

PYTHON

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

```
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 recieve 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 tha best option to write the code.
a=10
b=5
c=a+b
print(c);
```

15

```
In [189... # if the soluton getting it is ok ,getting error then systom version issue
_ + 3
```

```

-----
TypeError                                Traceback (most recent call last)
Cell In[189], line 2
      1 # if the solution getting it is ok ,getting error then system version issue
-----> 2 _ + 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 == categorical 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                                Traceback (most recent call last)
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...

```
aaaaaaaaaaaaaaaaaaaaa=15
print(aaaaaaaaaaaaaaaaaaaaa)
```

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                                Traceback (most recent call last)
```

```
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                                Traceback (most recent call last)  
Cell In[268], line 1  
----> 1 True/False  
  
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)

STRING

```
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

INTIGER

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

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

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

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

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

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[355], line 1  
----> 1 int(1+2j)  
  
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                                Traceback (most recent call last)  
Cell In[359], line 1  
----> 1 int('ten')  
  
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                                Traceback (most recent call last)  
Cell In[365], line 1  
----> 1 s2  
  
NameError: name 's2' is not defined
```

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

```
-----  
NameError                                Traceback (most recent call last)  
Cell In[367], line 1  
----> 1 np.nan  
  
NameError: name 'np' is not defined
```

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

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

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

FLOAT

```
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'
```

COMPLEX

```
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)

BOLLEAN

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='HELLOPYTHON'`
`index`

Out[425...] 'HELLOPYTHON'

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

Out[427...] 'HELLOPYTHON'

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

Out[429...] 'NOHTYPOLLEH'

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

Out[431...] 'NHYOLH'

In [433...] `index`

Out[433...] 'HELLOPYTHON'

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

Out[435... 'NYL'

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

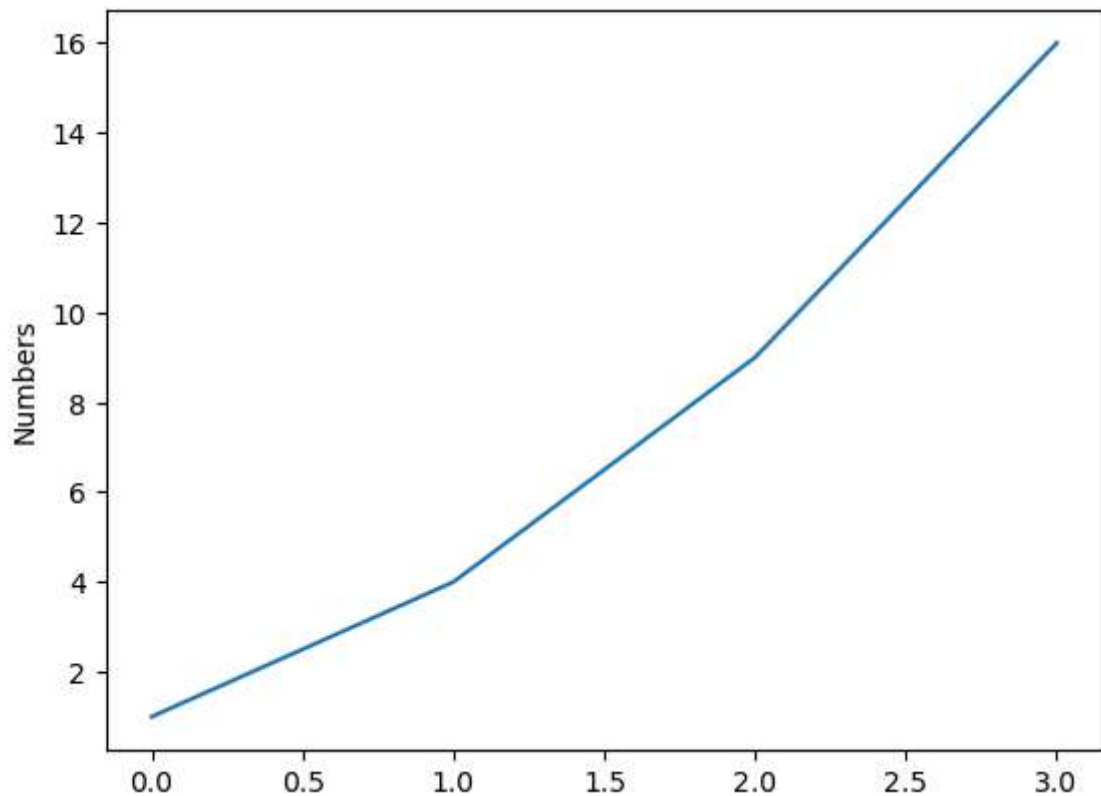
Out[437... 'HELLOPY'

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

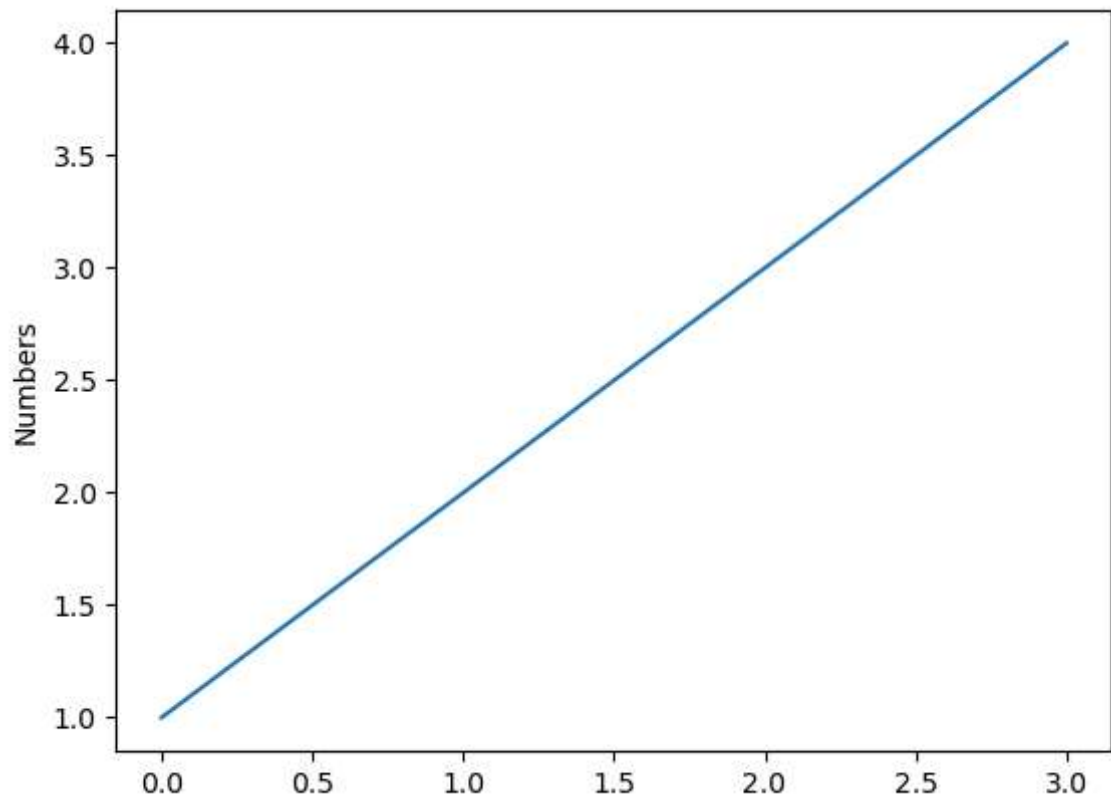
Out[439... 'EOT'

python type casting completed

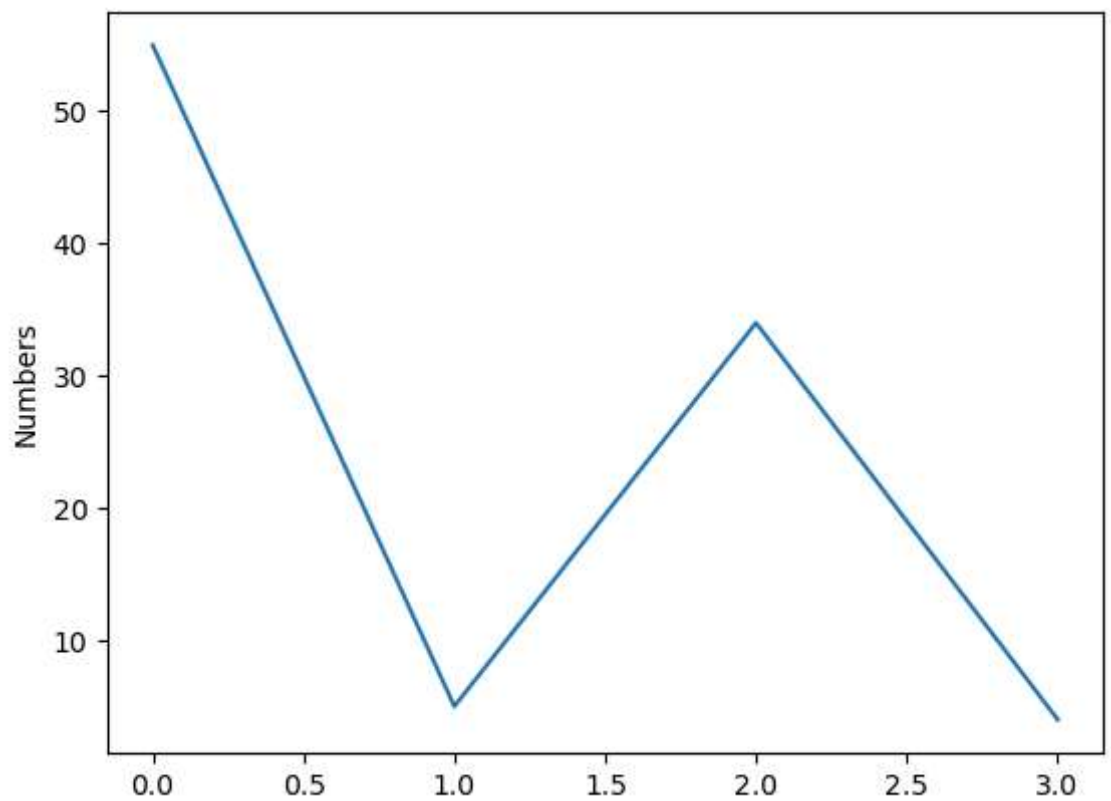
```
In [442... import matplotlib.pyplot as plt
plt.plot([1,4,9,16])
plt.ylabel('Numbers')
plt.show()
```



```
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
```

DATA STRUCTURE

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

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

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

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

```
In [469...  l
```

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

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

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

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

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

In [509...

```
s
```

Out[509...

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

In [511...

```
s==l1
```

Out[511...

```
False
```

In [513...

```
l1
```

Out[513...

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

In [515...

```
print(l1)  
print(s)
```

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

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

In [517...

```
s[4]
```

Out[517...

```
True
```

In []:

COMPLETE DATATYPE