```
In [100]: print("This is my first python program.")

          This is my first python program.

In [101]: student = "sam"

In [102]: print(student)

          sam

In [103]: student ="Matt"

In [104]: student

Out[104]: 'Matt'

In [105]: student ="Bob"

In [106]: student

Out[106]: 'Bob'

In [107]: a=10
          a

Out[107]: 10

In [108]: type(a)

Out[108]: int

In [109]: a=1.2
          a
```

```
            type(a)

Out[109]:   float

In [110]:   a

Out[110]:   1.2

In [111]:   a=True
            type(a)

Out[111]:   bool

In [112]:   a

Out[112]:   True

In [113]:   a=2+1j
            type(a)

Out[113]:   complex

In [114]:   a="Hello World"
            type(a)

Out[114]:   str

In [115]:   a=20, b=30
            a+b

              File "<ipython-input-115-5ccf2ab99565>", line 1
                a=20, b=30
               ^
            SyntaxError: can't assign to literal

In [116]:   a=10
            b=20
```

```
In [117]: a,b

Out[117]: (10, 20)

In [ ]:

In [118]: a=10
          a

Out[118]: 10

In [119]: b=20
          b

Out[119]: 20

In [120]: a,b

Out[120]: (10, 20)

In [121]: a=10
          b=20

In [122]: a,b

Out[122]: (10, 20)

In [ ]:

In [123]: a+b

Out[123]: 30

In [124]: a-b

Out[124]: -10
```

```
In [125]: a*b
Out[125]: 200

In [126]: a/b
Out[126]: 0.5

In [127]: #Arithmatic operators

In [128]: a=20
          b=30

In [129]: a,b
Out[129]: (20, 30)

In [130]: a+b
Out[130]: 50

In [131]: a-b
Out[131]: -10

In [132]: #Relational Operators (<,>,==,!=)

In [133]: a=50
          b=100

In [134]: a>b
Out[134]: False

In [135]: a<b
```

Out[135]: True

In [136]:
```
a==b
```

Out[136]: False

In [137]:
```
a!=b
```

Out[137]: True

In [138]:
```
#Logical operatore &(and),|(or)
```

In [139]:
```
a= True
b= False
```

In [140]:
```
a&b
```

Out[140]: False

In [141]:
```
a&a
```

Out[141]: True

In [142]:
```
a|b
```

Out[142]: True

In [143]:
```
s1='This'
s1
```

Out[143]: 'This'

In [144]:
```
s2="This"
s2
```

Out[144]: 'This'

```
In [145]: s3='''This'''
          s3
```

```
Out[145]: 'This'
```

```
In [146]: s4='''
          This is
          string
          lot
          in'''
          s4
```

```
Out[146]: '\nThis is\nstring\nlot\nin'
```

```
In [147]: my_string="This is Pyton"
          my_string
```

```
Out[147]: 'This is Pyton'
```

```
In [148]: my_string[0:7]
```

```
Out[148]: 'This is'
```

```
In [149]: my_string[-1]
```

```
Out[149]: 'n'
```

```
In [150]: my_string[0:]
```

```
Out[150]: 'This is Pyton'
```

```
In [151]: len(my_string)
```

```
Out[151]: 13
```

```
In [152]: my_string.lower
```

```
Out[152]: <function str.lower()>
```

```
In [153]: my_string.lower()

Out[153]: 'this is pyton'

In [154]: my_string_upper()

          ---------------------------------------------------------------------------
          ----
          NameError                                 Traceback (most recent call l
          ast)
          <ipython-input-154-c1e94f41546f> in <module>
          ----> 1 my_string_upper()

          NameError: name 'my_string_upper' is not defined

In [155]: my_string.upper()

Out[155]: 'THIS IS PYTON'

In [156]: my_string.replace('y','i')

Out[156]: 'This is Piton'

In [157]: my_string.count('i')

Out[157]: 2

In [158]: my_string.count('to')

Out[158]: 1

In [159]: my_string.find("This")

Out[159]: 0

In [160]: my_string.find('is')
```

```
Out[160]: 2

In [161]: my_string.find('Piton')

Out[161]: -1

In [162]: s1= "This is Python"

In [ ]: 

In [163]: s1

Out[163]: 'This is Python'

In [164]: s1.find('Python')

Out[164]: 8

In [165]: s2="Python is the most beautiful language"
          s2

Out[165]: 'Python is the most beautiful language'

In [166]: s2.find('most')

Out[166]: 14

In [167]: s2.find('language')

Out[167]: 29

In [168]: fruit=[apple banana grape]
          fruit.split(',')

            File "<ipython-input-168-e9ac4af8311d>", line 1
              fruit=[apple banana grape]
```

```
                ^
SyntaxError: invalid syntax
```

In [169]:

```
fruit=[apple,banana,grape]
fruit
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-169-166bda513493> in <module>
----> 1 fruit=[apple,banana,grape]
      2 fruit

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

In [ ]:

```
fruit = 'apple,banana,grp'
fruit
```

In [ ]:

```
fruit.split(' ')
```

In [ ]:

```
fruit.split('  ')
```

In [ ]:

```
fruit.split(',')
```

In [ ]:

```
fruit.split(':')
```

In [ ]:

```
fruit.split('\')
```

In [ ]:

```
fruit.split('*')
```

In [ ]:

```
fruit.split(',')
```

In [ ]:

```
f1='a,b,c,d'
f1
```

In [ ]:

```
f1.split(',')
```

In [170]:

```
f2=a:b:c:d
f2
```

```
  File "<ipython-input-170-f3e053e99b26>", line 1
    f2=a:b:c:d
        ^
SyntaxError: invalid syntax
```

In [ ]:

```
f2='a:b:c:d'
f2
```

In [ ]:

```
f2.split(':')
```

In [ ]:

```
f1='final event is here'
```

In [ ]:

```
f1.split('e')
```

In [ ]:

```
f1.split(' ')
```

In [ ]:

```
f1.split(',')
```

In [ ]:

```
tup1=(1,True,1.2,3+2j)
tup1
```

In [ ]:

```
tup1(type)
```

In [171]:

```
type(tup1)
```

Out[171]:

```
tuple
```

In [172]:

```
tup[0]
```

```
tup[3]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-172-255e5f77d945> in <module>
----> 1 tup[0]
      2 tup[3]

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

In [ ]:
```
tup1[0]
tup1[2]
```

In [ ]:
```
tup1
```

In [ ]:
```
tup[3]
```

In [ ]:
```
tup1[3]
```

In [ ]:
```
tup1[-1]
```

In [ ]:
```
tup1[-2]
```

In [ ]:
```
tup1[1:4]
```

In [ ]:
```
tup1=(1,2,'s','f','ox','True', 2.3)
```

In [ ]:
```
tup1
```

In [173]:
```
tup1[3:6]
```

Out[173]:
```
('f', 'ox', 'True')
```

In [174]:
```
tup1[0]= 'The'
```

```
---------------------------------------------------------------------------
----
TypeError                                 Traceback (most recent call l
ast)
<ipython-input-174-c723ec3d14f8> in <module>
----> 1 tup1[0]= 'The'

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

In [ ]:
```
len(tup1)
```

In [ ]:
```

```

In [175]:
```
len(tup1)
```

Out[175]:
```
7
```

In [176]:
```
tup1=(1,2,3)
tup2=(4,5,6)
tup1+tup2
```

Out[176]:
```
(1, 2, 3, 4, 5, 6)
```

In [177]:
```
tup1,tup2
```

Out[177]:
```
((1, 2, 3), (4, 5, 6))
```

In [178]:
```
tup1*tup2
```

```
---------------------------------------------------------------------------
----
TypeError                                 Traceback (most recent call l
ast)
<ipython-input-178-6c065929de1c> in <module>
----> 1 tup1*tup2
```

```
TypeError: can't multiply sequence by non-int of type 'tuple'
```

```
In [179]: tup2,tup1
```

```
Out[179]: ((4, 5, 6), (1, 2, 3))
```

```
In [180]: tup2+tup1
```

```
Out[180]: (4, 5, 6, 1, 2, 3)
```

```
In [181]: tup1*3
```

```
Out[181]: (1, 2, 3, 1, 2, 3, 1, 2, 3)
```

```
In [182]: tup2*2+tup1
```

```
Out[182]: (4, 5, 6, 4, 5, 6, 1, 2, 3)
```

```
In [183]: min(tup1)
```

```
Out[183]: 1
```

```
In [184]: max(tup1)
```

```
Out[184]: 3
```

```
In [185]: max(tup2)
```

```
Out[185]: 6
```

```
In [186]: l1 = [1,1.2,True,False,'ans',3.5j]
          l1
```

```
Out[186]: [1, 1.2, True, False, 'ans', 3.5j]
```

```
In [187]: type(l1)
```

```
Out[187]: list

In [188]: l1[2]

Out[188]: True

In [189]: l1[0]

Out[189]: 1

In [190]: l1[2:5]

Out[190]: [True, False, 'ans']

In [191]: l1

Out[191]: [1, 1.2, True, False, 'ans', 3.5j]

In [192]: l1[1]= 2.2

In [193]: l1.append('no')

In [194]: l1

Out[194]: [1, 2.2, True, False, 'ans', 3.5j, 'no']

In [195]: l1.pop()

Out[195]: 'no'

In [196]: l1

Out[196]: [1, 2.2, True, False, 'ans', 3.5j]

In [198]: l1
```

```
Out[198]: [1, 2.2, True, False, 'ans', 3.5j]
```

```
In [200]: l1.reverse()
          l1
```

```
Out[200]: [1, 2.2, True, False, 'ans', 3.5j]
```

```
In [201]: l1.insert(2.2,"xyz")
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-201-7104ea01a4b8> in <module>
----> 1 l1.insert(2.2,"xyz")

TypeError: integer argument expected, got float
```

```
In [203]: l1.insert(1,'xyz')
```

```
In [204]: l1
```

```
Out[204]: [1, 'xyz', 'xyz', 2.2, True, False, 'ans', 3.5j]
```

```
In [205]: l1.insert(0,'sas')
```

```
In [206]: l1
```

```
Out[206]: ['sas', 1, 'xyz', 'xyz', 2.2, True, False, 'ans', 3.5j]
```

```
In [207]: l1.sort()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-207-b8f5f256bbcf> in <module>
```

```
----> 1 l1.sort()

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

```
In [208]: l1=["mango","zumba","pine","turkey"]
          l1

Out[208]: ['mango', 'zumba', 'pine', 'turkey']
```

```
In [209]: l1.sort()
```

```
In [210]: l1

Out[210]: ['mango', 'pine', 'turkey', 'zumba']
```

```
In [211]: l2=[2,3,8,1,2,9,1,0]
          l2

Out[211]: [2, 3, 8, 1, 2, 9, 1, 0]
```

```
In [212]: l2.sort()
```

```
In [213]: l2

Out[213]: [0, 1, 1, 2, 2, 3, 8, 9]
```

```
In [214]: l1=[1,2,3]
          l2=["adas","sd","yrhr"]
          l1+l2

Out[214]: [1, 2, 3, 'adas', 'sd', 'yrhr']
```

```
In [215]: l1 = [1,'a',3.4]
          l1*3

Out[215]: [1, 'a', 3.4, 1, 'a', 3.4, 1, 'a', 3.4]
```

```
In [217]: l2=["apple","banana","lime"]
          l2*5+l1
```

```
Out[217]: ['apple',
           'banana',
           'lime',
           'apple',
           'banana',
           'lime',
           'apple',
           'banana',
           'lime',
           'apple',
           'banana',
           'lime',
           'apple',
           'banana',
           'lime',
           1,
           'a',
           3.4]
```

```
In [218]: d1={'apple':100,'banana':20,'brocoli':50}
          d1
```

```
Out[218]: {'apple': 100, 'banana': 20, 'brocoli': 50}
```

```
In [219]: type(d1)
```

```
Out[219]: dict
```

```
In [220]: d2={10:'sert', 20:'jhgjhj'}
          d2
```

```
Out[220]: {10: 'sert', 20: 'jhgjhj'}
```

```
In [221]: type(d2)
```

```
Out[221]: dict
```

```
In [222]: d1.keys

Out[222]: <function dict.keys>

In [224]: d1.keys()

Out[224]: dict_keys(['apple', 'banana', 'brocoli'])

In [226]: d2.values()

Out[226]: dict_values(['sert', 'jhgjhj'])

In [227]: d1['lime'=2000]
          d1

  File "<ipython-input-227-0f87cce82414>", line 1
    d1['lime'=2000]
             ^
SyntaxError: invalid syntax

In [228]: d1['lime']=2000
          d1

Out[228]: {'apple': 100, 'banana': 20, 'brocoli': 50, 'lime': 2000}

In [229]: d1[100]='kiwi'
          d1

Out[229]: {'apple': 100, 'banana': 20, 'brocoli': 50, 'lime': 2000, 100: 'kiwi'}

In [230]: d1['apple']= 50
          d1

Out[230]: {'apple': 50, 'banana': 20, 'brocoli': 50, 'lime': 2000, 100: 'kiwi'}

In [231]: d1['100']='ki'
```

```
d1
```

Out[231]:

```
{'apple': 50,
 'banana': 20,
 'brocoli': 50,
 'lime': 2000,
 100: 'kiwi',
 '100': 'ki'}
```

In [232]:

```
d1['zim']= 100
d1
```

Out[232]:

```
{'apple': 50,
 'banana': 20,
 'brocoli': 50,
 'lime': 2000,
 100: 'kiwi',
 '100': 'ki',
 'zim': 100}
```

In [233]:

```
d1['dds'] = 100
d1
```

Out[233]:

```
{'apple': 50,
 'banana': 20,
 'brocoli': 50,
 'lime': 2000,
 100: 'kiwi',
 '100': 'ki',
 'zim': 100,
 'dds': 100}
```

In [234]:

```
l1=[10:'a',20:'b',30:'c',40:'d']
l1
```

```
  File "<ipython-input-234-93e65189ac44>", line 1
    l1=[10:'a',20:'b',30:'c',40:'d']
          ^
SyntaxError: invalid syntax
```

```
In [235]: l1={10:'a',20:'b',30:'c',40:'d'}
          l1

Out[235]: {10: 'a', 20: 'b', 30: 'c', 40: 'd'}

In [ ]:

In [236]: l2={50:'e',60:'f'}
          l2

Out[236]: {50: 'e', 60: 'f'}

In [237]: l1.update(l3)

          ------------------------------------------------------------------------
          ----
          NameError                                 Traceback (most recent call l
          ast)
          <ipython-input-237-66f54763d455> in <module>
          ----> 1 l1.update(l3)

          NameError: name 'l3' is not defined

In [238]: l1.update(l2)

In [239]: l1

Out[239]: {10: 'a', 20: 'b', 30: 'c', 40: 'd', 50: 'e', 60: 'f'}

In [240]: l1.pop('f')
          l1

          ------------------------------------------------------------------------
          ----
          KeyError                                  Traceback (most recent call l
          ast)
          <ipython-input-240-44c5ddcb8a11> in <module>
```

```
----> 1 l1.pop('f')
      2 l1

KeyError: 'f'
```

In [241]:
```
l1.pop()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-241-f85fa3b895f0> in <module>
----> 1 l1.pop()

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

In [242]:
```
l1.pop('e')
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-242-4d0d9a5b8164> in <module>
----> 1 l1.pop('e')

KeyError: 'e'
```

In [243]:
```
l1
```

Out[243]:
```
{10: 'a', 20: 'b', 30: 'c', 40: 'd', 50: 'e', 60: 'f'}
```

In [244]:
```
l1.pop(10)
```

Out[244]:
```
'a'
```

In [245]:
```
l1
```

Out[245]:
```
{20: 'b', 30: 'c', 40: 'd', 50: 'e', 60: 'f'}
```

```
In [246]: l2 = {'banana':10, 'water':20, 'melon'= 30}
          l2

            File "<ipython-input-246-ebd3eeeea0ab>", line 1
              l2 = {'banana':10, 'water':20, 'melon'= 30}
                                                    ^

          SyntaxError: invalid syntax
```

```
In [247]: l2 = {'banana':10,'water':20,'melon':30}
          l2

Out[247]: {'banana': 10, 'water': 20, 'melon': 30}
```

```
In [248]: l2.pop('banana')
          l2

Out[248]: {'water': 20, 'melon': 30}
```

```
In [249]: s1={1,1.2,'a',True}
          s1

Out[249]: {1, 1.2, 'a'}
```

```
In [250]: type(s1)

Out[250]: set
```

```
In [251]: s2={}
          s2

Out[251]: {}
```

```
In [252]: type(s2)

Out[252]: dict
```

```
In [253]: s1={1,1.2,'a',True,True,Fakse,1,1.2}
```

```
s1
```

```
---------------------------------------------------------------------------
----
NameError                                 Traceback (most recent call l
ast)
<ipython-input-253-eea08a0d62fb> in <module>
----> 1 s1={1,1.2,'a',True,True,Fakse,1,1.2}
      2 s1

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

In [254]:
```
s1={1,1.2,'a',True,True,False,1,1.2}
s1
```

Out[254]:
```
{1, 1.2, False, 'a'}
```

In [255]:
```
s2={1,2,3,4}
s2
```

Out[255]:
```
{1, 2, 3, 4}
```

In [256]:
```
s2={1,2,3,4,3,4,5}
```

In [257]:
```
s2
```

Out[257]:
```
{1, 2, 3, 4, 5}
```

In [279]:
```
s3= {1,1.2,'a',True,True,False,1,1.2}
s3
```

Out[279]:
```
{1, 1.2, False, 'a'}
```

In [259]:
```
{1,1.2,'a',True,True,False,False,1,1.2}
```

Out[259]:
```
{1, 1.2, False, 'a'}
```

```
In [260]: s3.add(1000)
          s3

Out[260]: {1, 1.2, 1000, False, 'a'}

In [261]: s3.add(True)
          s3

Out[261]: {1, 1.2, 1000, False, 'a'}

In [262]: s3.update([10,20,30,'a'])
          s3

Out[262]: {1, 1.2, 10, 1000, 20, 30, False, 'a'}

In [263]: s3.remove(False)
          s3

Out[263]: {1, 1.2, 10, 1000, 20, 30, 'a'}

In [264]: s3.remove(1000)
          s3

Out[264]: {1, 1.2, 10, 20, 30, 'a'}

In [265]: s1={1,2,3}
          s2 = {'a','b','c'}

In [266]: s1

Out[266]: {1, 2, 3}

In [268]: s2

Out[268]: {'a', 'b', 'c'}

In [270]: s1.union(s2)
```

```
Out[270]: {1, 2, 3, 'a', 'b', 'c'}

In [271]: s4={5,6,7,8,9}

In [272]: s1.intersection(s4)

Out[272]: set()

In [273]: s3={6,7,8}

In [274]: s3.intersection.s4

          ---------------------------------------------------------------------------
          AttributeError                            Traceback (most recent call l
          ast)
          <ipython-input-274-e40111c82f1c> in <module>
          ----> 1 s3.intersection.s4

          AttributeError: 'builtin_function_or_method' object has no attribute 's
          4'

In [275]: s3

Out[275]: {6, 7, 8}

In [276]: s4

Out[276]: {5, 6, 7, 8, 9}

In [277]: s3.intersection(s4)

Out[277]: {6, 7, 8}

In [278]: s4.intersection(s3)

Out[278]: {6, 7, 8}
```

In [280]:
```
s3= {1,1.2,'a',True,True,False,1,1.2, False}
s3
```

Out[280]:
```
{1, 1.2, False, 'a'}
```

In [283]:
```
a=10
b=20
if a<b:
    print("a is less than b")
```

```
a is less than b
```

In [284]:
```
a=100
b=20
if a<b:
    print("a is less than b")
else:
    print("a is greater than b")

```

```
a is greater than b
```

In [290]:
```
a=10
b=20
c=30
if (a>b) & (a>c):
    print("a is greater.")
elif (b>a) & (b>c):
    print("b is greater")
else:
    print("c is greater")
```

```
c is greater
```

In [291]:
```
#if statement with tuple
```

In [293]:
```
tup1=('a','b','c')
```

```
if 'a' in tup1:
    print ("a is present in tup1.")
else:
    print ("a is not in tup1")
```

```
a is present in tup1.
```

In [294]:

```
#if with list
```

In [296]:

```
l1= ['a','b', 'c']
if l1[1] == 'b':
    l1[]='z'
```

```
  File "<ipython-input-296-e2b1d603bfc4>", line 3
    l1[]='z'
       ^
SyntaxError: invalid syntax
```

In [299]:

```
l1= ['a','b', 'c']
if l1[1] == 'b':
   l1[1]='z'
```

In [300]:

```
l1
```

Out[300]:

```
['a', 'z', 'c']
```

In [301]:

```
#if with dictionary
```

In [302]:

```
d1 ={'k1':10,'k2':20, 'k3':30}
d1
```

Out[302]:

```
{'k1': 10, 'k2': 20, 'k3': 30}
```

In [304]:

```
if d1[k3]==30:
    d1[k3]= d1['k3']+100
```

```
---------------------------------------------------------------------------
```

```
----
NameError                                 Traceback (most recent call l
ast)
<ipython-input-304-e122ce6b6a71> in <module>
----> 1 if d1[k3]==30:
      2     d1[k3]= d1['k3']+100

NameError: name 'k3' is not defined
```

In [305]:

```
if d1[k3]==30:
    d1[k3]= d1[k3]+100
```

```
---------------------------------------------------------------------------
----
NameError                                 Traceback (most recent call l
ast)
<ipython-input-305-ea7b7ab6df8a> in <module>
----> 1 if d1[k3]==30:
      2     d1[k3]= d1[k3]+100

NameError: name 'k3' is not defined
```

In [306]:

```
if d1['k3']==30:
    d1['k3']= d1['k3']+100
d1
```

Out[306]:

```
{'k1': 10, 'k2': 20, 'k3': 130}
```

In [307]:

```
i=1
while i<=10:
   print(i)
   i=i+1
```

```
1
2
3
4
5
6
```

```
7
8
9
10
```

In [309]:

```
i=1
n=2
while i<=10:
    print(n,'*',i,'=',n*i)
    i=i+1
```

```
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20
```

In [310]:

```
#while with list
```

In [311]:

```
l1= [1,2,3,4,5]
i=0
while i<len(l1):
     l1[i]=l1[i]+100
     i=i+1
```

In [312]:

```
l1
```

Out[312]:

```
[101, 102, 103, 104, 105]
```

In [313]:

```
l1 = ['a','b','c','d']
for i in l1:
    print (i)
```

```
a
b
c
d
```

In [314]:
```
l1= ['orange','black','white']
l2= ['chair','book','laptop']
```

In [315]:
```
for i in l1:
    for j in l2:
        print(i,j)
```

```
orange chair
orange book
orange laptop
black chair
black book
black laptop
white chair
white book
white laptop
```

In [316]:
```
def hello()
   print("Hello world")
```

```
  File "<ipython-input-316-523cf6a61f46>", line 1
    def hello()
               ^
SyntaxError: invalid syntax
```

In [317]:
```
def hello():
   print("Hello world")
```

In [318]:
```
hello()
```

```
Hello world
```

```
In [319]: def add_10(x)
              return x+10

            File "<ipython-input-319-62d50455b1ab>", line 1
              def add_10(x)
                           ^
          SyntaxError: invalid syntax
```

```
In [320]: def add_10(x):
              return x+10
```

```
In [322]: add_10(9)

Out[322]: 19
```

```
In [324]: add_10(5)

Out[324]: 15
```

```
In [325]: def odd_even(x):
              if (x%2==0):
                  print(x,"is even no")
              else:
                  print(x,"is odd no")
```

```
In [327]: odd_even(10)

          10 is even no
```

```
In [329]: odd_even(13)

          13 is odd no
```

```
In [330]: #Lambda function or ananimus function
In [331]: g=lamda x:x*x*x
```

```
  File "<ipython-input-331-9388b5773830>", line 1
    g=lamda x:x*x*x
            ^
SyntaxError: invalid syntax
```

```
In [332]: g=lambda x: x*x*x
```

```
In [334]: g(2)
```

```
Out[334]: 8
```

```
In [335]: #Lambda(x:expresseion) with filter (1st parameter is function and 2nd i
          s list)
```

```
In [336]: l1=[23,45,66,22,54,65,45]
          final_list= list(filter(lambda (x:x%2!=0),l1))
```

```
  File "<ipython-input-336-2bf400d6ca2e>", line 2
    final_list= list(filter(lambda (x:x%2!=0),l1))
                                   ^
SyntaxError: invalid syntax
```

```
In [337]: l1=[23,45,66,22,54,65,45]
          final_list= list(filter(lambda x:(x%2!=0),l1)
```

```
  File "<ipython-input-337-05e261ff13e9>", line 2
    final_list= list(filter(lambda x:(x%2!=0),l1)
                                                 ^
SyntaxError: unexpected EOF while parsing
```

```
In [338]: l1=[23,45,66,22,54,65,45]
          final_list=list(filter(lambda x: (x%2!=0),l1))
```

```
In [340]: final_list
```

```
Out[340]: [23, 45, 65, 45]
```

```
In [342]: l1=[23,45,66,22,54,65,45]
          final_list=list(filter(lambda x: (x%2=0),l1))
```

```
  File "<ipython-input-342-df852cc18b9a>", line 2
    final_list=list(filter(lambda x: (x%2=0),l1))
                                         ^

SyntaxError: invalid syntax
```

```
In [344]: l1=[23,45,66,22,54,65,45]
          final_list=list(filter(lambda x: (x%2!==0),l1))
```

```
  File "<ipython-input-344-0ea3e24214de>", line 2
    final_list=list(filter(lambda x: (x%2!==0),l1))
                                           ^

SyntaxError: invalid syntax
```

```
In [346]: l1=[23,45,66,22,54,65,45]
          final_list=list(filter(lambda x: (x%2==0),l1))
```

```
In [347]: final_list
```

```
Out[347]: [66, 22, 54]
```

```
In [348]: #Lambda function with map. map take 2 para (1st lambda, 2nd list)
```

```
In [349]: l1=[1,2,3,4,5,6,7]
```

```
In [350]: final_result=list(map(lambda x:x*2,l1))
```

```
In [352]: final_result
```

```
Out[352]: [2, 4, 6, 8, 10, 12, 14]
```

```
In [353]: #reduce (import from functool module), reduce take 2 para (lambda,list)
In [355]: l1

Out[355]: [1, 2, 3, 4, 5, 6, 7]

In [357]: from functools import reduce

In [359]: sum=reduce(lambda x,y: x+y, l1)

In [360]: sum

Out[360]: 28

In [361]: #single dim array

In [362]: import numpy as np

In [364]: l1=[1,2,3,6,7]
          np.array(l1)
          l1

Out[364]: [1, 2, 3, 6, 7]

In [365]: l1=[1,2,3,4,6,7]

In [366]: n1=np.array(l1)

In [368]: n1

Out[368]: array([1, 2, 3, 4, 6, 7])

In [370]: type(n1)

Out[370]: numpy.ndarray
```

```
In [372]: n2=np.array([[1,2,3,4],[3,4,6,7]])
          n2

Out[372]: array([[1, 2, 3, 4],
                 [3, 4, 6, 7]])

In [374]: type(n2)

Out[374]: numpy.ndarray

In [376]: #NumPy arrary with single dimensional

In [377]: l1 = [1,2,3,4,5,6]

In [378]: import numpy as np
          l1 = [1,2,3,4,5]

In [379]: n1=np.array(l1) #np.array will create array and assign to n1

In [381]: l1

Out[381]: [1, 2, 3, 4, 5]

In [383]: type(l1)

Out[383]: list

In [385]: type(n1)

Out[385]: numpy.ndarray

In [391]: #Multidim array creation

In [387]: n2 = np.array([[2,4,6,8],[1,3,5,7]])
In [386]: np.array([[2,4,6,8],[1,3,5,7]])

Out[386]: array([[2, 4, 6, 8],
                 [1, 3, 5, 7]])

In [389]: n2

Out[389]: array([[2, 4, 6, 8],
                 [1, 3, 5, 7]])

In [392]: type(n2)

Out[392]: numpy.ndarray

In [393]: #Initializing Numpy array with zero

In [397]: n3=np.zeros((2,3)) # 2 rows and 3 columns

In [399]: n3

Out[399]: array([[0., 0., 0.],
                 [0., 0., 0.]])

In [400]: #multidim

In [402]: type(n3)

Out[402]: numpy.ndarray

In [404]: n4=np.zeros((10,10))
          n4

Out[404]: 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.]])
```

```
In [406]: type(n4)

Out[406]: numpy.ndarray
```

```
In [408]: n5=np.full((4,4),22)
          n5

Out[408]: array([[22, 22, 22, 22],
                 [22, 22, 22, 22],
                 [22, 22, 22, 22],
                 [22, 22, 22, 22]])
```

```
In [410]: n6=np.full((4,4),22,22)
          n6

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-410-9ca4a5bfc42f> in <module>
----> 1 n6=np.full((4,4),22,22)
      2 n6

~\anaconda3\lib\site-packages\numpy\core\numeric.py in full(shape, fill_value, dtype, order)
    323     if dtype is None:
    324         dtype = array(fill_value).dtype
--> 325     a = empty(shape, dtype, order)
    326     multiarray.copyto(a, fill_value, casting='unsafe')
    327     return a

TypeError: data type not understood
```

```
In [412]: na=np.arange(0,20)
          na

Out[412]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15,
          16,
                 17, 18, 19])

In [413]: type(na)

Out[413]: numpy.ndarray

In [421]: nb=np.arange(10,20,5)
          nb

Out[421]: array([10, 15])

In [418]: nb=np.arange(10,100,5)
          nb

Out[418]: array([10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85,
          90,
                 95])

In [419]: import numpy as np

In [422]: np.random.randint(10,100,10)

Out[422]: array([71, 32, 25, 37, 82, 48, 80, 83, 69, 87])

In [423]: import numpy as np

In [426]: n1 = np.array([[1,2,3,4],[5,6,7,8]])
          n1

Out[426]: array([[1, 2, 3, 4],
                 [5, 6, 7, 8]])
```

```
In [429]: n1.shape = (4,2)
```

```
In [431]: n1
```

```
Out[431]: array([[1, 2],
                 [3, 4],
                 [5, 6],
                 [7, 8]])
```

```
In [433]: n1.shape = (7,1)
          n1
```

```
---------------------------------------------------------------------------
----
ValueError                                Traceback (most recent call l
ast)
<ipython-input-433-5b1b37dca403> in <module>
----> 1 n1.shape = (7,1)
      2 n1

ValueError: cannot reshape array of size 8 into shape (7,1)
```

```
In [435]: n1.shape = (8,1)
          n1
```

```
Out[435]: array([[1],
                 [2],
                 [3],
                 [4],
                 [5],
                 [6],
                 [7],
                 [8]])
```

```
In [437]: import numpy as np
```

```
In [438]: n1 = np.array([10,20,30,40])
          n2 = np.array([50,60,70,80])
```

```
n3 =np.array([1,2,3,4])
```

In [444]:
```
np.vstack((n1,n2,n3))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-444-4d73170c28cb> in <module>
----> 1 np.vstack((n1,n2,n3))

<__array_function__ internals> in vstack(*args, **kwargs)

~\anaconda3\lib\site-packages\numpy\core\shape_base.py in vstack(tup)
    281     if not isinstance(arrs, list):
    282         arrs = [arrs]
--> 283     return _nx.concatenate(arrs, 0)
    284
    285

<__array_function__ internals> in concatenate(*args, **kwargs)

ValueError: all the input array dimensions for the concatenation axis must match exactly, but along dimension 1, the array at index 0 has size 4 and the array at index 2 has size 3
```

In [442]:
```
np.hstack((n1,n2))
```

Out[442]:
```
array([10, 20, 30, 40, 50, 60, 70, 80])
```

In [445]:
```
np.column_stack((n1,n2))
```

Out[445]:
```
array([[10, 50],
       [20, 60],
       [30, 70],
       [40, 80]])
```

In [449]:
```
n1 = np.array([10,20,30,40])
```

```
n2 = np.array([50,60,70,80])
n3 = np.array([1,2,3,4])
```

In [451]:

```
np.vstack((n1,n2,n3))
```

Out[451]:

```
array([[10, 20, 30, 40],
       [50, 60, 70, 80],
       [ 1,  2,  3,  4]])
```

In [452]:

```
import numpy as np
```

In [453]:

```
n1= np.array([1,2,3,4])
n2 = np.array([3,4,5,6])
```

In [461]:

```
np.intersect1d(n2,n1) #will give u common elements
```

Out[461]:

```
array([3, 4])
```

In [457]:

```
np.setdiff1d(n1,n2)
```

Out[457]:

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

In [459]:

```
np.setdiff1d(n2,n1)
```

Out[459]:

```
array([5, 6])
```

In [462]:

```
np.setdiff1d(n1,n2)
```

Out[462]:

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

In [463]:

```
import numpy as np
```

In [479]:

```
n1 = np.array([10,50])
n2 = np.array([40,30])
```

In [480]:

```
np.sum([n1,n2])
```

```
Out[480]: 130

In [481]: np.sum([n1,n2], axis=0) #will add elements vertically

Out[481]: array([50, 80])

In [482]: np.sum([n1,n2],axis=1)#will add elements horizontally

Out[482]: array([60, 70])

In [484]: import numpy as np

In [491]: n1=np.array([10,20,30])

In [492]: n1

Out[492]: array([10, 20, 30])

In [493]: type(n1)

Out[493]: numpy.ndarray

In [495]: n1=n1+5

In [496]: n1

Out[496]: array([15, 25, 35])

In [497]: n1= n1-3
          n1

Out[497]: array([12, 22, 32])

  In [ ]:
```