1- NUMBERS

```
In [1]: 10
```

```
Out[1]: 10
```

```
In [2]: 5
```

```
Out[2]: 5
```

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

```
Out[3]: 15
```

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

```
Out[4]: 5
```

```
In [5]: 10 / 5  #FLOAT DIVISION
```

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

```
In [6]: 10 // 5 #INT DIVISION
```

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

2- TEXT (STRING)

```
In [7]: 'welcome to 7pm fsdsbatch under prakash senapati guidance'
```

```
Out[7]: 'welcome to 7pm fsdsbatch under prakash senapati guidance'
```

```
In [8]: "welcome to 7pm fsdsbatch under prakash senapati guidance"
```

```
Out[8]: 'welcome to 7pm fsdsbatch under prakash senapati guidance'
```

```
In [9]: ''' welcome to 7pm fsdsbatch
        under prakash senapati guidance "
```

```
  Cell In[9], line 1
    ''' welcome to 7pm fsdsbatch
    ^
_IncompleteInputError: incomplete input
```

```
In [10]: " welcome to 7pm fsdsbatch
         under prakash senapati guidance '''
```

```
  Cell In[10], line 1
    " welcome to 7pm fsdsbatch
    ^
SyntaxError: unterminated string literal (detected at line 1)
```

```
In [11]: ''' welcome to 7pm fsdsbatch
         under prakash senapati guidance '''
```

```
Out[11]: ' welcome to 7pm fsdsbatch \nunder prakash senapati guidance '
```

3- python variable creation

- variable name = value

```
In [12]: v = 9
```

```
In [13]: v
```

```
Out[13]: 9
```

```
In [14]: id(v)
```

```
Out[14]: 140718199780520
```

```
In [15]: 9 = v
```

```
  Cell In[15], line 1
    9 = v
    ^
SyntaxError: cannot assign to literal here. Maybe you meant '==' instead of '='?
```

```
In [16]: 9v = 8
```

```
  Cell In[16], line 1
    9v = 8
    ^
SyntaxError: invalid decimal literal
```

```
In [17]: v9 = 8
         v9
```

```
Out[17]: 8
```

```
In [18]: var = 10
```

```
In [19]: VAR
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[19], line 1
----> 1 VAR

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

```
In [20]: var
```

```
Out[20]: 10
```

```
In [21]: nit = 7
         narehsit
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[21], line 2
      1 nit = 7
----> 2 narehsit

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

```
In [22]: puspa1, puspa2 = 2000
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[22], line 1
----> 1 puspa1, puspa2 = 2000

TypeError: cannot unpack non-iterable int object
```

```
In [23]: puspa1, puspa2 = 2000, 1800
```

```
In [24]: puspa1
         puspa2
```

```
Out[24]: 1800
```

```
In [25]: print(puspa1)
         print(puspa2)
```

```
2000
1800
```

```
In [26]: nit@ = 7
```

```
  Cell In[26], line 1
    nit@ = 7
         ^
SyntaxError: invalid syntax
```

```
In [27]: nit$
```

```
  Cell In[27], line 1
    nit$
       ^
SyntaxError: invalid syntax
```

In [28]:

```
nit_ = 7
nit_
```

Out[28]:

```
7
```

In [29]:

```
if = 78
if
```

```
  Cell In[29], line 1
    if = 78
       ^
SyntaxError: invalid syntax
```

In [30]:

```
import keyword
keyword.kwlist
```

```
Out[30]: ['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 [31]: False = 90
```

```
  Cell In[31], line 1
    False = 90
    ^
SyntaxError: cannot assign to False
```

```
In [32]: len(keyword.kwlist)
```

```
Out[32]: 35
```

28th -- python data type

```
In [33]: i = 45
         i
```

```
Out[33]: 45
```

```
In [34]: type(i)

Out[34]: int

In [35]: f = 110.45
         f

Out[35]: 110.45

In [36]: type(f)

Out[36]: float

In [37]: f_gold_price = 99999.90

In [38]: f_gold_price

Out[38]: 99999.9

In [39]: f1 = 1e0
         f1

Out[39]: 1.0

In [40]: type(f1)

Out[40]: float

In [41]: f2 = 2e1
         f2

Out[41]: 20.0

In [42]: f3 = 2.4e2
         f3

Out[42]: 240.0

In [43]: f4 = 2.5E3
         f4

Out[43]: 2500.0

In [44]: f5 = 2.5z1
         f5

  Cell In[44], line 1
    f5 = 2.5z1
           ^
SyntaxError: invalid decimal literal

In [45]: true
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[45], line 1
----> 1 true

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

In [46]:

```
import keyword
keyword.kwlist
```

Out[46]:

```
['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 [47]:

```
TRUE
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[47], line 1
----> 1 TRUE

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

In [48]:

```
True
```

```
Out[48]:  True

In [49]:  false

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[49], line 1
----> 1 false

NameError: name 'false' is not defined

In [50]:  False

Out[50]:  False

In [51]:  True + False

Out[51]:  1

In [52]:  True + True

Out[52]:  2

In [53]:  True * fALSE

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[53], line 1
----> 1 True * fALSE

NameError: name 'fALSE' is not defined

In [54]:  True * False

Out[54]:  0

In [55]:  False / True

Out[55]:  0.0

In [56]:  False // True

Out[56]:  0

In [57]:  True / False

---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[57], line 1
----> 1 True / False

ZeroDivisionError: division by zero

In [58]:  True
```

```
Out[58]: True

In [59]: int(True)

Out[59]: 1

In [60]: int(False)

Out[60]: 0
```

29th

int, float, bool

```
In [61]: c = 1 + 2j
         c

Out[61]: (1+2j)

In [62]: type(c)

Out[62]: complex

In [63]: c.real

Out[63]: 1.0

In [64]: c.imag

Out[64]: 2.0

In [65]: c = 10 + 20j
         d = 30 + 40j

In [66]: e = c + d
         e

Out[66]: (40+60j)

In [67]: s = 'nareshit'

In [68]: s

Out[68]: 'nareshit'

In [69]: s[0]

Out[69]: 'n'
```

```
In [70]: s[1]
```

```
Out[70]: 'a'
```

```
In [71]: s
```

```
Out[71]: 'nareshit'
```

```
In [72]: s1 = 'technology'
         s2 = 'under guidance of prakash senapati'
```

```
In [73]: s
         s1
         s2
```

```
Out[73]: 'under guidance of prakash senapati'
```

```
In [74]: print(s)
         print(s1)
         print(s2)
```

```
nareshit
technology
under guidance of prakash senapati
```

```
In [75]: s + s1 + s2
```

```
Out[75]: 'nareshittechnologyunder guidance of prakash senapati'
```

```
In [76]: s +  s1
```

```
Out[76]: 'nareshittechnology'
```

```
In [77]: s * s1
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[77], line 1
----> 1 s * s1

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

```
In [78]:  print(10)
```

```
10
```

```
In [79]: print(10, 2.4,'nit',True,1+2j,)
```

```
10 2.4 nit True (1+2j)
```

```
In [80]: num1=20
         num2=30
         add=num1+num2
         add
```

Out[80]:

```
50
```

In [81]:

```
print('The addition of',num1,'and',num2,'is=',add)
```

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

print Format method

In [82]:

```
num1=20
num2=30
add=num1+num2

print('The addition of {} and {} is= {}'.format(num1,num2,add))
```

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

In [83]:

```
num1=20
num2=30
num3=40
add=num1+num2+num3

print('The addition of {} and {} and {} is= {}'.format(num1,num2,num3,add))
```

```
The addition of 20 and 30 and 40 is= 90
```

In [84]:

```
print('hello') # 1st statement
print('good moorning') # 2nd statement)

# i want print like:- hellow good morning
```

```
hello
good moorning
```

In [85]:

```
print('hello', end=' ') # 1st statement
print('good night') # 2nd statement
```

```
hello good night
```

In [86]:

```
print('hello','hai','how are you',sep='--->')
```

```
hello--->hai--->how are you
```

In [87]:

```
print('hello','hai','how are you',sep='***&&&&--->')
```

```
hello***&&&&--->hai***&&&&--->how are you
```

31st python type casting

In [88]:

```
int(2.3)
```

Out[88]:

```
2
```

In [89]:

```
int(2.3, 3.4)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[89], line 1
----> 1 int(2.3, 3.4)

TypeError: 'float' object cannot be interpreted as an integer
```

```
In [90]: int(True)
```

```
Out[90]: 1
```

```
In [91]: int(False)
```

```
Out[91]: 0
```

```
In [92]: int(1+2j)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[92], line 1
----> 1 int(1+2j)

TypeError: int() argument must be a string, a bytes-like object or a real number, no
t 'complex'
```

```
In [93]: int('10')
```

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

```
In [94]: int('ten')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[94], line 1
----> 1 int('ten')

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

```
In [95]: float(10)
```

```
Out[95]: 10.0
```

```
In [96]: float(10, 20)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[96], line 1
----> 1 float(10, 20)

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

```
In [97]: float(True)
```

```
Out[97]: 1.0
```

In [98]:
```
float(False)
```

Out[98]:
```
0.0
```

In [99]:
```
float(1+2j)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[99], line 1
----> 1 float(1+2j)

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

In [100…
```
float('10')
```

Out[100…
```
10.0
```

In [101…
```
float('ten')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[101], line 1
----> 1 float('ten')

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

In [102…
```
complex(10)
```

Out[102…
```
(10+0j)
```

In [103…
```
complex(10,20)
```

Out[103…
```
(10+20j)
```

In [104…
```
complex(10,20,30)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[104], line 1
----> 1 complex(10,20,30)

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

In [105…
```
complex(2.3)
```

Out[105…
```
(2.3+0j)
```

In [106…
```
complex(2.3, 3.4)
```

Out[106…
```
(2.3+3.4j)
```

In [107…
```
complex(10, '10')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[107], line 1
----> 1 complex(10, '10')

TypeError: complex() second arg can't be a string
```

In [108…

```
complex('10', 10)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[108], line 1
----> 1 complex('10', 10)

TypeError: complex() can't take second arg if first is a string
```

In [109…

```
complex('10')
```

Out[109…

```
(10+0j)
```

In [110…

```
complex(10,int('10'))
```

Out[110…

```
(10+10j)
```

In [111…

```
complex(True, False)
```

Out[111…

```
(1+0j)
```

In [112…

```
complex(False,False)
```

Out[112…

```
0j
```

In [113…

```
bool(1)
```

Out[113…

```
True
```

In [287…

```
bool(10)
```

Out[287…

```
True
```

In [114…

```
bool(13)
```

Out[114…

```
True
```

In [115…

```
bool()
```

Out[115…

```
False
```

In [116…

```
bool( )
```

Out[116…

```
False
```

In [117…

```
bool(1.2)
```

Out[117… True

In [118…

```
bool('10')
```

Out[118… True

In [119…

```
bool('ten')
```

Out[119… True

In [120…

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

Out[120… True

In [121…

```
bool(0+0j)
```

Out[121… False

In [122…

```
print(str(1))
print(str(1.2))
print(str(True))
print(str(1+2j))
```

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

string indexing

```
- forward indexing
- backward indexing
- step indexinbg
```

In [123…

```
s = 'hello'
s
```

Out[123… 'hello'

In [124…

```
s[0]
```

Out[124… 'h'

In [125…

```
s[-1]
```

Out[125… 'o'

In [126…

```
s
```

Out[126… 'hello'

In [127…

```
s[10]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[127], line 1
----> 1 s[10]

IndexError: string index out of range
```

In [128…

```
s
```

Out[128…

```
'hello'
```

In [129…

```
s[:]
```

Out[129…

```
'hello'
```

In [130…

```
s[0:1]
```

Out[130…

```
'h'
```

In [131…

```
s[0:2]
```

Out[131…

```
'he'
```

In [132…

```
s
```

Out[132…

```
'hello'
```

In [133…

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

```
h
e
l
l
o
```

In [134…

```
s
```

Out[134…

```
'hello'
```

In [135…

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

```
h
e
l
l
o
```

In [136…
```
s
```
Out[136…
```
'hello'
```
In [137…
```
s1 = 'nareshit'
```
In [138…
```
s + s1
```
Out[138…
```
'hellonareshit'
```
In [139…
```
s3 = s + s1
s3
```
Out[139…
```
'hellonareshit'
```
In [140…
```
s
```
Out[140…
```
'hello'
```
In [141…
```
s[2:5]
```
Out[141…
```
'llo'
```
In [142…
```
s
```
Out[142…
```
'hello'
```
In [143…
```
s2 = ['r','g','y']
s2
```
Out[143…
```
['r', 'g', 'y']
```
In [144…
```
s2[1:4]
```
Out[144…
```
['g', 'y']
```

type casting we are completed

1st -- PYTHON DATASTRUCTURE

In [145…
```
l = []
l
```
Out[145…
```
[]
```
In [146…
```
type(l)
```
Out[146…
```
list
```

In [147…
```
len(l)
```
Out[147…
```
0
```
In [148…
```
id(l)
```
Out[148…
```
2446912420032
```
In [149…
```
l.append()
```
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[149], line 1
----> 1 l.append()

TypeError: list.append() takes exactly one argument (0 given)
```
In [150…
```
l.append(10)
```
In [151…
```
l
```
Out[151…
```
[10]
```
In [152…
```
len(l)
```
Out[152…
```
1
```
In [153…
```
l.append(20)
l.append(30)
l.append(40)
l.append(50)
```
In [154…
```
l
```
Out[154…
```
[10, 20, 30, 40, 50]
```
In [155…
```
len(l)
```
Out[155…
```
5
```
In [156…
```
l
```
Out[156…
```
[10, 20, 30, 40, 50]
```
In [157…
```
l1 = l.copy()
l1
```
Out[157…
```
[10, 20, 30, 40, 50]
```
In [158…
```
l == l1
```
Out[158…
```
True
```

```
In [159…  l != l1

Out[159…  False

In [160…  l1

Out[160…  [10, 20, 30, 40, 50]

In [161…  l1.append(100)

In [162…  l1

Out[162…  [10, 20, 30, 40, 50, 100]

In [163…  l == l1

Out[163…  False

In [164…  l != l1

Out[164…  True

In [165…  print(len(l))
          print(len(l1))

         5
         6

In [166…  print(l)
          print(l1)

         [10, 20, 30, 40, 50]
         [10, 20, 30, 40, 50, 100]

In [167…  id(l1)

Out[167…  2446913339968

In [168…  l1.clear()

In [169…  l1

Out[169…  []

In [170…  id(l1)

Out[170…  2446913339968

In [171…  l

Out[171…  [10, 20, 30, 40, 50]
```

In [172…

```
l.append('nit')
l.append(2.3)
l.append(1+2j)
l.append(True)
l.append([1,2,3])
```

In [173…

```
l
```

Out[173…

```
[10, 20, 30, 40, 50, 'nit', 2.3, (1+2j), True, [1, 2, 3]]
```

In [174…

```
l.append(10)
l
```

Out[174…

```
[10, 20, 30, 40, 50, 'nit', 2.3, (1+2j), True, [1, 2, 3], 10]
```

In [175…

```
print(l)
```

```
[10, 20, 30, 40, 50, 'nit', 2.3, (1+2j), True, [1, 2, 3], 10]
```

In [176…

```
l1=[]
l1
```

Out[176…

```
[]
```

In [177…

```
l1 = [10, 2.3, 1+2j, True, 'hello', [1,2,3]]
l1
```

Out[177…

```
[10, 2.3, (1+2j), True, 'hello', [1, 2, 3]]
```

In [178…

```
l1.append(10)
```

In [179…

```
l1
```

Out[179…

```
[10, 2.3, (1+2j), True, 'hello', [1, 2, 3], 10]
```

In [180…

```
l1.count(2.3)
```

Out[180…

```
1
```

In [181…

```
l1
```

Out[181…

```
[10, 2.3, (1+2j), True, 'hello', [1, 2, 3], 10]
```

In [182…

```
l1.remove(1+2j)
```

In [183…

```
l1
```

Out[183…

```
[10, 2.3, True, 'hello', [1, 2, 3], 10]
```

In [184…

```
l1
```

Out[184…

```
[10, 2.3, True, 'hello', [1, 2, 3], 10]
```

```
In [185…  l1.pop()

Out[185…  10

In [186…  l1

Out[186…  [10, 2.3, True, 'hello', [1, 2, 3]]

In [187…  l1.pop()

Out[187…  [1, 2, 3]

In [188…  l1

Out[188…  [10, 2.3, True, 'hello']

In [189…  l1.remove(True)

In [190…  l1

Out[190…  [10, 2.3, 'hello']

In [191…  l

Out[191…  [10, 20, 30, 40, 50, 'nit', 2.3, (1+2j), True, [1, 2, 3], 10]

In [192…  l.pop(2)

Out[192…  30

In [193…  l

Out[193…  [10, 20, 40, 50, 'nit', 2.3, (1+2j), True, [1, 2, 3], 10]

In [194…  l

Out[194…  [10, 20, 40, 50, 'nit', 2.3, (1+2j), True, [1, 2, 3], 10]

In [195…  l1

Out[195…  [10, 2.3, 'hello']

In [196…  l

Out[196…  [10, 20, 40, 50, 'nit', 2.3, (1+2j), True, [1, 2, 3], 10]

In [197…  l.index(2.3)

Out[197…  5

In [198…  l
```

```
Out[198…  [10, 20, 40, 50, 'nit', 2.3, (1+2j), True, [1, 2, 3], 10]
```

```
In [199…  l.insert(5,4)
          l
```

```
Out[199…  [10, 20, 40, 50, 'nit', 4, 2.3, (1+2j), True, [1, 2, 3], 10]
```

```
In [200…  l
```

```
Out[200…  [10, 20, 40, 50, 'nit', 4, 2.3, (1+2j), True, [1, 2, 3], 10]
```

```
In [201…  l.insert(4, 5)
```

```
In [202…  l
```

```
Out[202…  [10, 20, 40, 50, 5, 'nit', 4, 2.3, (1+2j), True, [1, 2, 3], 10]
```

```
In [203…  l
```

```
Out[203…  [10, 20, 40, 50, 5, 'nit', 4, 2.3, (1+2j), True, [1, 2, 3], 10]
```

```
In [204…  l.insert(6, 6)
```

```
In [205…  l
```

```
Out[205…  [10, 20, 40, 50, 5, 'nit', 6, 4, 2.3, (1+2j), True, [1, 2, 3], 10]
```

```
In [206…  print(l1)
          print(l)
```

```
[10, 2.3, 'hello']
[10, 20, 40, 50, 5, 'nit', 6, 4, 2.3, (1+2j), True, [1, 2, 3], 10]
```

```
In [207…  l1.extend(l)
```

```
In [208…  l1
```

```
Out[208…  [10,
           2.3,
           'hello',
           10,
           20,
           40,
           50,
           5,
           'nit',
           6,
           4,
           2.3,
           (1+2j),
           True,
           [1, 2, 3],
           10]
```

```
In [209…  l
Out[209…  [10, 20, 40, 50, 5, 'nit', 6, 4, 2.3, (1+2j), True, [1, 2, 3], 10]

In [210…  l.reverse()

In [211…  l
Out[211…  [10, [1, 2, 3], True, (1+2j), 2.3, 4, 6, 'nit', 5, 50, 40, 20, 10]

In [212…  l5 = [300, 3, 34, 9, 100]
          l5
Out[212…  [300, 3, 34, 9, 100]

In [213…  l5.sort()

In [214…  l5
Out[214…  [3, 9, 34, 100, 300]

In [215…  l5.sort(reverse=True)

In [216…  l5
Out[216…  [300, 100, 34, 9, 3]

In [217…  l
Out[217…  [10, [1, 2, 3], True, (1+2j), 2.3, 4, 6, 'nit', 5, 50, 40, 20, 10]

In [218…  l5
Out[218…  [300, 100, 34, 9, 3]

In [219…  l5[0] = 3000

In [220…  l5
Out[220…  [3000, 100, 34, 9, 3]
```

list is completed

```
In [221…  l
Out[221…  [10, [1, 2, 3], True, (1+2j), 2.3, 4, 6, 'nit', 5, 50, 40, 20, 10]

In [222…  l6= l.copy()
```

In [223…

```
l6
```

Out[223…

```
[10, [1, 2, 3], True, (1+2j), 2.3, 4, 6, 'nit', 5, 50, 40, 20, 10]
```

4th -- TUPLE() & RANGE()

In [224…

```
t = ()
t
```

Out[224…

```
()
```

In [225…

```
type(t)
```

Out[225…

```
tuple
```

In [226…

```
t = (10,20,30)
t
```

Out[226…

```
(10, 20, 30)
```

In [227…

```
len(t)
```

Out[227…

```
3
```

In [228…

```
t1 = (10, 2.3, 'nit', 1+2j, True, 10, 20)
t1
```

Out[228…

```
(10, 2.3, 'nit', (1+2j), True, 10, 20)
```

In [229…

```
print(t)
print(t1)
```

```
(10, 20, 30)
(10, 2.3, 'nit', (1+2j), True, 10, 20)
```

In [230…

```
t.count(10)
```

Out[230…

```
1
```

In [231…

```
t1.count(2.3)
```

Out[231…

```
1
```

In [232…

```
t1
```

Out[232…

```
(10, 2.3, 'nit', (1+2j), True, 10, 20)
```

In [233…

```
t1.index(2.3)
```

Out[233…

```
1
```

```
In [234…  t

Out[234…  (10, 20, 30)

In [235…  t[0]

Out[235…  10

In [236…  t[0] = 100

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[236], line 1
----> 1 t[0] = 100

TypeError: 'tuple' object does not support item assignment

In [237…  icici = (123466, 'cizps67898y', 'dob-3rd mar 1987', 'mob:1234')
          icici

Out[237…  (123466, 'cizps67898y', 'dob-3rd mar 1987', 'mob:1234')

In [238…  icici[0] = 987645

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[238], line 1
----> 1 icici[0] = 987645

TypeError: 'tuple' object does not support item assignment

In [239…  t

Out[239…  (10, 20, 30)

In [240…  t2 = t * 3
          t2

Out[240…  (10, 20, 30, 10, 20, 30, 10, 20, 30)

In [241…  t2

Out[241…  (10, 20, 30, 10, 20, 30, 10, 20, 30)

In [242…  t1

Out[242…  (10, 2.3, 'nit', (1+2j), True, 10, 20)

In [243…  t1[1:6]

Out[243…  (2.3, 'nit', (1+2j), True, 10)

In [244…  t2
```

```
Out[244…  (10, 20, 30, 10, 20, 30, 10, 20, 30)

In [245…  t2[1:6:3]

Out[245…  (20, 20)

In [246…  t2

Out[246…  (10, 20, 30, 10, 20, 30, 10, 20, 30)

In [247…  t2[1:6:5]

Out[247…  (20,)

In [248…  t2

Out[248…  (10, 20, 30, 10, 20, 30, 10, 20, 30)

In [249…  t1

Out[249…  (10, 2.3, 'nit', (1+2j), True, 10, 20)

In [250…  t1[:]

Out[250…  (10, 2.3, 'nit', (1+2j), True, 10, 20)

In [251…  t1

Out[251…  (10, 2.3, 'nit', (1+2j), True, 10, 20)

In [252…  t1[::-1]

Out[252…  (20, 10, True, (1+2j), 'nit', 2.3, 10)

In [253…  t2

Out[253…  (10, 20, 30, 10, 20, 30, 10, 20, 30)

In [254…  t2[::-1]

Out[254…  (30, 20, 10, 30, 20, 10, 30, 20, 10)

