__, attr))

AttributeError: module 'numpy' has no attribute 'rand'
```

```
In [48]: np.random.rand(50)
```

```
Out[48]: array([4.17858380e-01, 8.33447602e-01, 1.60025316e-01, 9.25976541e-01,
                3.43232305e-01, 8.43868693e-01, 8.86932462e-01, 9.53041001e-01,
                3.26557975e-02, 9.99376659e-01, 8.34708858e-01, 4.54589216e-01,
                2.68051486e-01, 7.98454390e-01, 1.84279368e-01, 8.86563280e-01,
                2.92293791e-01, 5.10960767e-01, 8.13925792e-01, 1.26918795e-01,
                7.62654518e-01, 2.86069627e-01, 4.76795347e-01, 6.50585901e-02,
                6.94751669e-01, 6.14052953e-01, 3.77358353e-01, 5.41519972e-01,
                8.97313274e-01, 1.36641460e-01, 8.99759718e-01, 9.79839309e-01,
                5.06034566e-01, 1.92260541e-01, 2.47213557e-03, 9.84359578e-01,
                9.32788471e-01, 3.53732610e-01, 5.94319435e-02, 6.03332433e-01,
                5.36188065e-01, 5.26796144e-01, 6.29379995e-01, 8.59319798e-02,
                9.83827201e-04, 4.23452141e-01, 8.11962010e-01, 9.46189925e-01,
                2.40986898e-01, 9.42743749e-02])
```

```
In [50]: np.random.rand(2,3)
```

```
Out[50]: array([[0.53610461, 0.98758278, 0.02532292],
                [0.68012169, 0.61057635, 0.6819755 ]])
```

```
In [52]: np.random.randint(2,5)
```

```
Out[52]: 4
```

```
In [54]: np.random.randint(0,1)

Out[54]: 0

In [56]: np.random.randint(1,2,3)

Out[56]: array([1, 1, 1])

In [58]: np.random.randint(1,6,4)

Out[58]: array([5, 4, 1, 1])

In [60]: np.random.randint(3,6,4)

Out[60]: array([3, 4, 5, 5])

In [62]: np.random.randint(3,4,5)

Out[62]: array([3, 3, 3, 3, 3])

In [66]: np.random.randint(10,40,(10,10))

Out[66]: array([[24, 21, 27, 20, 21, 11, 10, 17, 25, 23],
                [37, 39, 32, 26, 26, 12, 38, 36, 28, 37],
                [27, 10, 34, 31, 36, 21, 13, 39, 30, 10],
                [35, 33, 13, 38, 17, 26, 29, 37, 25, 38],
                [37, 26, 32, 20, 36, 12, 36, 26, 12, 31],
                [16, 16, 27, 21, 38, 14, 22, 21, 38, 35],
                [18, 19, 37, 19, 15, 26, 25, 11, 37, 27],
                [25, 28, 10, 39, 19, 35, 16, 25, 11, 37],
                [31, 12, 18, 37, 21, 31, 30, 25, 24, 11],
                [33, 25, 33, 19, 25, 29, 14, 25, 12, 17]])

In [68]: np.random.randint(10,20,(4,5))

Out[68]: array([[17, 16, 12, 15, 18],
                [12, 13, 14, 18, 11],
                [18, 19, 16, 13, 10],
                [13, 17, 18, 15, 15]])

In [70]: b = np.random.randint(10,15,(5,4))

In [72]: b

Out[72]: array([[11, 14, 14, 13],
                [13, 13, 14, 13],
                [10, 14, 13, 12],
                [14, 12, 10, 12],
                [12, 11, 11, 12]])

In [74]: b[:]
```

```
Out[74]:  array([[11, 14, 14, 13],
                 [13, 13, 14, 13],
                 [10, 14, 13, 12],
                 [14, 12, 10, 12],
                 [12, 11, 11, 12]])
```

```
In [76]:  b[1:3]
```

```
Out[76]:  array([[13, 13, 14, 13],
                 [10, 14, 13, 12]])
```

```
In [78]:  b[1,3]
```

```
Out[78]:  13
```

```
In [80]:  b[2:3]
```

```
Out[80]:  array([[10, 14, 13, 12]])
```

```
In [82]:  b
```

```
Out[82]:  array([[11, 14, 14, 13],
                 [13, 13, 14, 13],
                 [10, 14, 13, 12],
                 [14, 12, 10, 12],
                 [12, 11, 11, 12]])
```

```
In [84]:  b[0:-1]
```

```
Out[84]:  array([[11, 14, 14, 13],
                 [13, 13, 14, 13],
                 [10, 14, 13, 12],
                 [14, 12, 10, 12]])
```

```
In [86]:  b[0,2]
```

```
Out[86]:  14
```

```
In [88]:  b[-5,-3]
```

```
Out[88]:  14
```

```
In [90]:  np.random.randint(10,20,(4,4))
```

```
Out[90]:  array([[13, 14, 18, 17],
                 [16, 15, 18, 19],
                 [16, 12, 11, 12],
                 [11, 16, 12, 13]])
```

```
In [92]:  b
```

Out[92]:

```
array([[11, 14, 14, 13],
       [13, 13, 14, 13],
       [10, 14, 13, 12],
       [14, 12, 10, 12],
       [12, 11, 11, 12]])
```

In [94]:

```
b[4,2]
```

Out[94]:

```
11
```

In [96]:

```
b[-4:2]
```

Out[96]:

```
array([[13, 13, 14, 13]])
```

In [98]:

```
b[-4:]
```

Out[98]:

```
array([[13, 13, 14, 13],
       [10, 14, 13, 12],
       [14, 12, 10, 12],
       [12, 11, 11, 12]])
```

In [100…

```
a = np.random.randint(10,20,5)
```

In [102…

```
a
```

Out[102…

```
array([10, 18, 18, 19, 19])
```

In [104…

```
arr
```

Out[104…

```
array([0, 1, 2, 3, 4, 5])
```

In [110…

```
arr1 = np.random.randint(0,100,(10,10))
```

In [112…

```
arr1
```

Out[112…

```
array([[16, 90,  2, 13, 78,  3, 72, 67, 90, 22],
       [10, 15, 27, 42, 21, 75, 47, 83, 90, 60],
       [30, 18, 23, 48,  9, 34, 65, 34, 59, 49],
       [67, 20, 39, 68, 10, 18, 59, 28, 47,  0],
       [80, 23, 98, 84, 87, 91, 29, 77, 32, 93],
       [81, 18, 53, 77, 98, 89, 78, 60, 60, 63],
       [62,  2, 49, 99,  9, 14, 75, 89, 13, 25],
       [31, 53, 44, 33, 94, 18, 60, 12, 31, 64],
       [ 5, 91, 61,  5, 28, 54, 97, 57,  1, 51],
       [69, 40, 47, 30, 56, 87, 32, 37, 61,  6]])
```

In [114…

```
arr[:4]
```

Out[114…

```
array([0, 1, 2, 3])
```

In [116…

```
arr2[0:5]
```

```
Out[116…  array([[57, 71,  6, 15,  5, 53, 57, 36, 86, 79],
                 [67, 13, 50, 38, 45, 51, 73, 29, 52, 44],
                 [48, 91, 27, 35, 16, 83, 34, 53, 20, 32],
                 [14, 79, 15, 80, 91, 51, 30, 53, 46, 49],
                 [64, 33, 81,  5,  6, 10, 84, 90, 93, 90]])
```

```
In [118…  arr2[1,5]
```

```
Out[118…  51
```

```
In [120…  arr2[::-1]
```

```
Out[120…  array([[70, 96, 58, 57,  1, 56, 49, 80, 45, 13],
                 [66, 10, 87, 46, 18, 25, 85, 88,  8,  7],
                 [36, 97, 13, 50, 74, 92,  2, 23, 67,  2],
                 [ 7, 71, 54, 53, 27, 89, 12, 92, 75,  9],
                 [76, 41, 58, 68, 51, 21, 14,  9, 98, 19],
                 [64, 33, 81,  5,  6, 10, 84, 90, 93, 90],
                 [14, 79, 15, 80, 91, 51, 30, 53, 46, 49],
                 [48, 91, 27, 35, 16, 83, 34, 53, 20, 32],
                 [67, 13, 50, 38, 45, 51, 73, 29, 52, 44],
                 [57, 71,  6, 15,  5, 53, 57, 36, 86, 79]])
```

```
In [124…  arr2
```

```
Out[124…  array([[57, 71,  6, 15,  5, 53, 57, 36, 86, 79],
                 [67, 13, 50, 38, 45, 51, 73, 29, 52, 44],
                 [48, 91, 27, 35, 16, 83, 34, 53, 20, 32],
                 [14, 79, 15, 80, 91, 51, 30, 53, 46, 49],
                 [64, 33, 81,  5,  6, 10, 84, 90, 93, 90],
                 [76, 41, 58, 68, 51, 21, 14,  9, 98, 19],
                 [ 7, 71, 54, 53, 27, 89, 12, 92, 75,  9],
                 [36, 97, 13, 50, 74, 92,  2, 23, 67,  2],
                 [66, 10, 87, 46, 18, 25, 85, 88,  8,  7],
                 [70, 96, 58, 57,  1, 56, 49, 80, 45, 13]])
```

```
In [126…  arr2[::-2]
```

```
Out[126…  array([[70, 96, 58, 57,  1, 56, 49, 80, 45, 13],
                 [36, 97, 13, 50, 74, 92,  2, 23, 67,  2],
                 [76, 41, 58, 68, 51, 21, 14,  9, 98, 19],
                 [14, 79, 15, 80, 91, 51, 30, 53, 46, 49],
                 [67, 13, 50, 38, 45, 51, 73, 29, 52, 44]])
```

```
In [128…  arr2
```

```
Out[128…  array([[57, 71,  6, 15,  5, 53, 57, 36, 86, 79],
                 [67, 13, 50, 38, 45, 51, 73, 29, 52, 44],
                 [48, 91, 27, 35, 16, 83, 34, 53, 20, 32],
                 [14, 79, 15, 80, 91, 51, 30, 53, 46, 49],
                 [64, 33, 81,  5,  6, 10, 84, 90, 93, 90],
                 [76, 41, 58, 68, 51, 21, 14,  9, 98, 19],
                 [ 7, 71, 54, 53, 27, 89, 12, 92, 75,  9],
                 [36, 97, 13, 50, 74, 92,  2, 23, 67,  2],
                 [66, 10, 87, 46, 18, 25, 85, 88,  8,  7],
                 [70, 96, 58, 57,  1, 56, 49, 80, 45, 13]])
```

```
In [130…  arr2[::-3]
```

```
Out[130…  array([[70, 96, 58, 57,  1, 56, 49, 80, 45, 13],
                 [ 7, 71, 54, 53, 27, 89, 12, 92, 75,  9],
                 [14, 79, 15, 80, 91, 51, 30, 53, 46, 49],
                 [57, 71,  6, 15,  5, 53, 57, 36, 86, 79]])
```

```
In [132…  arr2[:-3]
```

```
Out[132…  array([[57, 71,  6, 15,  5, 53, 57, 36, 86, 79],
                 [67, 13, 50, 38, 45, 51, 73, 29, 52, 44],
                 [48, 91, 27, 35, 16, 83, 34, 53, 20, 32],
                 [14, 79, 15, 80, 91, 51, 30, 53, 46, 49],
                 [64, 33, 81,  5,  6, 10, 84, 90, 93, 90],
                 [76, 41, 58, 68, 51, 21, 14,  9, 98, 19],
                 [ 7, 71, 54, 53, 27, 89, 12, 92, 75,  9]])
```

```
In [134…  arr
```

```
Out[134…  array([0, 1, 2, 3, 4, 5])
```

```
In [136…  arr.max()
```

```
Out[136…  5
```

```
In [138…  arr.min()
```

```
Out[138…  0
```

```
In [140…  arr.mean()
```

```
Out[140…  2.5
```

```
In [142…  arr.median()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[142], line 1
----> 1 arr.median()

AttributeError: 'numpy.ndarray' object has no attribute 'median'
```

```
In [144…  from numpy import *
          a = array([1,2,3,4,9])
```

```
median(a)
```

Out[144…

```
3.0
```

without import

In [146…

```
arr
```

Out[146…

```
array([0, 1, 2, 3, 4, 5])
```

In [ ]:

```
reshape
```

Indexing

In [150…

```
mat = np.arange(0,100).reshape(10,10)
```

In [152…

```
mat
```

Out[152…

```
array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14, 15, 16, 17, 18, 19],
       [20, 21, 22, 23, 24, 25, 26, 27, 28, 29],
       [30, 31, 32, 33, 34, 35, 36, 37, 38, 39],
       [40, 41, 42, 43, 44, 45, 46, 47, 48, 49],
       [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
       [60, 61, 62, 63, 64, 65, 66, 67, 68, 69],
       [70, 71, 72, 73, 74, 75, 76, 77, 78, 79],
       [80, 81, 82, 83, 84, 85, 86, 87, 88, 89],
       [90, 91, 92, 93, 94, 95, 96, 97, 98, 99]])
```

In [154…

```
mat[row,col]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[154], line 1
----> 1 mat[row,col]

NameError: name 'row' is not defined
```

In [156…

```
col = 6
```

In [158…

```
mat
```

```
Out[158…  array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9],
                 [10, 11, 12, 13, 14, 15, 16, 17, 18, 19],
                 [20, 21, 22, 23, 24, 25, 26, 27, 28, 29],
                 [30, 31, 32, 33, 34, 35, 36, 37, 38, 39],
                 [40, 41, 42, 43, 44, 45, 46, 47, 48, 49],
                 [50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
                 [60, 61, 62, 63, 64, 65, 66, 67, 68, 69],
                 [70, 71, 72, 73, 74, 75, 76, 77, 78, 79],
                 [80, 81, 82, 83, 84, 85, 86, 87, 88, 89],
                 [90, 91, 92, 93, 94, 95, 96, 97, 98, 99]])
```

```
In [160…  mat[6]
```

```
Out[160…  array([60, 61, 62, 63, 64, 65, 66, 67, 68, 69])
```

```
In [ ]:  row = 4
```

```
In [1]:  x = 9
```

```
In [3]:  _ + 2
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[3], line 1
----> 1 _ + 2

TypeError: can only concatenate str (not "int") to str
```

```
In [5]:  _ - 6
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[5], line 1
----> 1 _ - 6

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

```
In [1]:  help()
```

```
Welcome to Python 3.12's help utility! If this is your first time using
Python, you should definitely check out the tutorial at
https://docs.python.org/3.12/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules.  To get a list of available
modules, keywords, symbols, or topics, enter "modules", "keywords",
"symbols", or "topics".

Each module also comes with a one-line summary of what it does; to list
the modules whose name or summary contain a given string such as "spam",
enter "modules spam".

To quit this help utility and return to the interpreter,
enter "q" or "quit".
```

```
You are now leaving help and returning to the Python interpreter.
If you want to ask for help on a particular object directly from the
interpreter, you can type "help(object)".  Executing "help('string')"
has the same effect as typing a particular string at the help> prompt.
```

In [3]:

```
a = sqrt(25)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[3], line 1
----> 1 a = sqrt(25)

NameError: name 'sqrt' is not defined
```

In [5]:

```
import math
```

In [9]:

```
a = math.sqrt(25)
a
```

Out[9]:

```
5.0
```

In [11]:

```
a1 = math.sqrt(15)
a1
```

Out[11]:

```
3.872983346207417
```

In [13]:

```
print(math.floor(3.87))
```

```
3
```

In [17]:

```
print(math.ceil(3.87))
```

```
4
```

In [23]:

```
print(math.pi)
```

```
3.141592653589793
```

In [25]:

```
x = input()
```

In [27]:

```
a = input()
b = input()
c = a + b
c
```

Out[27]:

```
'54'
```

In [29]:

```
type(c)
```

Out[29]:

```
str
```

In [35]:

```
a1 = input("Enter first number")
b1 = input("enter second no")
x1 = int(a1)
y1 = int(b1)
```

```
c = x1 + y1
c
```

Out[35]:
```
15
```

In [37]:
```
a1 = int(input("enter no"))
b1 = int(input("enter no"))
c = a1 + b1
c
```

Out[37]:
```
17
```

In [ ]: