python 1st code

```
In [2]: 10+5
```

```
Out[2]: 15
```

```
In [3]: 10-5
```

```
Out[3]: 5
```

```
In [4]: 10*5
```

```
Out[4]: 50
```

```
In [5]: 10/5
```

```
Out[5]: 2.0
```

```
In [6]: 10//5
```

```
Out[6]: 2
```

```
In [7]: (10+5)-7+6
```

```
Out[7]: 14
```

```
In [8]: (5+5)*5
```

```
Out[8]: 50
```

```
In [9]: 5+(5*5)
```

```
Out[9]: 30
```

```
In [10]: _+3
```

```
Out[10]: 33
```

```
In [11]: _*3
```

```
Out[11]: 99
```

```
In [12]: 1+1
         1+2
         1+3
         1+4
```

```
Out[12]: 5
```

```
In [13]: print(1+1)
         print(1+2)
```

```
2
3
```

```
In [14]: print(1+1)
         print(1+2)
         print(1+3)
         print(1+4)
         print(1-4)
         print(1+5)
```

```
2
3
4
5
-3
6
```

```
In [15]: a=10
         b=20
         c=a+b
         print(c)
```

```
30
```

```
In [16]: print(a)
         print(b)
print(c)
```

```
10
20
30
```

In [17]:
```
100=d
```

```
  Cell In[17], line 1
    100=d
    ^
SyntaxError: cannot assign to literal here. Maybe you meant '==' instead of '='?
```

In [224…
```
num1=20
num2=30
add=num1+num2
print(add)
```

```
50
```

In [226…
```
num1=20
num2=30
add=num1+num2
print('the addition of ==,'num1','and','num2','is==add')
```

```
  Cell In[226], line 4
    print('the addition of ==,'num1','and','num2','is==add')
                                                          ^
SyntaxError: unterminated string literal (detected at line 4)
```

In [228…
```
num1=20
num2=30
add=num1+num2
print('The addition of --',num1,'and',num2,'is==',add)
```

```
The addition of -- 20 and 30 is== 50
```

In [230…
```
import keyword
keyword.kwlist
```

Out[230…
```
['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 [232…
```
len(keyword.kwlist)
```

Out[232…
```
35
```

In [234…
```
for=45
for
```

```
  Cell In[234], line 1
    for=45
       ^
SyntaxError: invalid syntax
```

22nd oct

python variable concept =python identifier concept

In [236…

```
NIT=15
NIT
```

Out[236…

```
15
```

In [238…

```
NIT=20
NIT
```

Out[238…

```
20
```

In [240…

```
v=15
v
```

Out[240…

```
15
```

In [242…

```
print(v)
print(NIT)
```

```
15
20
```

In [244…

```
NIT
```

Out[244…

```
20
```

In [246…

```
V
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[246], line 1
----> 1 V

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

In [248…

```
V
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[248], line 1
----> 1 V

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

In [250…

```
v
```

Out[250…

```
15
```

In [252…

```
1var=20
1var
```

```
  Cell In[252], line 1
    1var=20
    ^
SyntaxError: invalid decimal literal
```

In [254…

```
var1=30
var1
```

Out[254…

```
30
```

In [256…

```
var$=20
var$
```

```
  Cell In[256], line 1
    var$=20
       ^
SyntaxError: invalid syntax
```

In [258…

```
var*=20
var*
```

```
  Cell In[258], line 2
    var*
        ^
SyntaxError: invalid syntax
```

```
In [260…  var_=60
          var_

Out[260…  60
```

```
In [262…  x_train, x_test= 80, 20, 50
          x_train
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[262], line 1
----> 1 x_train, x_test= 80, 20, 50
      2 x_train

ValueError: too many values to unpack (expected 2)
```

```
In [264…  x_train, x_test= 80,20
          print(x_train)
          print(x_test)

80
20
```

```
In [266…  a=10
          b=20
          c=30
          d=40
```

```
In [268…  a, b, c, d=10,20,30,40
          print(a)
          print(b)
          print(c)
          print(d)

10
20
30
40
```

```
In [270…  aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa=70
          print(aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa)

70
```

```
In [272…  ABC=100
          abc
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[272], line 2
      1 ABC=100
----> 2 abc

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

```
In [274…  nit@=6
          nit@
```

```
  Cell In[274], line 2
    nit@
        ^
SyntaxError: invalid syntax
```

```
In [276…  nit_=50
          nit_

Out[276…  50
```

```
In [278…  1nit=20
          1nit
```

```
  Cell In[278], line 1
    1nit=20
    ^
SyntaxError: invalid decimal literal
```

python identifier we are completed

23rd oct python data types

```
In [280… a=25
```

```
In [282… i=30
         i
```

```
Out[282… 30
```

```
In [284… type(i)
```

```
Out[284… int
```

```
In [286… print(type(i))
```

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

```
In [288… i
```

```
Out[288… 30
```

```
In [290… i1, i2= 20, 30
```

```
In [292… i+i1+i2
```

```
Out[292… 80
```

```
In [294… i-i1+i2
```

```
Out[294… 40
```

```
In [296… print(i)
         print(i1)
         print(i2)
```

```
30
20
30
```

```
In [298… i-(i2+i1)
```

```
Out[298… -20
```

integer data type we are completed

```
In [300… f=110.23
         f
```

```
Out[300… 110.23
```

```
In [302… type(f)
```

```
Out[302… float
```

```
In [304… f1, f2, f3 =2.3, 3.4, 5.1
```

```
In [306… print(f)
         print(f1)
         print(f2)
         print(f3)
```

```
110.23
2.3
3.4
5.1
```

```
In [308… 1f=1e0
```

```
  Cell In[308], line 1
    1f=1e0
    ^
SyntaxError: invalid decimal literal
```

```
In [310… f1=1e0
         f1
```

```
Out[310… 1.0
```

```
In [312… f2=2e1
```

```
f2
```

Out[312…
```
20.0
```

In [314…
```
f4=3e3
f4
```

Out[314…
```
3000.0
```

In [316…
```
f5=2.4e2
f5
```

Out[316…
```
240.0
```

In [318…
```
f6=2b3
```

```
  Cell In[318], line 1
    f6=2b3
       ^
SyntaxError: invalid decimal literal
```

Bool or Boolean

In [320…
```
b=True
b
```

Out[320…
```
True
```

In [322…
```
b1=false
b1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[322], line 1
----> 1 b1=false
      2 b1

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

In [324…
```
b1=False
b1
```

Out[324…
```
False
```

In [326…
```
print(b)
print(b1)
```

```
True
False
```

In [328…
```
True+False
```

Out[328…
```
1
```

In [330…
```
True-False
```

Out[330…
```
1
```

In [332…
```
False-True
```

Out[332…
```
-1
```

In [334…
```
True+True+True+False-True
```

Out[334…
```
2
```

In [336…
```
False*True
```

Out[336…
```
0
```

In [338…
```
True*True
```

Out[338…
```
1
```

In [340…
```
False/True
```

Out[340…
```
0.0
```

In [342…
```
True/False
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[342], line 1
----> 1 True/False

ZeroDivisionError: division by zero
```

complex data types

```
In [344…  c=1+20j
          c
```

```
Out[344…  (1+20j)
```

```
In [346…  type(c)
```

```
Out[346…  complex
```

```
In [348…  c
```

```
Out[348…  (1+20j)
```

```
In [350…  c.real
```

```
Out[350…  1.0
```

```
In [352…  c.imag
```

```
Out[352…  20.0
```

```
In [354…  c1=10+20j
          c2=30+40j
          print(c1+c2)
          print(c1-c2)
```

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

24th oct

```
In [356…  s='nit'
          s
```

```
Out[356…  'nit'
```

```
In [358…  type(s)
```

```
Out[358…  str
```

```
In [360…  s1="hello python"
          s1
```

```
Out[360…  'hello python'
```

```
In [362…  s2='''nit
               hello python'''
          s2
```

```
Out[362…  'nit\n     hello python'
```

```
In [364…  s1
```

```
Out[364…  'hello python'
```

```
In [366…  s1[0]
```

```
Out[366…  'h'
```

```
In [368…  s1[-4]
```

```
Out[368…  't'
```

```
In [370…  s1[4]
```

```
Out[370…  'o'
```

```
In [372…  s1[5]
```

```
Out[372… ' '
```

```
In [374… s1
```

```
Out[374… 'hello python'
```

```
In [376… s1[-7]
```

```
Out[376… ' '
```

```
In [378… s
```

```
Out[378… 'nit'
```

```
In [380… print(s[0])
         print(s[1])
         print(s[2])
```

```
n
i
t
```

```
In [382… s1
```

```
Out[382… 'hello python'
```

```
In [384… s1[2:7]
```

```
Out[384… 'llo p'
```

```
In [386… s2
```

```
Out[386… 'nit\n     hello python'
```

```
In [388… s3='dataanalyst'
         s3
```

```
Out[388… 'dataanalyst'
```

```
In [390… s3[0:10]
```

```
Out[390… 'dataanalys'
```

```
In [392… s3[0:11]
```

```
Out[392… 'dataanalyst'
```

```
In [394… s3
```

```
Out[394… 'dataanalyst'
```

```
In [396… s3[11]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[396], line 1
----> 1 s3[11]

IndexError: string index out of range
```

```
In [398… s3
```

```
Out[398… 'dataanalyst'
```

```
In [400… s3[9:12]
```

```
Out[400… 'st'
```

```
In [402… s3
```

```
Out[402… 'dataanalyst'
```

```
In [404… s3[0:11:2]
```

```
Out[404… 'dtaayt'
```

```
In [406… s3
```

```
Out[406… 'dataanalyst'
```

In [408…
```
s3[2:-2]
```

Out[408…
```
'taanaly'
```

In [410…
```
s3
```

Out[410…
```
'dataanalyst'
```

In [412…
```
print(s)
print(s1)
print(s2)
print(s3)
```

```
nit
hello python
nit
     hello python
dataanalyst
```

In [414…
```
import keyword
keyword.kwlist
```

Out[414…
```
['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 [416…
```
for i in s3:
     print(i)
```

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

python data type completed

python type casting | type conversion

In [418…
```
int(2.3)
```

Out[418…
```
2
```

In [420…
```
int(False)
```
Out[420…
```
0
```
In [422…
```
int(1+2j)
```
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[422], line 1
----> 1 int(1+2j)

TypeError: int() argument must be a string, a bytes-like object or a real number, not 'complex'
```
In [424…
```
int('10')
```
Out[424…
```
10
```
In [426…
```
int('ten')
```
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[426], line 1
----> 1 int('ten')

ValueError: invalid literal for int() with base 10: 'ten'
```
In [ ]:
```
s2
```
In [430…
```
del s2
```
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[430], line 1
----> 1 del s2

NameError: name 's2' is not defined
```
In [432…
```
s2
```
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[432], line 1
----> 1 s2

NameError: name 's2' is not defined
```
In [434…
```
np.nan
```
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[434], line 1
----> 1 np.nan

NameError: name 'np' is not defined
```
In [436…
```
import numpy as np
a = np.nan
```
In [438…
```
type(a)
```
Out[438…
```
float
```

25th oct

In [440…
```
float(3)
```
Out[440…
```
3.0
```
In [442…
```
float(True)
```
Out[442…
```
1.0
```
In [444…
```
float(1+2j)
```
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[444], line 1
----> 1 float(1+2j)

TypeError: float() argument must be a string or a real number, not 'complex'
```

```
In [446… float(3,4)

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[446], line 1
----> 1 float(3,4)

TypeError: float expected at most 1 argument, got 2
```

```
In [448… float('10')

Out[448… 10.0
```

```
In [450… float('ten')

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[450], line 1
----> 1 float('ten')

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

```
In [452… complex(10)

Out[452… (10+0j)
```

```
In [454… complex(10, 20)

Out[454… (10+20j)
```

```
In [456… complex(10, 20, 30)

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[456], line 1
----> 1 complex(10, 20, 30)

TypeError: complex() takes at most 2 arguments (3 given)
```

```
In [458… complex(2.3)

Out[458… (2.3+0j)
```

```
In [460… complex(True)

Out[460… (1+0j)
```

```
In [464… complex(False)

Out[464… 0j
```

```
In [462… complex('10')

Out[462… (10+0j)
```

```
In [466… bool(0)

Out[466… False
```

```
In [468… bool(2.3)

Out[468… True
```

```
In [470… bool()

Out[470… False
```

```
In [472… bool( )

Out[472… False
```

```
In [474… bool('nit')

Out[474… True
```

```
In [476… bool(10+2j)

Out[476… True
```

```
In [478… bool(0+0j)
```

Out[478… False

In [480…
```
print(str(2))
print(str(2.3))
print(str(True))
print(str(1+2j))
```

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

python type casting (convert all other datatype of ane datatype)

In [482…
```
index='HELLOPYTHON'
index
```

Out[482… 'HELLOPYTHON'

In [484…
```
index[:]
```

Out[484… 'HELLOPYTHON'

In [486…
```
index[::-1]
```

Out[486… 'NOHTYPOLLEH'

In [488…
```
index
```

Out[488… 'HELLOPYTHON'

In [490…
```
index[::-2]
```

Out[490… 'NHYOLH'

In [492…
```
index
```

Out[492… 'HELLOPYTHON'

In [494…
```
index[:-4]
```

Out[494… 'HELLOPY'

In [496…
```
index
```

Out[496… 'HELLOPYTHON'

In [502…
```
index[1:10:3]
```

Out[502… 'EOT'

In [500…
```
index
```

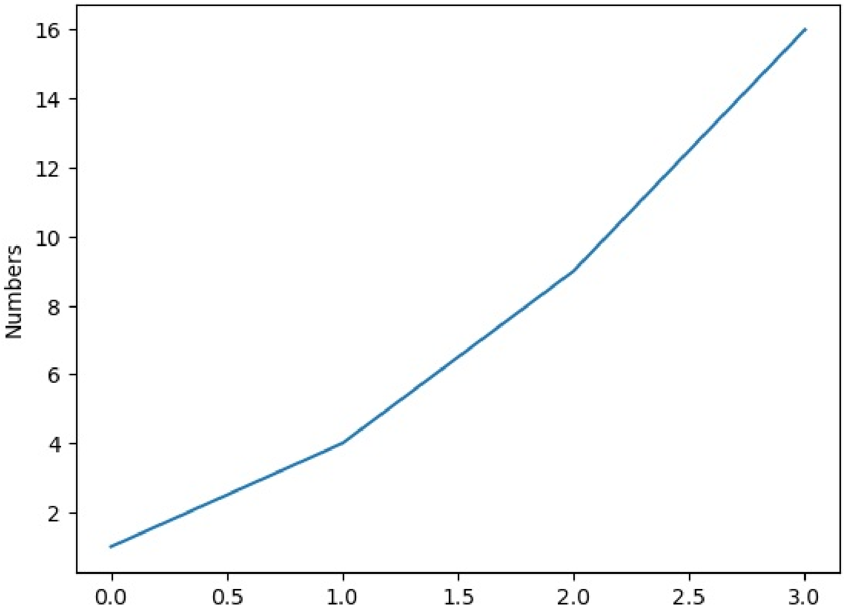
Out[500… 'HELLOPYTHON'

In [498…
```
index[::-2]
```

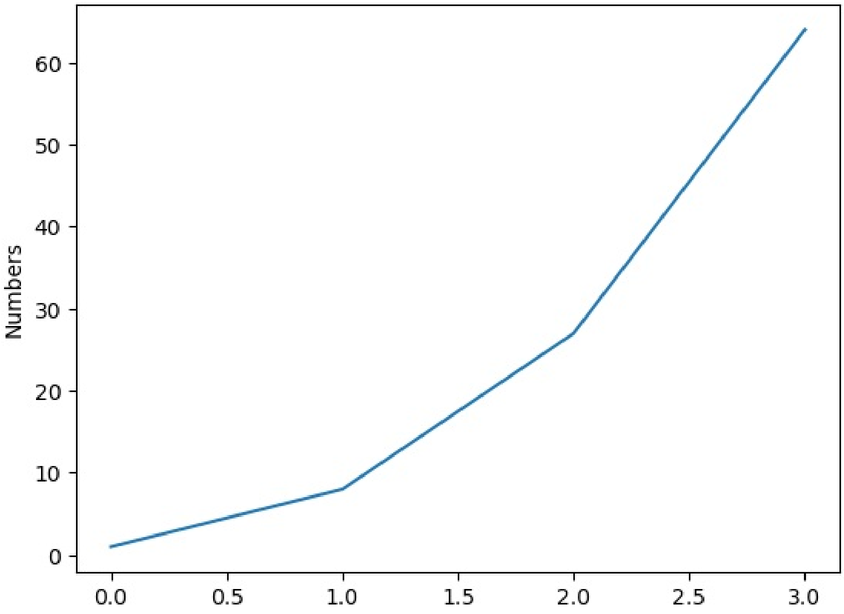
Out[498… 'NHYOLH'

python type casting we are completed

In [504…
```
import matplotlib.pyplot as plt
plt.plot([1,4,9,16])
plt.ylabel('Numbers')
plt.show()
```



In [506…
```
import matplotlib.pyplot as plt
plt.plot([1,8,27,64])
plt.ylabel('Numbers')
plt.show()
```



26th- Data structure

In [508…
```
l=[]
l
```

Out[508…
```
[]
```

In [510…
```
type(l)
```

Out[510…
```
list
```

In [512…
```
l
```

Out[512…
```
[]
```

In [514…
```
l.append(10)
```

In [518…
```
l
```

Out[518…
```
[10, 20, 30]
```

In [516…
```
l.append(20)
l.append(30)
```

In [522…
```
l
```

```
Out[522… [10, 20, 30]

In [524… l

Out[524… [10, 20, 30]

In [526… l.append(2.3)
          l

Out[526… [10, 20, 30, 2.3]

In [528… l.append(1+2j)
          l.append(True)
          l.append('nit')

In [530… l

Out[530… [10, 20, 30, 2.3, (1+2j), True, 'nit']

In [532… l.append(10)

In [534… l

Out[534… [10, 20, 30, 2.3, (1+2j), True, 'nit', 10]

In [536… l.remove(10)
          l

Out[536… [20, 30, 2.3, (1+2j), True, 'nit', 10]

In [538… l1=l.copy()
          l1

Out[538… [20, 30, 2.3, (1+2j), True, 'nit', 10]

In [540… l==l1

Out[540… True

In [542… l

Out[542… [20, 30, 2.3, (1+2j), True, 'nit', 10]

In [544… l.count(20)

Out[544… 1

In [546… l.append(20)

In [548… l

Out[548… [20, 30, 2.3, (1+2j), True, 'nit', 10, 20]

In [550… l

Out[550… [20, 30, 2.3, (1+2j), True, 'nit', 10, 20]

In [552… l.count(20)

Out[552… 2

In [554… l

Out[554… [20, 30, 2.3, (1+2j), True, 'nit', 10, 20]

In [556… l1

Out[556… [20, 30, 2.3, (1+2j), True, 'nit', 10]

In [558… l==l1

Out[558… False

In [560… print(len(l))
          print(len(l1))
```

```
8
7
```

```
In [562… l
```

```
Out[562… [20, 30, 2.3, (1+2j), True, 'nit', 10, 20]
```

```
In [564… l[:]
```

```
Out[564… [20, 30, 2.3, (1+2j), True, 'nit', 10, 20]
```

```
In [566… l[4]
```

```
Out[566… True
```

```
In [568… l
```

```
Out[568… [20, 30, 2.3, (1+2j), True, 'nit', 10, 20]
```

```
In [570… l.count(30)
```

```
Out[570… 1
```

```
In [572… l
```

```
Out[572… [20, 30, 2.3, (1+2j), True, 'nit', 10, 20]
```

28th-List data structure

```
In [575… l2=[]
         l2
```

```
Out[575… []
```

```
In [577… l2.append(1)
         l2.append(2.3)
         l2.append(True)
         l2.append('nit')
```

```
In [579… l2
```

```
Out[579… [1, 2.3, True, 'nit']
```

```
In [581… l3=l2.copy()
         l3
```

```
Out[581… [1, 2.3, True, 'nit']
```

```
In [583… l2
```

```
Out[583… [1, 2.3, True, 'nit']
```

```
In [585… l3
```

```
Out[585… [1, 2.3, True, 'nit']
```

```
In [587… len(l3)
```

```
Out[587… 4
```

```
In [589… l3.clear()
```

```
In [591… l3
```

```
Out[591… []
```

```
In [593… del l3
```

```
In [595… l2
```

```
Out[595… [1, 2.3, True, 'nit']
```

```
In [597… l3
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[597], line 1
----> 1 l3

NameError: name 'l3' is not defined
```

```
In [599… l2

Out[599… [1, 2.3, True, 'nit']

In [601… l2

Out[601… [1, 2.3, True, 'nit']

In [603… l2.append(1)

In [605… l2

Out[605… [1, 2.3, True, 'nit', 1]

In [607… l3=[]
         l3.append(10)

In [609… l3

Out[609… [10]

In [611… l2

Out[611… [1, 2.3, True, 'nit', 1]

In [613… l3.extend(l2)

In [615… l3

Out[615… [10, 1, 2.3, True, 'nit', 1]

In [617… l3.index(2.3)

Out[617… 2

In [619… l2

Out[619… [1, 2.3, True, 'nit', 1]

In [621… l3

Out[621… [10, 1, 2.3, True, 'nit', 1]

In [623… l2

Out[623… [1, 2.3, True, 'nit', 1]

In [625… l2.index('nit')

Out[625… 3

In [627… l2

Out[627… [1, 2.3, True, 'nit', 1]

In [629… l3

Out[629… [10, 1, 2.3, True, 'nit', 1]

In [631… l3.insert(3,False)

In [633… l3

Out[633… [10, 1, 2.3, False, True, 'nit', 1]

In [635… l3.pop(4)

Out[635… True

In [637… l3
```

```
Out[637… [10, 1, 2.3, False, 'nit', 1]

In [639… l3.pop(1)

Out[639… 1

In [641… l3

Out[641… [10, 2.3, False, 'nit', 1]

In [643… l4=[10, 100, 3, 45, 76, 24]

In [645… l4.sort()

In [647… l4

Out[647… [3, 10, 24, 45, 76, 100]

In [649… l4.sort(reverse=True)
         l4

Out[649… [100, 76, 45, 24, 10, 3]

In [651… l5=['z', 'm', 'c', 'w']
         l5

Out[651… ['z', 'm', 'c', 'w']

In [653… l5

Out[653… ['z', 'm', 'c', 'w']

In [655… l6=[1, 2, 3, 'a', 'z', 'w']

In [657… 16.sort()
         l6

           Cell In[657], line 1
             16.sort()
               ^
         SyntaxError: invalid decimal literal

In [659… l2

Out[659… [1, 2.3, True, 'nit', 1]

In [661… l3

Out[661… [10, 2.3, False, 'nit', 1]

In [663… l3.reverse()

In [664… l3

Out[664… [1, 'nit', False, 2.3, 10]

In [665… l2

Out[665… [1, 2.3, True, 'nit', 1]

In [666… l2[3]

Out[666… 'nit'

In [667… print(l2[3][0])
         print(l2[3][1])
         print(l2[3][2])

         n
         i
         t

In [668… l3

Out[668… [1, 'nit', False, 2.3, 10]

In [669… l3[2]=0
```

In [670…
```
l3
```
Out[670…
```
[1, 'nit', 0, 2.3, 10]
```
In [671…
```
l3[1]='mit'
l3
```
Out[671…
```
[1, 'mit', 0, 2.3, 10]
```
In [672…
```
for i in l3:
     print(i)
```
```
1
mit
0
2.3
10
```
In [673…
```
l6=['sbi', 'icic']
l7=['hdf', 'kotak']
```
In [674…
```
family_bank=l6+l7
family_bank
```
Out[674…
```
['sbi', 'icic', 'hdf', 'kotak']
```
In [675…
```
l3
```
Out[675…
```
[1, 'mit', 0, 2.3, 10]
```
In [676…
```
for i in enumerate (l3):
    print(i)

```
```
(0, 1)
(1, 'mit')
(2, 0)
(3, 2.3)
(4, 10)
```
In [677…
```
l
```
Out[677…
```
[20, 30, 2.3, (1+2j), True, 'nit', 10, 20]
```
In [678…
```
l[:]
```
Out[678…
```
[20, 30, 2.3, (1+2j), True, 'nit', 10, 20]
```
In [679…
```
l[::-1]
```
Out[679…
```
[20, 10, 'nit', True, (1+2j), 2.3, 30, 20]
```
In [680…
```
l[::-2]
```
Out[680…
```
[20, 'nit', (1+2j), 30]
```
In [681…
```
l
```
Out[681…
```
[20, 30, 2.3, (1+2j), True, 'nit', 10, 20]
```
In [682…
```
l[l7]
```
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[682], line 1
----> 1 l[l7]

TypeError: list indices must be integers or slices, not list
```
In [683…
```
l
```
Out[683…
```
[20, 30, 2.3, (1+2j), True, 'nit', 10, 20]
```
In [684…
```
l[2:]
```
Out[684…
```
[2.3, (1+2j), True, 'nit', 10, 20]
```
In [685…
```
l
```
Out[685…
```
[20, 30, 2.3, (1+2j), True, 'nit', 10, 20]
```

```
In [686… l[:7]

Out[686… [20, 30, 2.3, (1+2j), True, 'nit', 10]

In [687… l[0:7:2]

Out[687… [20, 2.3, True, 10]
```

List data structure we are completed

```
In [689… t=()
         t

Out[689… ()

In [690… type(t)

Out[690… tuple

In [691… type(l)

Out[691… list

In [692… t1=tuple()
         type(t1)

Out[692… tuple

In [693… t=(10, 10, 20, 30)

In [694… t

Out[694… (10, 10, 20, 30)

In [695… icic=(1234,'cizp', '4thmar')
         icic

Out[695… (1234, 'cizp', '4thmar')

In [696… i=icic.copy()

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[696], line 1
----> 1 i=icic.copy()

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

In [ ]:  t

In [697… t[0]=100

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[697], line 1
----> 1 t[0]=100

TypeError: 'tuple' object does not support item assignment

In [ ]:  t[0]

In [698… t1=(10, 1.2, 'nit' ,1+2j, True)
         t1

Out[698… (10, 1.2, 'nit', (1+2j), True)

In [699… t1.count(10)

Out[699… 1

In [700… t1

Out[700… (10, 1.2, 'nit', (1+2j), True)

In [701… for i in t1:
             print(i)
```

```
10
1.2
nit
(1+2j)
True
```

```
In [762…  for i in enumerate(t1):
              print(i)
```

```
(0, 10)
(1, 1.2)
(2, 'nit')
(3, (1+2j))
(4, True)
```

```
In [764…  t

Out[764…  (10, 10, 20, 30)

In [766…  t[:]

Out[766…  (10, 10, 20, 30)

In [768…  t

Out[768…  (10, 10, 20, 30)

In [770…  t4=t*4
          t4

Out[770…  (10, 10, 20, 30, 10, 10, 20, 30, 10, 10, 20, 30, 10, 10, 20, 30)

In [772…  l

Out[772…  [20, 30, 2.3, (1+2j), True, 'nit', 10, 20]

In [774…  l3

Out[774…  [1, 'mit', 0, 2.3, 10]

In [776…  t

Out[776…  (10, 10, 20, 30)

In [778…  t

Out[778…  (10, 10, 20, 30)

In [780…  print(id(l))
          print(id(t))
```

```
2259264294080
2259251439200
```

tuple completed

```
In [783…  s={}
          s

Out[783…  {}

In [785…  type(s)

Out[785…  dict

In [787…  s1={100,20,3,15,47}
          s1

Out[787…  {3, 15, 20, 47, 100}

In [789…  s2={2.3,4.5,1.3}
          s2

Out[789…  {1.3, 2.3, 4.5}

In [791…  s3={'z', 'm', 'a','x'}
          s3

Out[791…  {'a', 'm', 'x', 'z'}
```

In [793…
```
s4={10,2.3,'a',5,6.7}
s4
```

Out[793…
```
{10, 2.3, 5, 6.7, 'a'}
```

In [795…
```
print(s1)
print(s2)
print(s3)
print(s4)
```

```
{3, 100, 20, 47, 15}
{1.3, 2.3, 4.5}
{'z', 'm', 'x', 'a'}
{2.3, 5, 6.7, 'a', 10}
```

In [797…
```
for i in s1:
    print(i)
```

```
3
100
20
47
15
```

In [799…
```
for i in enumerate(s1):
    print(i)
```

```
(0, 3)
(1, 100)
(2, 20)
(3, 47)
(4, 15)
```

In [801…
```
s4
```

Out[801…
```
{10, 2.3, 5, 6.7, 'a'}
```

In [803…
```
s4.add(10,2.3)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[803], line 1
----> 1 s4.add(10,2.3)

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

In [805…
```
s4.add(10)
s4.add(20)
s4.add(2.3)
```

In [807…
```
s4
```

Out[807…
```
{10, 2.3, 20, 5, 6.7, 'a'}
```

In [809…
```
s1
```

Out[809…
```
{3, 15, 20, 47, 100}
```

In [813…
```
s1.add(4)
s1
```

Out[813…
```
{3, 4, 15, 20, 47, 100}
```

In [815…
```
s1
```

Out[815…
```
{3, 4, 15, 20, 47, 100}
```

In [817…
```
s2
```

Out[817…
```
{1.3, 2.3, 4.5}
```

In [819…
```
s3
```

Out[819…
```
{'a', 'm', 'x', 'z'}
```

In [821…
```
s4
```

Out[821…
```
{10, 2.3, 20, 5, 6.7, 'a'}
```

In [823…
```
len(s4)
```

```
Out[823… 6

In [825… s4.clear()

In [827… s4

Out[827… set()

In [829… len(s4)

Out[829… 0

In [831… del s4

In [833… s4

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[833], line 1
----> 1 s4

NameError: name 's4' is not defined

In [835… s1

Out[835… {3, 4, 15, 20, 47, 100}

In [839… s4=s1.copy()
         s4

Out[839… {3, 4, 15, 20, 47, 100}

In [841… s1==s4

Out[841… True

In [843… s1

Out[843… {3, 4, 15, 20, 47, 100}

In [845… s1.remove(100)

In [847… s1

Out[847… {3, 4, 15, 20, 47}

In [849… s1[:]

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[849], line 1
----> 1 s1[:]

TypeError: 'set' object is not subscriptable

In [851… s1[0]

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[851], line 1
----> 1 s1[0]

TypeError: 'set' object is not subscriptable

In [853… s2

Out[853… {1.3, 2.3, 4.5}

In [855… s3

Out[855… {'a', 'm', 'x', 'z'}

In [857… s3.pop()

Out[857… 'z'

In [859… s2

Out[859… {1.3, 2.3, 4.5}
```

```
In [861… s2.pop()

Out[861… 1.3
```

```
In [863… s4

Out[863… {3, 4, 15, 20, 47, 100}
```

```
In [865… s4.pop(3)

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[865], line 1
----> 1 s4.pop(3)

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

```
In [867… s4.pop(3)

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[867], line 1
----> 1 s4.pop(3)

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

```
In [871… s4[4]

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[871], line 1
----> 1 s4[4]

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

```
In [873… s3

Out[873… {'a', 'm', 'x'}
```

```
In [875… 'a' in s3

Out[875… True
```

```
In [877… s3

Out[877… {'a', 'm', 'x'}
```

```
In [879… s3.index('m')

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[879], line 1
----> 1 s3.index('m')

AttributeError: 'set' object has no attribute 'index'
```

```
In [881… a={1,2,3,4,5}
         b={4,5,6,7,8}
         c={8,9,10}
```

```
In [883… type(c)

Out[883… set
```

```
In [885… a.union(b)

Out[885… {1, 2, 3, 4, 5, 6, 7, 8}
```

```
In [887… print(a)
         print(b)
         print(c)

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

```
In [889… d_union=a.union(b)
         d_union

Out[889… {1, 2, 3, 4, 5, 6, 7, 8}
```

```
In [891…  print(a)
          print(b)
          print(c)
          print(d_union)

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

In [893…  b.union(a,c)

Out[893…  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [895…  a|b

Out[895…  {1, 2, 3, 4, 5, 6, 7, 8}

In [897…  a|b|c

Out[897…  {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In [901…  print(a)
          print(b)
          print(c)
          print(d_union)

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

In [903…  a.update(b)

In [907…  print(a)
          print(b)
          print(c)
          print(d_union)

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

In [909…  c.update(b)

In [911…  print(c)

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

In [913…  c.update(4)

          ---------------------------------------------------------------------------
          TypeError                                 Traceback (most recent call last)
          Cell In[913], line 1
          ----> 1 c.update(4)

          TypeError: 'int' object is not iterable

In [915…  a2 = {1,2,3,4,5}
          b2 = {4,5,6,7,8}
          c2 = {8,9,10}

In [917…  a2-b2

Out[917…  {1, 2, 3}

In [919…  b2-a2

Out[919…  {6, 7, 8}

In [921…  a2-c2

Out[921…  {1, 2, 3, 4, 5}

In [923…  a2 = {1,2,3,4,5}
          b2 = {4,5,6,7,8}
          c2 = {8,9,10}

In [925…  b2.difference(c2)

Out[925…  {4, 5, 6, 7}
```

```
In [929…  a2.symmetric_difference(b2)

Out[929…  {1, 2, 3, 6, 7, 8}

In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js