In [255…  t2

Out[255…  (10, 20, 30, 10, 20, 30, 10, 20, 30)

In [256…  t2[::-2]

Out[256…  (30, 10, 20, 30, 10)

In [257…  t2
```

```
Out[257…  (10, 20, 30, 10, 20, 30, 10, 20, 30)

In [258…  t2[::-3]

Out[258…  (30, 30, 30)

In [259…  t2

Out[259…  (10, 20, 30, 10, 20, 30, 10, 20, 30)

In [260…  t2[::2]

Out[260…  (10, 30, 20, 10, 30)

In [261…  t = (1,2,3,4,5,6,7,8)
          t

Out[261…  (1, 2, 3, 4, 5, 6, 7, 8)

In [262…  t[3:-1]

Out[262…  (4, 5, 6, 7)

In [263…  t

Out[263…  (1, 2, 3, 4, 5, 6, 7, 8)

In [264…  t[1:4]

Out[264…  (2, 3, 4)

In [265…  t

Out[265…  (1, 2, 3, 4, 5, 6, 7, 8)

In [266…  t [1:]

Out[266…  (2, 3, 4, 5, 6, 7, 8)

In [267…  t

Out[267…  (1, 2, 3, 4, 5, 6, 7, 8)

In [268…  t[:5]

Out[268…  (1, 2, 3, 4, 5)

In [269…  t

Out[269…  (1, 2, 3, 4, 5, 6, 7, 8)
```

In [270…

```
t[1:7:3]
```

Out[270…

```
(2, 5)
```

In [271…

```
t
```

Out[271…

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

In [272…

```
t[::-1]
```

Out[272…

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

In [273…

```
t[::1]
```

Out[273…

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

In [274…

```
t
```

Out[274…

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

In [275…

```
t[::-3]
```

Out[275…

```
(8, 5, 2)
```

In [276…

```
t
```

Out[276…

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

In [277…

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

```
1
2
3
4
5
6
7
8
```

In [288…

```
t1
```

Out[288…

```
(10, 2.3, 'nit', (1+2j), True, 10, 20)
```

In [278…

```
for i in enumerate(t1):
    print(i)
```

```
(0, 10)
(1, 2.3)
(2, 'nit')
(3, (1+2j))
(4, True)
(5, 10)
(6, 20)
```

tupel we are completed

```
In [279…  range()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[279], line 1
----> 1 range()

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

```
In [280…  range(5)

Out[280…  range(0, 5)

In [281…  list(range(5))

Out[281…  [0, 1, 2, 3, 4]

In [282…  range(10,20)

Out[282…  range(10, 20)

In [283…  list(range(10,20))

Out[283…  [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]

In [284…  range(10,50,5)

Out[284…  range(10, 50, 5)

In [285…  list(range(10,50,5))

Out[285…  [10, 15, 20, 25, 30, 35, 40, 45]

In [286…  range(10,50,5,1)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[286], line 1
----> 1 range(10,50,5,1)

TypeError: range expected at most 3 arguments, got 4
```

LIST, TUPLE, RANGE -- We are completed

In [ ]:

In [ ]:

In [ ]:

In [ ]: