

# python 1st code

lets start the code

```
In [173...]: #addition  
          10+5
```

```
Out[173...]: 15
```

```
In [175...]: #difference  
           10-5
```

```
Out[175...]: 5
```

```
In [177...]: #division  
           10/5
```

```
Out[177...]: 2.0
```

```
In [178...]: #multiplication  
           10*5
```

```
Out[178...]: 50
```

```
In [180...]: #modulo  
           10//5
```

```
Out[180...]: 2
```

```
In [183...]: #if we randomly do any math operation then receive last solution only.  
           1+1  
           2+1  
           3+1
```

```
Out[183...]: 4
```

```
In [185...]: print(1+1)  
           print(2+1)  
           print(3+1)
```

```
2  
3  
4
```

```
In [187...]: #it is the best option to write the code.  
           a=10  
           b=5  
           c=a+b  
           print(c);
```

```
15
```

```
In [189...]: # if the solution getting it is ok ,getting error then system version issue  
           - + 3
```

```
TypeError                                     Traceback (most recent call last)
Cell In[189], line 2
      1 # if the soluton getting it is ok ,getting error then system version issue
      2 _ + 3
      3
TypeError: can only concatenate str (not "int") to str
```

function() : function contain bracket.() variable : variable not contain bracket. family(dad,mom):this function having we pass 2 arguments or parameter. family1(dad,mom,bro,sis) :this function having we pass 4 arguments. family2() :this having no argument ,this is empty, without arguments. variable=value :it is right. value=variable :it is false having error. shift+enter then run the code. \_ means it stored previous output. statistic == number == numerical data == text == catagorial data python variable = python identifier = python object

```
In [191... (10+5)-7+5
```

```
Out[191... 13
```

```
In [193... 5+(5*5)
```

```
Out[193... 30
```

```
In [195... (5+5)*5
```

```
Out[195... 50
```

```
In [197... import sys
         sys.version
```

```
Out[197... '3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:03:56) [MSC v.192
9 64 bit (AMD64)]'
```

```
In [199... print(10)
         print(10,20)
         print('python')
         print(10,30,'python')
```

```
10
10 20
python
10 30 python
```

```
In [201... num1=20
         num2=30
         add=num1+num2
         print('Add two number =', add);
```

```
Add two number = 50
```

## python variable concept=python identifier concept

. syntax of define variable || (variable name=value) || identifier=value

```
In [204... NIT=15
         NIT
```

```
Out[204...]: 15
```

```
In [206...]: NIT=20  
NIT
```

```
Out[206...]: 20
```

```
In [208...]: V=15  
V
```

```
Out[208...]: 15
```

```
In [210...]: print(V)  
print(NIT)
```

```
15  
20
```

```
In [212...]: Nit
```

```
NameError  
Cell In[212], line 1  
----> 1 Nit  
  
NameError: name 'Nit' is not defined
```

```
In [214...]: 1var=10  
1var
```

```
Cell In[214], line 1  
 1var=10  
  ^  
SyntaxError: invalid decimal literal
```

```
In [216...]: var1=10  
var1
```

```
Out[216...]: 10
```

```
In [218...]: var$=10
```

```
Cell In[218], line 1  
  var$=10  
  ^  
SyntaxError: invalid syntax
```

```
In [220...]: var_=10  
var_
```

```
Out[220...]: 10
```

```
In [222...]: a,b=1,10  
print(a)  
print(b)
```

```
1  
10
```

```
In [224...]: a,b=5,13,13  
          print(a)  
          print(b)
```

```
-----  
ValueError                                     Traceback (most recent call last)  
Cell In[224], line 1  
----> 1 a,b=5,13,13  
      2 print(a)  
      3 print(b)  
  
ValueError: too many values to unpack (expected 2)
```

```
In [226...]: aaaaaaaaaaaaaaaaaaa=15  
          print(aaaaaaaaaaaaaaaaaaa)
```

```
15  
python variable complete
```

## python datatype

```
In [230...]: i=5  
          i
```

```
Out[230...]: 5
```

```
In [232...]: type(i)
```

```
Out[232...]: int
```

```
In [234...]: print(type(i))
```

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

```
In [236...]: i=30  
          i
```

```
Out[236...]: 30
```

```
In [238...]: i1,i2=20,30  
          print(i)  
          print(i1)  
          print(i2)
```

```
30  
20  
30
```

```
In [240...]: i-i1+i2
```

```
Out[240...]: 40
```

```
In [242...]: i+(i2-i1)  
          print(i+(i2-i1))
```

```
40
```

```
In [244... f=110.35  
f
```

```
Out[244... 110.35
```

```
In [246... type(f)
```

```
Out[246... float
```

```
In [248... f1,f2,f3 = 2.0,3.4,5.4  
print(f)  
print(f1)  
print(f2)  
print(f3)
```

```
110.35  
2.0  
3.4  
5.4
```

```
In [250... 1f=5.4
```

```
Cell In[250], line 1  
1f=5.4  
^  
SyntaxError: invalid decimal literal
```

```
In [252... f1=1e0  
f1
```

```
Out[252... 1.0
```

```
In [254... f3 =3e2  
f3
```

```
Out[254... 300.0
```

```
In [256... f4=4b2  
f4
```

```
Cell In[256], line 1  
f4=4b2  
^  
SyntaxError: invalid decimal literal
```

```
In [258... b=true  
b
```

```
NameError  
Cell In[258], line 1  
----> 1 b=true  
      2 b  
  
NameError: name 'true' is not defined
```

```
In [260... b=True  
b
```

```
Out[260...]: True
```

```
In [262...]: b1=False  
b1
```

```
Out[262...]: False
```

```
In [264...]: True +False
```

```
Out[264...]: 1
```

```
In [266...]: False/True
```

```
Out[266...]: 0.0
```

```
In [268...]: True/False
```

```
ZeroDivisionError  
Cell In[268], line 1  
----> 1 True/False
```

```
Traceback (most recent call last)
```

```
ZeroDivisionError: division by zero
```

complex data type

```
In [271...]: c=(1+20j)  
c
```

```
Out[271...]: (1+20j)
```

```
In [273...]: type(c)
```

```
Out[273...]: complex
```

```
In [275...]: import keyword  
keyword.kwlist
```

```
Out[275... ['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 [277... len(keyword.kwlist)
```

```
Out[277... 35
```

```
In [279... if=45
if
```

```
Cell In[279], line 1
if=45
^
SyntaxError: invalid syntax
```

```
In [281... p,q,r=20,20,20
p
q
r
```

```
Out[281... 20
```

```
In [283... p=20
p=p+10
p
```

```
Out[283... 30
```

```
In [285... c=5+2j  
c
```

```
Out[285... (5+2j)
```

```
In [287... c.imag
```

```
Out[287... 2.0
```

```
In [289... c.real
```

```
Out[289... 5.0
```

```
In [291... c1=10+20j  
c2=30+40j  
c1+c2
```

```
Out[291... (40+60j)
```

```
In [293... print(c1+c2)  
print(c1-c2)
```

```
(40+60j)  
(-20-20j)
```

## 24th oct

```
In [296... s='nit'  
s
```

```
Out[296... 'nit'
```

```
In [298... type(s)
```

```
Out[298... str
```

```
In [300... s1="hello python"  
s1
```

```
Out[300... 'hello python'
```

```
In [302... s2=""" nit  
            hello python """  
s2
```

```
Out[302... ' nit\n            hello python '
```

```
In [304... s1
```

```
Out[304... 'hello python'
```

```
In [306... s1[0]
```

```
Out[306... 'h'
```

```
In [308... s1[-4]
```

```
Out[308... 't'
```

```
In [310... s1[4]
```

```
Out[310... 'o'
```

```
In [312... s1[5]
```

```
Out[312... '
```

```
In [314... s1[-7]
```

```
Out[314... '
```

```
In [316... s
```

```
Out[316... 'nit'
```

```
In [318... print(s[0])
         print(s[1])
         print(s[2])
```

```
n
i
t
```

```
In [320... s1
```

```
Out[320... 'hello python'
```

```
In [322... s1[:]
```

```
Out[322... 'hello python'
```

```
In [324... s1[2:7]
```

```
Out[324... 'llo p'
```

```
In [326... s3='dataanalyst'
         s3
```

```
Out[326... 'dataanalyst'
```

```
In [328... s3[0:10]
```

```
Out[328... 'dataanalys'
```

```
In [330... s3=[0:12]
```

```
Cell In[330], line 1
    s3=[0:12]
    ^
SyntaxError: invalid syntax
```

```
In [332... s3[10]
```

```
Out[332... 't'
```

```
In [334... s3[::-2]
```

```
Out[334... 'tyaatd'
```

```
In [336... s3[0:11:2]
```

```
Out[336... 'dtaayt'
```

```
In [338... s3[0:11:3]
```

```
Out[338... 'daas'
```

```
In [340... s3
```

```
Out[340... 'dataanalyst'
```

```
In [342... s3[2:-2]
```

```
Out[342... 'taanaly'
```

```
In [344... print(s)
         print(s1)
         print(s3)
```

```
nit
hello python
dataanalyst
```

```
In [346... s3
```

```
Out[346... 'dataanalyst'
```

```
In [348... for i in s3:
            print(i)
```

```
d
a
t
a
a
n
a
l
y
s
t
```

## Python type casting or type conversion

```
In [351... int(2.3)
```

```
Out[351... 2
```

```
In [353... int(True) #bool to int
```

```
Out[353... 1
```

```
In [355... int(1+2j)
```

```
-----  
TypeError  
Cell In[355], line 1  
----> 1 int(1+2j)
```

```
Traceback (most recent call last)
```

```
TypeError: int() argument must be a string, a bytes-like object or a real number,  
not 'complex'
```

```
In [357... int('10')
```

```
Out[357... 10
```

```
In [359... int('ten')
```

```
-----  
ValueError  
Cell In[359], line 1  
----> 1 int('ten')
```

```
Traceback (most recent call last)
```

```
ValueError: invalid literal for int() with base 10: 'ten'
```

```
In [361... s2
```

```
Out[361... ' nit\n      hello python'
```

```
In [363... del s2 #delete the variable.
```

```
In [365... s2
```

```
-----  
NameError  
Cell In[365], line 1  
----> 1 s2
```

```
Traceback (most recent call last)
```

```
NameError: name 's2' is not defined
```

```
In [367... np.nan
```

```
-----  
NameError  
Cell In[367], line 1  
----> 1 np.nan
```

```
Traceback (most recent call last)
```

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

```
In [369... import numpy as np  
a= np.nan
```

```
In [371... type(a)
```

```
Out[371... float
```

## 25th oct

```
In [374... float(3)
```

```
Out[374... 3.0
```

```
In [376... float(True)
```

```
Out[376... 1.0
```

```
In [378... float(5+2j)
```

```
-----  
TypeError Traceback (most recent call last)  
Cell In[378], line 1  
----> 1 float(5+2j)  
  
TypeError: float() argument must be a string or a real number, not 'complex'
```

```
In [380... float(3,2)
```

```
-----  
TypeError Traceback (most recent call last)  
Cell In[380], line 1  
----> 1 float(3,2)  
  
TypeError: float expected at most 1 argument, got 2
```

```
In [382... float('10')
```

```
Out[382... 10.0
```

```
In [384... float('ten')
```

```
-----  
ValueError Traceback (most recent call last)  
Cell In[384], line 1  
----> 1 float('ten')  
  
ValueError: could not convert string to float: 'ten'
```

```
In [386... complex(10)
```

```
Out[386... (10+0j)
```

```
In [388... complex(10,20)
```

```
Out[388... (10+20j)
```

```
In [390... complex(10,20,30)
```

```
-----  
TypeError                                         Traceback (most recent call last)  
Cell In[390], line 1  
----> 1 complex(10,20,30)  
  
TypeError: complex() takes at most 2 arguments (3 given)  
  
In [392... complex(20.3)  
  
Out[392... (20.3+0j)  
  
In [394... complex(20.3,10)  
  
Out[394... (20.3+10j)  
  
In [396... complex(True)  
  
Out[396... (1+0j)  
  
In [398... complex(False)  
  
Out[398... 0j  
  
In [400... complex('10')  
  
Out[400... (10+0j)  
  
In [402... bool(5)  
  
Out[402... True  
  
In [404... bool(20.5)  
  
Out[404... True  
  
In [406... bool(0)  
  
Out[406... False  
  
In [408... bool(1)  
  
Out[408... True  
  
In [410... bool('10')  
  
Out[410... True  
  
In [412... bool('ten')  
  
Out[412... True  
  
In [414... bool()  
  
Out[414... False  
  
In [416... bool( )
```

```
Out[416... False
```

```
In [418... bool(10+52j)
```

```
Out[418... True
```

```
In [420... bool(0+0j)
```

```
Out[420... False
```

```
In [422... print(str(2))
         print(str(2.3))
         print(str(True))
         print(str(1+2j))
```

```
2
2.3
True
(1+2j)
```

## completed python type casting

```
In [425... index='HELOPYTHON'
index
```

```
Out[425... 'HELOPYTHON'
```

```
In [427... index[::-1]
index
```

```
Out[427... 'HELOPYTHON'
```

```
In [429... index[::-1] # reverse string formula
```

```
Out[429... 'NOHTYPOLLEH'
```

```
In [431... index[::-2] # print
```

```
Out[431... 'NHYOLH'
```

```
In [433... index
```

```
Out[433... 'HELOPYTHON'
```

```
In [435... index[::-4]
```

```
Out[435... 'NYL'
```

```
In [437... index[:-4]
```

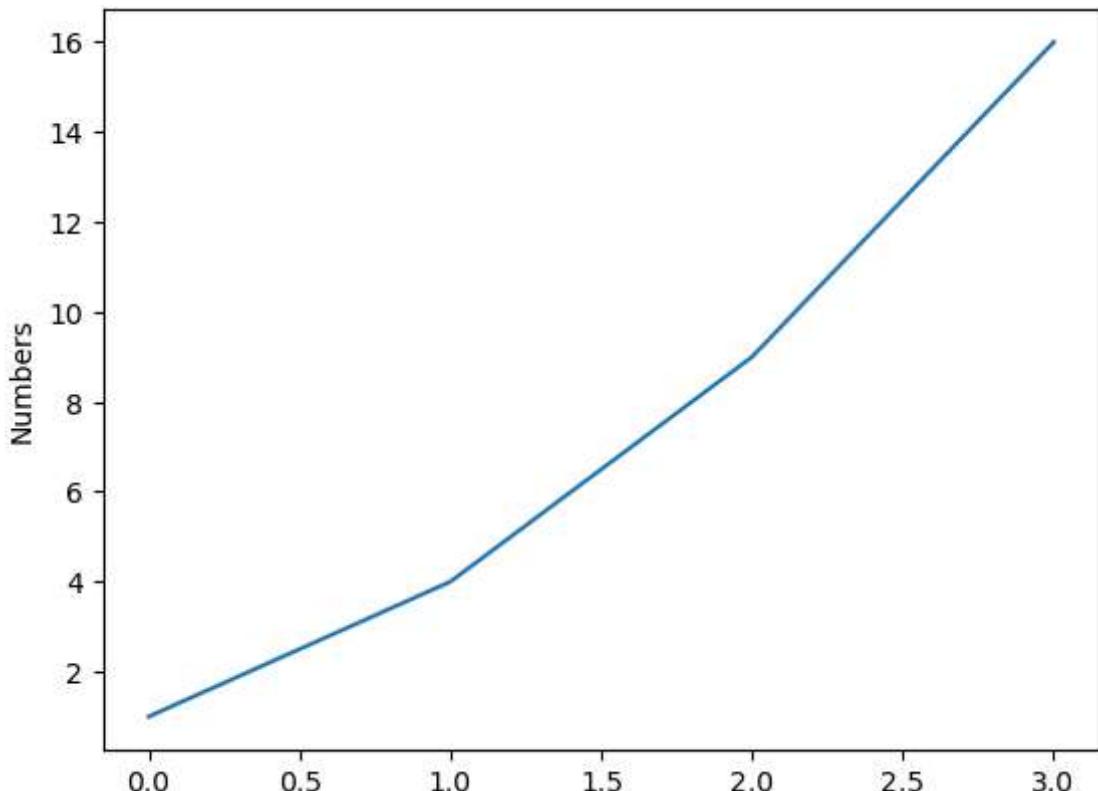
```
Out[437... 'HELOPY'
```

```
In [439... index[1:10:3]
```

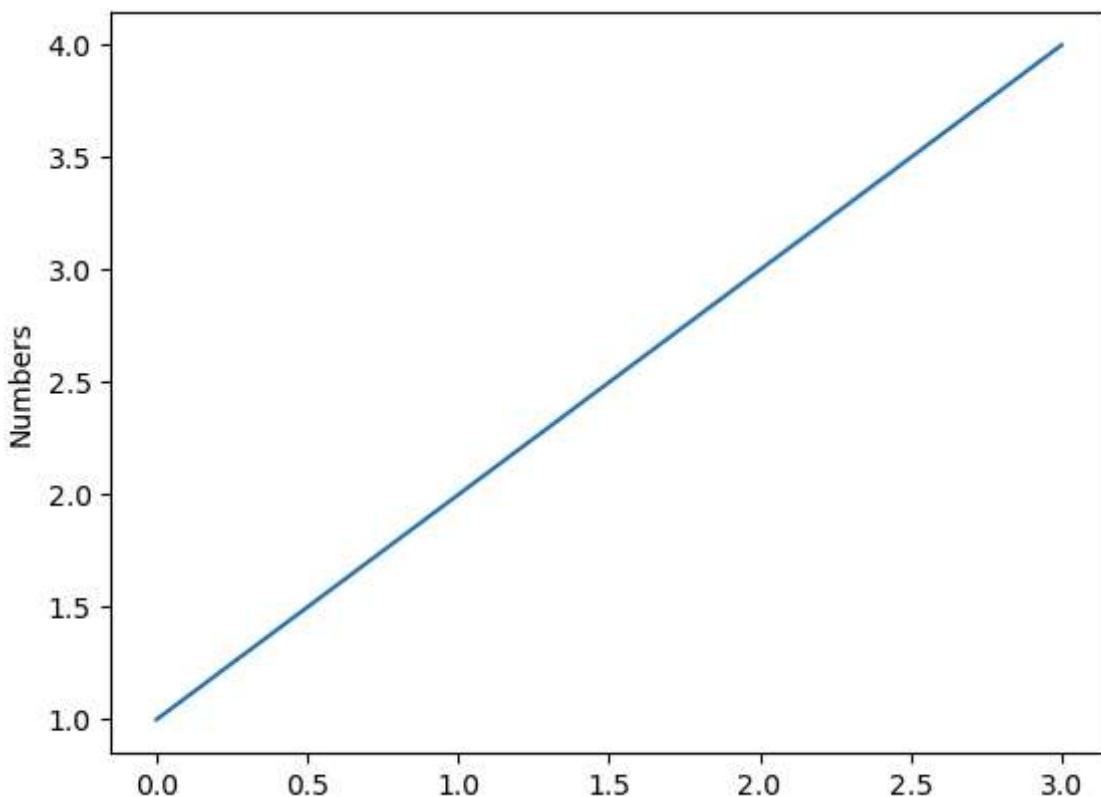
```
Out[439... 'EOT'
```

## python type casting completed

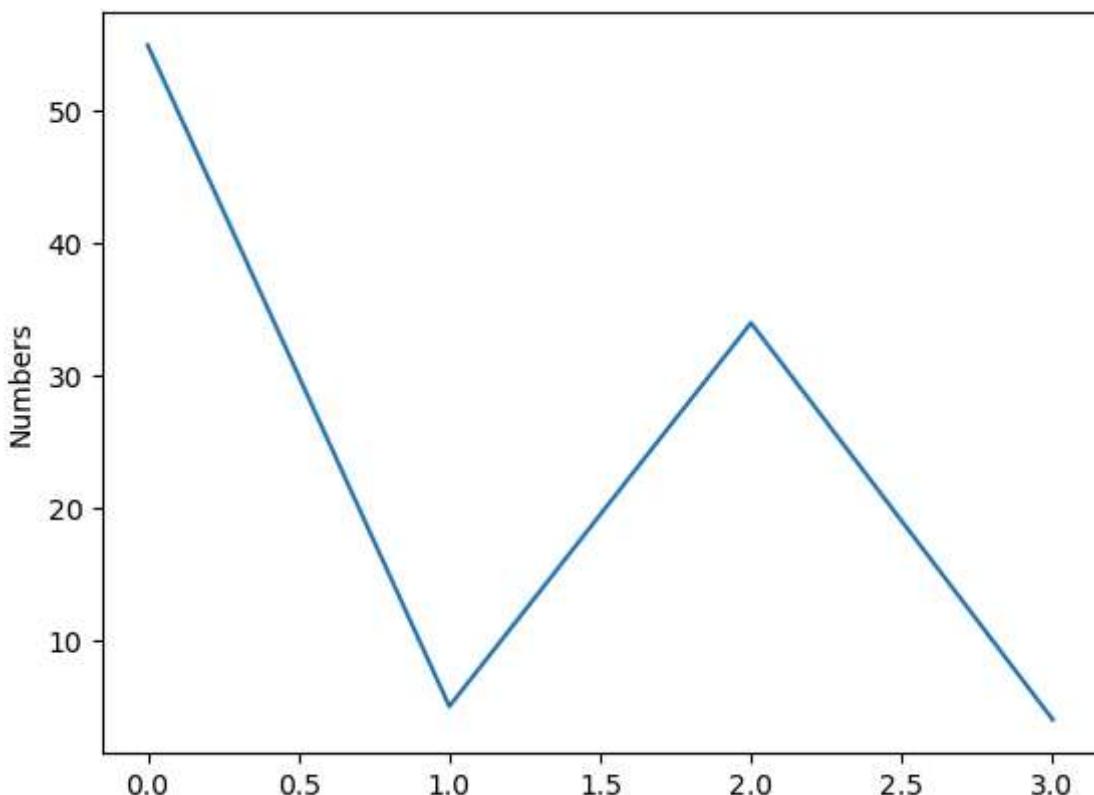
```
In [442...  
import matplotlib.pyplot as plt  
plt.plot([1,4,9,16])  
plt.ylabel('Numbers')  
plt.show()
```



```
In [443...  
import matplotlib.pyplot as plt  
plt.plot([1,2,3,4])  
plt.ylabel('Numbers')  
plt.show()
```



```
In [444...]: import matplotlib.pyplot as plt  
plt.plot([55,5,34,4])  
plt.ylabel('Numbers')  
plt.show()
```



```
In [445...]: import matplotlib.pyplot as plt  
plt.plot([1,2,3,4],[1,4,9,16],go)  
plt.ylabel('Numbers')  
plt.show()
```

```
NameError Traceback (most recent call last)
Cell In[445], line 2
      1 import matplotlib.pyplot as plt
----> 2 plt.plot([1,2,3,4],[1,4,9,16],go)
      3 plt.ylabel('Numbers')
      4 plt.show()

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

## 26th - DATA STRUCTURE

```
In [465... 1 =[]
1
```

```
Out[465... []
```

```
In [467... type(1)
```

```
Out[467... list
```

```
In [469... 1
```

```
Out[469... []
```

```
In [471... 1.append(11)
1.append(22)
1.append(33)
1
```

```
Out[471... [11, 22, 33]
```

```
In [473... 1.remove(20)
1
```

```
ValueError Traceback (most recent call last)
Cell In[473], line 1
----> 1 1.remove(20)
      2 1

ValueError: list.remove(x): x not in list
```

```
In [475... s=[]
s
```

```
Out[475... []
```

```
In [477... type(s)
```

```
Out[477... list
```

```
In [479... s.append(10)
s.append(20)
s.append(30)
s
```

```
Out[479... [10, 20, 30]
```

```
In [481... s.append(2.3)
s
```

```
Out[481... [10, 20, 30, 2.3]
```

```
In [483... s.append(1+2j)
s
```

```
Out[483... [10, 20, 30, 2.3, (1+2j)]
```

```
In [485... s.append('nit')
s
```

```
Out[485... [10, 20, 30, 2.3, (1+2j), 'nit']
```

```
In [487... s.append(True)
s
```

```
Out[487... [10, 20, 30, 2.3, (1+2j), 'nit', True]
```

```
In [489... len(s)
```

```
Out[489... 7
```

```
In [491... s.append(10)
s
```

```
Out[491... [10, 20, 30, 2.3, (1+2j), 'nit', True, 10]
```

```
In [493... s.remove(10)
s.remove(2.3)
s
```

```
Out[493... [20, 30, (1+2j), 'nit', True, 10]
```

```
In [495... s.append(10)
s
```

```
Out[495... [20, 30, (1+2j), 'nit', True, 10, 10]
```

```
In [497... s.remove(10)
s
```

```
Out[497... [20, 30, (1+2j), 'nit', True, 10]
```

```
In [499... l1=s.copy()
l1
```

```
Out[499... [20, 30, (1+2j), 'nit', True, 10]
```

```
In [501... s==l1
```

```
Out[501... True
```

```
In [503...]: s
Out[503...]: [20, 30, (1+2j), 'nit', True, 10]

In [505...]: s.count(10)
Out[505...]: 1

In [507...]: s.append(10)
Out[509...]: s
Out[509...]: [20, 30, (1+2j), 'nit', True, 10, 10]

In [511...]: s==11
Out[511...]: False

In [513...]: l1
Out[513...]: [20, 30, (1+2j), 'nit', True, 10]

In [515...]: print(l1)
              print(s)
Out[515...]: [20, 30, (1+2j), 'nit', True, 10]
              [20, 30, (1+2j), 'nit', True, 10, 10]

In [517...]: s[4]
Out[517...]: True
```

## 28th OCT

```
In [520...]: l
Out[520...]: [11, 22, 33]

In [521...]: l2=[]
              l2
Out[521...]: []

In [524...]: l2.append(2.3)
              l2.append(2)
              l2.append(True)
              l2.append(1+2j)
              l2.append('nit')
              l2
Out[524...]: [2.3, 2, True, (1+2j), 'nit']

In [526...]: l2.append(2.3)
```

```
In [528...]
```

```
12
```

```
Out[528...]
```

```
[2.3, 2, True, (1+2j), 'nit', 2.3]
```

```
In [530...]
```

```
len(12)
```

```
Out[530...]
```

```
6
```

```
In [532...]
```

```
13.clear()
```

```
NameError
```

```
Cell In[532], line 1
----> 1 13.clear()
```

```
Traceback (most recent call last)
```

```
NameError: name '13' is not defined
```

```
In [534...]
```

```
len(12)
```

```
Out[534...]
```

```
6
```

```
In [536...]
```

```
del(13)
```

```
NameError
```

```
Cell In[536], line 1
----> 1 del(13)
```

```
Traceback (most recent call last)
```

```
NameError: name '13' is not defined
```

```
In [538...]
```

```
13
```

```
NameError
```

```
Cell In[538], line 1
----> 1 13
```

```
Traceback (most recent call last)
```

```
NameError: name '13' is not defined
```

```
In [540...]
```

```
12
```

```
Out[540...]
```

```
[2.3, 2, True, (1+2j), 'nit', 2.3]
```

```
In [542...]
```

```
12.remove(2.3)
```

```
In [544...]
```

```
12
```

```
Out[544...]
```

```
[2, True, (1+2j), 'nit', 2.3]
```

```
In [546...]
```

```
13=[]
13.append(10)
13
```

```
Out[546...]
```

```
[10]
```

```
In [548...]
```

```
13.extend(12)
```

```
In [550...]
```

```
13
```

```
Out[550...]
```

```
[10, 2, True, (1+2j), 'nit', 2.3]
```

```
In [552...]
```

```
13.index(1+2j)
```

```
Out[552...]
```

```
3
```

```
In [554...]
```

```
13.index(True)
```

```
Out[554...]
```

```
2
```

```
In [556...]
```

```
12
```

```
Out[556...]
```

```
[2, True, (1+2j), 'nit', 2.3]
```

```
In [558...]
```

```
12.index('nit')
```

```
Out[558...]
```

```
3
```

```
In [560...]
```

```
13.insert(5,'technology')
```

```
In [562...]
```

```
13
```

```
Out[562...]
```

```
[10, 2, True, (1+2j), 'nit', 'technology', 2.3]
```

```
In [564...]
```

```
13.insert(3,False)
```

```
13
```

```
Out[564...]
```

```
[10, 2, True, False, (1+2j), 'nit', 'technology', 2.3]
```

```
In [566...]
```

```
13.pop()
```

```
Out[566...]
```

```
2.3
```

```
In [568...]
```

```
13
```

```
Out[568...]
```

```
[10, 2, True, False, (1+2j), 'nit', 'technology']
```

```
In [570...]
```

```
13.pop(4)
```

```
Out[570...]
```

```
(1+2j)
```

```
In [572...]
```

```
13
```

```
Out[572...]
```

```
[10, 2, True, False, 'nit', 'technology']
```

```
In [574...]
```

```
13.pop(3)
```

```
Out[574...]
```

```
False
```

```
In [576...]
```

```
13
```

```
Out[576...]
```

```
[10, 2, True, 'nit', 'technology']
```

```
In [578...]: 14=[10,100,3,45,76,24]
14
```

```
Out[578...]: [10, 100, 3, 45, 76, 24]
```

```
In [580...]: 14.sort()
14
```

```
Out[580...]: [3, 10, 24, 45, 76, 100]
```

```
In [582...]: 14.sort(reverse=True)
14
```

```
Out[582...]: [100, 76, 45, 24, 10, 3]
```

```
In [584...]: 15=['z','m','c','w']
15
```

```
Out[584...]: ['z', 'm', 'c', 'w']
```

```
In [586...]: 15.sort()
15
```

```
Out[586...]: ['c', 'm', 'w', 'z']
```

```
In [588...]: 16=[1,2,3,'a','g','z']
16
```

```
Out[588...]: [1, 2, 3, 'a', 'g', 'z']
```

```
In [590...]: 16.sort()
16
```

```
-----  
TypeError  
Cell In[590], line 1  
----> 1 16.sort()  
      2 16
```

```
Traceback (most recent call last)
```

```
TypeError: '<' not supported between instances of 'str' and 'int'
```

```
In [592...]: 13
```

```
Out[592...]: [10, 2, True, 'nit', 'technology']
```

```
In [594...]: 13.reverse()
```

```
In [596...]: 13
```

```
Out[596...]: ['technology', 'nit', True, 2, 10]
```

```
In [598...]: 12
```

```
Out[598...]: [2, True, (1+2j), 'nit', 2.3]
```

```
In [600...]: 12[3]
```

```
Out[600... 'nit'
```

```
In [602... print(l2[3][0])
print(l2[3][1])
print(l2[3][2])
```

```
n
i
t
```

```
In [604... 13
```

```
Out[604... ['technology', 'nit', True, 2, 10]
```

```
In [606... for i in l3:
    print(i)
```

```
technology
nit
True
2
10
```

## 29th oct

```
In [609... 13
```

```
Out[609... ['technology', 'nit', True, 2, 10]
```

```
In [611... l3[1]='mit'
```

```
In [613... 13
```

```
Out[613... ['technology', 'mit', True, 2, 10]
```

```
In [615... 16=['sbi','kotka']
17=['hdfc','boi']
```

```
In [617... 18=16+17
18
```

```
Out[617... ['sbi', 'kotka', 'hdfc', 'boi']
```

```
In [619... 13
```

```
Out[619... ['technology', 'mit', True, 2, 10]
```

```
In [621... for i in enumerate (l3):
    print(i)
```

```
(0, 'technology')
(1, 'mit')
(2, True)
(3, 2)
(4, 10)
```

```
In [623...]: 13
Out[623...]: ['technology', 'mit', True, 2, 10]

In [625...]: 13[::]
Out[625...]: ['technology', 'mit', True, 2, 10]

In [627...]: 13[::-1]
Out[627...]: [10, 2, True, 'mit', 'technology']

In [629...]: 13[::-2]
Out[629...]: [10, True, 'technology']

In [631...]: 13[17]
-----
IndexError                                     Traceback (most recent call last)
Cell In[631], line 1
      1 13[17]
IndexError: list index out of range

In [633...]: 13
Out[633...]: ['technology', 'mit', True, 2, 10]

In [635...]: 13[2:]
Out[635...]: [True, 2, 10]

In [637...]: 13[:4]
Out[637...]: ['technology', 'mit', True, 2]

In [639...]: 13[:9]
Out[639...]: ['technology', 'mit', True, 2, 10]

In [641...]: 13
Out[641...]: ['technology', 'mit', True, 2, 10]

In [643...]: 13[0:5:3]
Out[643...]: ['technology', 2]

In [645...]: 13[0:5:2]
Out[645...]: ['technology', True, 10]
```

**list is completed**

# tuple

```
In [649...]: t=()
t
```

```
Out[649...]: ()
```

```
In [651...]: type(t)
```

```
Out[651...]: tuple
```

```
In [653...]: t1=tuple()
t1
```

```
Out[653...]: ()
```

```
In [655...]: print(type(t1))
<class 'tuple'>
```

```
In [657...]: t=(10,10,20,30)
t
```

```
Out[657...]: (10, 10, 20, 30)
```

```
In [ ]: t[0]
```

```
In [ ]: t[0]=100 #immutable
```

```
In [ ]: t1=(10,2.6,1+2j,'nit',True)
t1
```

```
In [ ]: t1.count(10)
```

```
In [ ]: t1.index('nit')
```

```
In [ ]: for i in t1:
    print(i)
```

```
In [ ]: for i in enumerate (t1):
    print(i)
```

```
In [ ]: t
```

```
In [ ]: t4= t * 4
```

```
In [ ]: t4
```

```
In [ ]: t
```

```
In [ ]: t[:]
```

```
In [ ]: l=(10,20,30,40)
         l1=(90,45,78,89)
         l3=l+l1
         l3
```

tuple is completed,

## 30th oct SET

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

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

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

```
In [ ]: s1={52,46,79,2,5}
         s1
```

```
In [ ]: s2={2.3,4.5,8.7,1.9}
         s2
```

```
In [ ]: type(s2)
```

```
In [ ]: s3={'z','f','s','a'}
         s3
```

```
In [ ]: s4={10,2.3,'a',5,6.7}
         s4
```

```
In [ ]: print(s1)
         print(s2)
         print(s3)
         print(s4)
```

```
In [ ]: for i in s1:
         print(i)
```

```
In [ ]: for i in enumerate (s4):
         print(i)
```

```
In [ ]: s4.add(10)
         s4.add(2.3)
         s4.add(20)
         s4
```

```
In [ ]: s1
```

```
In [ ]: s1.add(4)  
s1
```

```
In [ ]: s4
```

```
In [ ]: len(s4)
```

```
In [ ]: s4.clear()  
s4
```

```
In [ ]: len(s4)
```

```
In [ ]: del.s4  
s4
```

```
In [ ]: s4=s1.copy()
```

```
In [ ]: s4
```

```
In [ ]: s1
```

```
In [ ]: s1==s4
```

```
In [ ]: s1
```

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

```
In [ ]: s2
```

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

```
In [ ]: s4
```

```
In [ ]: s4.pop()
```

```
In [ ]: A={1,2,5,8,9}  
B={4,5,1,2,6,}  
C={7,9,2,6,1,3,}
```

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

```
In [ ]: A.union(B)
```

```
In [ ]: d_union=A.union(B)  
d_union
```

```
In [ ]: print(A)  
print(B)  
print(C)  
print(d_union)
```

```
In [ ]: B.union(A,C)
```

```
In [ ]: All_union=B.union(A,C)
All_union
```

```
In [ ]: A | B | C
```

```
In [ ]: print(A)
print(B)
print(C)
print(d_union)
```

```
In [ ]: A.update(B)
A
```

```
In [ ]: print(A)
print(B)
print(C)
print(d_union)
```

```
In [ ]: C.update(5)
```

```
In [ ]: a={1,2,5,8,9}
b={4,5,1,2,6,}
c={7,9,2,6,1,3,}
```

```
In [ ]: a.intersection(b)
```

```
In [ ]: b & c
```

```
In [ ]: a1={1,2,5,8,9}
b1={4,5,1,2,6,}
c1={7,9,2,6,1,3,}
```

```
In [ ]: a1 - b1
```

```
In [ ]: a1-c1
```

```
In [ ]: b1-c1
```

```
In [ ]: a1={1,2,5,8,9}
b1={4,5,1,2,6,}
c1={7,9,2,6,1,3,}
```

```
In [ ]: b1.difference(c1)
```

```
In [ ]: b1.symmetric_difference(a1)
```

```
In [ ]: a1.difference(b1,c1)
```

```
In [ ]: a1-b1-c1
```

```
In [ ]: a5={}
```

# 1st nov 2024

```
In [ ]: a5={1,2,3,4,5,6,7,8,9}  
b5={3,4,5,6,7,8}  
c5={10,20,30,40}
```

```
In [ ]: a5.issuperset(b5)
```

```
In [ ]: a5.issubset(b5)
```

```
In [ ]: b5.issubset(a5)
```

```
In [ ]: a5.isdisjoint(b5)
```

```
In [ ]: a5
```

```
In [ ]: sum(a5)
```

```
In [ ]: max(a5)
```

```
In [ ]: min(a5)
```

```
In [ ]: len(a5)
```

```
In [ ]: list(enumerate(a5))
```

## set is completed. Dict start

```
In [ ]: d={}  
d
```

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

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

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

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

```
In [ ]: print(len(d1))
```

```
In [ ]: d{2}
```

```
In [ ]: d[:]
```

```
In [ ]: d
```

```
In [ ]: d1
```

```
In [ ]: d.keys()
```

```
In [ ]: d.values()
```

```
In [ ]: d1.keys()
```

```
In [ ]: d1.values()
```

```
In [ ]: for i in d:  
      print(i)
```

```
In [ ]: d1
```

```
In [ ]: for i in d1:  
      print(i,':',d1[i])
```

```
In [ ]: for i in enumerate (d1):  
      print(i)
```

```
In [ ]: d2={1:2,2.3:2:5,'nit':'nit',True:False,2+3j:4+5j}  
d2
```

```
In [ ]: d.items()
```

```
In [ ]: d
```

```
In [ ]: len(d.items())
```

```
In [ ]: id(d)
```

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

```
In [ ]: d
```

```
In [ ]: d[2]
```

```
In [ ]: d.popitem()
```

```
In [ ]: d
```

```
In [ ]: d.popitem[1]
```

```
In [ ]:
```

