8:30 AM -- BASIC PYTHON PROGRAMMING TODAY

In [ ]:
```
import sys
sys.version
```

work with numbers

In [ ]:
```
3
```

In [ ]:
```
2
```

In [ ]:
```
3 + 2
```

In [ ]:
```
3 - 2
```

In [ ]:
```
3 * 2
```

In [ ]:
```
3 ** 2
```

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

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

In [ ]:
```
# work with string
```

In [ ]:
```
nareshit
```

In [ ]:
```
'nareshit'
```

In [ ]:
```
" nareshit "
```

In [ ]:
```
''' naresh it '''
```

variable = object

In [ ]:
```
v = 5 #v - variable & 5 for value
v
```

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

In [ ]:
```
v1 = 'nit'
v2
```

In [ ]:
```
v1
```

26 Th

In [ ]:

```
a = 5.5
type(a)
```

In [ ]:

```
import sys
syst.version
```

In [ ]:

```
nit = 15
NIT
```

In [ ]:

```
nit
```

In [ ]:

```
1a = 67
1a
```

In [ ]:

```
a1 = 67
a1
```

In [ ]:

```
nit$ = 89
nit$
```

In [ ]:

```
x_train, x_test, y_train, y_test = 80, 20, 70, 30
```

In [ ]:

```
x_train
x_test
y_train
y_test
```

In [ ]:

```
print(x_train)
print(x_test)
print(y_train)
print(y_test)
```

In [ ]:

```
import keyword
keyword.kwlist
```

In [ ]:

```
if = 90
if
```

In [ ]:

```
a10 = 78
a9 = 89
```

In [ ]:

```
print(a10)
print(a9)
```

In [ ]:

```
del a10
```

In [ ]:

```
a10
```

In [ ]:

```
for = 90
```

```
In [ ]: For = 90
        For
```

```
In [ ]: a = True
        a
```

```
In [ ]: b = 'true'
        b
```

```
In [ ]: pi = 3.17
        pi
```

```
In [ ]: pi = 3.20
        pi
```

```
In [ ]: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa = 90
        aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
In [ ]: a# = 100
```

Variable are completed

27th -- DATA TYPES

INT FLOAT BOOLEAN COMPLEX STRING

```
In [ ]: i = 25 #value without decimal
        i
```

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

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

```
In [ ]: petrol = 109.50 #value with decimal
        petrol
```

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

```
In [ ]: b = true
        b
```

```
In [ ]: b = True
        b
```

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

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

In [ ]:
```
True -  True
```

In [ ]:
```
True * False
```

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

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

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

In [ ]:
```
c1 = 10 + 20j
c1
```

In [ ]:
```
type(c1)
```

In [ ]:
```
c1.real
```

In [ ]:
```
c1.imaginary
```

In [ ]:
```
c1.imag
```

In [ ]:
```
c1
```

In [ ]:
```
c2 = 20 + 30j
```

In [ ]:
```
print(c1)
print(c2)
```

In [ ]:
```
c1 + c2
```

In [ ]:
```
c1 - c2
```

In [ ]:
```
c2 - c1
```

In [ ]:
```
print(c1)
print(c2)
```

In [ ]:
```
c3 = 20+ 15i
```

In [ ]:
```
c1 * c2
```

In [ ]:
```
s = 'nareshit'
s
```

In [ ]:
```
s1 = "naresh it"
s1
```

In [ ]:
```
s2 = '''naresh
      it'''
s2
```

In [ ]:
```
s
```

string slicing[:]

```
In [ ]: s
```

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

```
In [ ]: s[4] # forward indexin
```

```
In [ ]: s
```

```
In [ ]: s[-4] #backward indxing
```

```
In [ ]: b
```

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

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

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

```
In [ ]: True
```

```
In [ ]: s
```

```
In [ ]: s[1:7]
```

```
In [ ]: s
```

```
In [ ]: s[10]
```

```
In [ ]: s
```

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

python data types are completed

type casting

```
In [ ]: int(2.3) #cast from float to int
```

```
In [ ]: int(2.3, 3.0)
```

```
In [ ]: int(True) #cast from bool to int
```

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

```
In [ ]: True
In [ ]: True + True

In [ ]: int(1+2j)

In [ ]: int('10')

In [ ]: int('ten')

In [ ]: float(10)

In [ ]: float(10, 20)

In [ ]: float(True)

In [ ]: float(False)

In [ ]: float(1+2J)

In [ ]: float('10')

In [ ]: float('ten')

In [ ]: True

In [ ]: True + True

In [ ]: complex(10)

In [ ]: complex(10, 20)

In [ ]: complex(10,20,30,40,50)

In [ ]: complex(2.3)

In [ ]: complex(2.3, 4)

In [ ]: complex(True, True)

In [ ]: complex(False)

In [ ]: complex('10')
```

1st march

```
In [ ]: complex('10', '20')

In [ ]: complex(10, '20')

In [ ]: bool(2)
In [ ]: bool(0)
```

```
In [ ]: bool(2, 5)
```

```
In [ ]: bool(3.2)
```

```
In [ ]: bool(1 + 2j)
```

```
In [ ]: bool(0+0j)
```

```
In [ ]: bool('hi')
```

```
In [ ]: bool( )
```

```
In [ ]: bool(*)
```

```
In [ ]: str(7)
```

```
In [ ]: str(3.4)
```

```
In [ ]: str(1+2j)
```

```
In [ ]: str(TRUE)
```

```
In [ ]: str(True)
```

Type casting we are completed

python operator

```
    - arithmetic operator ( +, -, *, /, //, **)
    -
```

```
In [ ]: x1, y1 = 10, 5
```

```
In [ ]: x1 + y1
```

```
In [ ]: x1 - y1
```

```
In [ ]: x1 / y1
```

```
In [ ]: x1 // y1
```

```
In [ ]: x1 ** y1
```

assignment operator

```
In [ ]: x = 2
        x
```

```
In [ ]: x = x + 2
        x
```

```
In [ ]: x += 2
        x
```

```
In [ ]: x += 4
```

```
In [ ]: x
```

```
In [ ]: x -= 2
        x
```

```
In [ ]: x *= 3
        x
```

```
In [ ]: x /= 2
        x
```

```
In [ ]: x //= 2
        x
```

unary operator

```
In [ ]: n = 7 #negattion
        n
```

```
In [ ]: m = -(n)
        m
```

```
In [ ]: n
```

```
In [ ]: -n
```

Realtional operator

```
In [ ]: r1 = 5
        r2 = 6
```

```
In [ ]: r1 > r2
```

```
In [ ]: r1 < r2
```

```
In [ ]: r1 == r2
```

```
In [ ]: r1 != r2
```

In [ ]:
```
r1
```

In [ ]:
```
r2
```

In [ ]:
```
r3 = 6
```

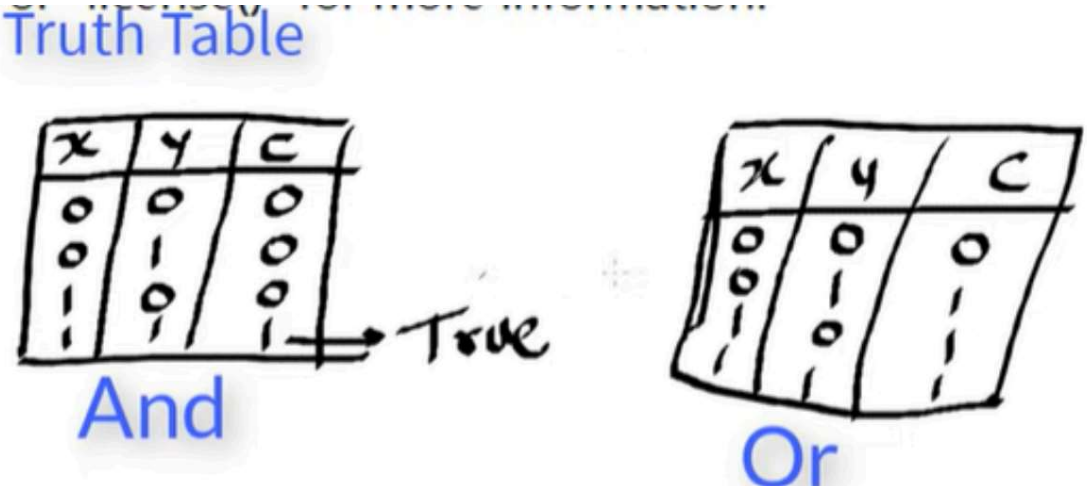
In [ ]:
```
r1 == r3
```

In [ ]:
```
r2 == r3
```

In [ ]:
```
print(r1)
print(r2)
print(r3)
```

In [ ]:
```
r3 >= r2
```

In [ ]:
```
r2 <= r3
```



Truth Table

x	y	c
0	0	0
0	1	0
1	0	0
1	1	1

And

x	y	c
0	0	0
0	1	1
1	0	1
1	1	1

Or

In [ ]:
```
a = 5
b = 4
```

In [ ]:
```
a < 8 and b < 5
```

In [ ]:
```
a<8 or b<5
```

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

In [ ]:
```
b>5 or a<10
```

In [ ]:
```
x = False
x
```

In [ ]:
```
not x
```

In [ ]:
```
y = True
y
```

not y

python operator

3rd march

Datastruture - user will define the value more then one

- list
- tuple
- set
- **dict**

```
In [ ]: l = []
        l
```

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

```
In [ ]: l.append(10)
```

```
In [ ]: l
```

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

```
In [ ]: l
```

```
In [ ]: l.append(20)
        l.append(30)
        l.append(40)
        l.append(40)
```

```
In [ ]: l
```

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

```
In [ ]: l
```

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

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

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

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

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

In [ ]:
```
l
```

In [ ]:
```
l[:]
```

In [ ]:
```
l[0]
```

In [ ]:
```
l[1]
```

In [ ]:
```
l[-3]
```

In [ ]:
```
l
```

In [ ]:
```
l1 = l.copy()
l1
```

In [ ]:
```
l == l1
```

In [ ]:
```
print(len(l))
print(len(l1))
```

In [ ]:
```
l1
```

In [ ]:
```
l1.append(2.3)
l1.append(True)
l1.append(1+2j)
```

In [ ]:
```
l1
```

In [ ]:
```
l1.append(50)
l1
```

In [ ]:
```
l
```

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

In [ ]:
```
l.count(40)
```

In [ ]:
```
l
```

In [ ]:
```
l.count(100)
```

In [ ]:
```
l
```

In [ ]:
```
l1
```

In [ ]:
```
l2 = l1.copy()
```

In [ ]:
```
l2
```

In [ ]:
```
l2.remove(True)
```

In [ ]:
```
l2
```

In [ ]:
```
l2
```

In [ ]:
```
l2.remove(1+2j)
l2
```

In [ ]:
```
l2
```

In [ ]:
```
l2.clear()
```

In [ ]:
```
l2
```

In [ ]:
```
del l2
```

In [ ]:
```
l2
```

4th march - list

In [ ]:
```
print(l)
print(l1)
```

In [ ]:
```
print(len(l))
print(len(l1))
```

In [ ]:
```
l
```

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

In [ ]:
```
l.append([1,2,3,'hi']) #nested list
l
```

In [ ]:
```
l
```

In [ ]:
```
l.remove(40) #remove the element
```

In [ ]:
```
l
```

In [ ]:
```
l
```

In [ ]:
```
l[4]
```

In [ ]:
```
l
```

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

```
In [ ]: l
```

```
In [ ]: l1
```

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

```
In [ ]: l1
```

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

```
In [ ]: l1
```

```
In [ ]: l1.pop(-1)
```

```
In [ ]: l1
```

```
In [ ]: l1.pop(3)
```

```
In [ ]: l1
```

```
In [ ]: print(l)
        print(l1)
```

```
In [ ]: l
```

```
In [ ]: l.insert(35,3)
```

```
In [ ]: l
```

```
In [ ]: l.insert(3,35)
```

```
In [ ]: l
```

```
In [ ]: l1
```

```
In [ ]: l1.insert(15,1)
```

```
In [ ]: l1
```

```
In [ ]: l1.insert(1, 15)
```

```
In [ ]: l1
```

```
In [ ]: l2 = []
```

```
In [ ]: l2
```

```
In [ ]: l2.extend(l1)
```

```
In [ ]: l2
```

In [ ]:
```
l
```

In [ ]:
```
l1
```

In [ ]:
```
l.extend(l1)
```

In [ ]:
```
l
```

In [ ]:
```
print(l)
```

In [ ]:
```
print(l1)
```

In [ ]:
```
print(l2)
```

In [ ]:
```
l2.index(30)
```

In [ ]:
```
l2.
```

In [ ]:
```
l
```

In [ ]:
```
l.index(30)
```

In [ ]:
```
l
```

In [ ]:
```
l1
```

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

In [ ]:
```
l1
```

In [ ]:
```
l1.sort(reverse=True) #descending order
```

In [ ]:
```
l1
```

In [ ]:
```
l3 = [3, 100, 4]
l3
```

In [ ]:
```
l3.sort()
```

In [ ]:
```
l3
```

In [ ]:
```
l3.sort(reverse = True)
```

In [ ]:
```
l3
```

In [ ]:
```
l6 = [3, 5.6, 'a', 1+2j]
```

In [ ]:
```
l6.sort()
```

In [ ]:
```
l5 = ['z', 'm', 'n', 'b']
l5
```

In [ ]:
```
l5.sort()
```

In [ ]:
```
l5
```

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

In [ ]:
```
l1
```

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

In [ ]:
```
l1
```

In [ ]:
```
l
```

In [ ]:
```
l[::-1]
```

5th

In [ ]:
```
print(l)
print(l1)
print(l2)
```

strint list slicing (datatype)

In [ ]:
```
s1 = 'nit'
s1
```

In [ ]:
```
s1[0]
```

In [ ]:
```
s1[1]
```

In [ ]:
```
s1[2]
```

In [ ]:
```
s1[3]
```

In [ ]:
```
s1
```

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

In [ ]:
```
s1
```

list slicing

In [ ]:
```
print(l)
```

In [ ]:
```
l[:]
```

```
In [ ]: l[0:8]
```

```
In [ ]: l
```

```
In [ ]: l[3:]
```

```
In [ ]: l
```

```
In [ ]: l[:7]
```

```
In [ ]: l
```

```
In [ ]: l[0:20:5]
```

```
In [ ]: l
```

```
In [ ]: l[3:10:3]
```

```
In [ ]: l
```

```
In [ ]: l[::-1]
```

```
In [ ]: l
```

```
In [ ]: l[::-2]
```

```
In [ ]: l
```

```
In [ ]: l[::-3]
```

```
In [ ]: l1
```

```
In [ ]: l1[0]
```

```
In [ ]: l1[0] =  45
```

```
In [ ]: l1
```

```
In [ ]: l1
```

```
In [ ]: l1[-1] = 'nit'
```

```
In [ ]: l1
```

```
In [ ]: l1[-1][0] # nested slicing
```

```
In [ ]: print(l1[-1][0])
        print(l1[-1][1])
        print(l1[-1][2])
```

```
In [ ]: l1
In [ ]: l2
```

```
In [ ]: len(l2)
```

```
In [ ]: l3
```

```
In [ ]: l4 = l2 + l3
```

```
In [ ]: # list membership
```

```
In [ ]: l4
```

```
In [ ]: 15 in l4
```

ENumerate

```
In [ ]: l1
```

```
In [ ]: for i in l1:
            print(i)
```

```
In [ ]: for i in enumerate(l1):
            print (i)
```

```
In [ ]: l1
```

```
In [ ]: l3
```

```
In [ ]: all(l3)
```

```
In [ ]: any(l3)
```

l3.append(0) l3

```
In [ ]: all(l3)
```

```
In [ ]: any(l3)
```

list completed

6th mar

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

```
In [ ]: print(type(t))
In [ ]: t1 = (10,20,30,40,40)
        t1
```

```
In [ ]: len(t1)
```

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

```
In [ ]: t1.count(40)
```

```
In [ ]: t1
```

```
In [ ]: t1.index(20)
```

```
In [ ]: l5 = ['a', 'b', 'c', 'd']
        l5
```

```
In [ ]: l5[1] = 10
```

```
In [ ]: l5
```

```
In [ ]: t2 = (100, 3.4, 'nit', True, 1+2j, [1,2,3], (5,6,7))
        t2
```

```
In [ ]: print(t)
        print(t1)
        print(t2)
```

```
In [ ]: t1
```

```
In [ ]: t1[0]
```

```
In [ ]: t1[0] = 1000
        t1
```

```
In [ ]: icici = (45678, 'cizps7789', 332000, 98765)
        icici
```

```
In [ ]: icici[0] = 1234
        icici
```

```
In [ ]: t1
```

```
In [ ]: t4 = t1 * 3
        t4
```

```
In [ ]: t4
```

```
In [ ]: t4[:]
```

```
In [ ]: t1
```

```
In [ ]: t1[:7]
In [ ]: t1[2:]
```

```
In [ ]: t1
```

```
In [ ]: t1[0]
```

```
In [ ]: t1[0:10:2]
```

```
In [ ]: t1.add(30)
```

```
In [ ]: t2
```

```
In [ ]: t2.index('nit')
```

tuple is completed

```
In [ ]: x = 5
        y = 2
        print(x / y)
        print(x//y)
```

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

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

7th

set

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

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

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

```
In [ ]: s1
```

```
In [ ]: s2 = {20, 100, 3, 45}
        s2
```

```
In [ ]: s3 = {'z', 'l', 'c', 'e', 'f'}
        s3
```

```
In [ ]: s4 = {1, 2.3, 'nit', 1+2j, [1,2,3], (4,5,6), True}
```

```
s4
```

In [ ]:
```
s5 = {2, 3.4, 'nit', 1+2j, False}
```

In [ ]:
```
s5
```

In [ ]:
```
print(s1)
print(s2)
print(s3)
print(s5)
```

In [ ]:
```
s2
```

In [ ]:
```
s2.add(30)
```

In [ ]:
```
s2
```

In [ ]:
```
s2.add(200)
```

In [ ]:
```
s2
```

In [ ]:
```
s2
```

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

In [ ]:
```
s2
```

In [ ]:
```
s2[1:5]
```

In [ ]:
```
s5
```

In [ ]:
```
s4 = s5.copy()
s4
```

In [ ]:
```
s4
```

In [ ]:
```
s4.add(2)
```

In [ ]:
```
s4
```

In [ ]:
```
s5
```

In [ ]:
```
s5.clear()
```

In [ ]:
```
s5
```

In [ ]:
```
del s5
```

In [ ]:
```
s4
```

In [ ]:
```
s4.remove((1+2j))
```

In [ ]:

```
s4
```

In [ ]:

```
s3
```

In [ ]:

```
s3.discard('m')
```

In [ ]:

```
s3.remove('m')
```

In [ ]:

```
s3
```

In [ ]:

```
s3.discard('f')
s3
```

In [ ]:

```
s3
```

In [ ]:

```
s3.pop()
```

In [ ]:

```
s3
```

In [ ]:

```
s2
```

In [ ]:

```
s2.pop(3)
```

In [ ]:

```
s2.pop()
```

In [ ]:

```
for i in s2:
    print(i)
```

In [ ]:

```
for i in enumerate(s2):
    print(i)
```

In [ ]:

```
s2
```

In [ ]:

```
5 in s2
```

In [ ]:

```
45 in s2
```

In [ ]:

```
s2
```

In [ ]:

```
s3
```

In [ ]:

```
s2.update(s3)
```

In [ ]:

```
s2
```

SET OPERATION

In [ ]:

```
s6 = {1,2,3,4,5}
s7 = {4,5,6,7,8}
s8 = {8,9,10}
```

In [ ]:
```
s6.union(s7)
```

In [ ]:
```
s6.union(s7, s8)
```

In [ ]:
```
s6 | s7
```

In [ ]:
```
s6 | s7 | s8
```

In [ ]:
```
print(s6)
print(s7)
print(s8)
```

In [ ]:
```
s6.intersection(s7)
```

In [ ]:
```
s6.intersection(s8)
```

In [ ]:
```
s7.intersection(s8)
```

In [ ]:
```
s6 & s7
```

In [ ]:
```
print(s6)
print(s7)
print(s8)
```

In [ ]:
```
s6.difference(s7)
```

In [ ]:
```
s6 - s7
```

In [ ]:
```
s7 - s8
```

In [ ]:
```
print(s6)
print(s7)
print(s8)
```

In [ ]:
```
s8 - s7
```

In [ ]:
```
print(s6)
print(s7)
print(s8)
```

In [ ]:
```
s6.symmetric_difference(s7)
```

In [ ]:
```
s10 = {50, 4, 3, 10}
s10
```

In [ ]:
```
print(s10)
```

In [ ]:
```
print(s10)
```

10th

- superset
- subset
- disjoint

```
In [ ]: s11 = {1,2,3,4,5,6,7,8,9}
        s12 = {3,4,5,6,7,8}
        s13 = {10,20,30,40}
```

```
In [ ]: s12.issubset(s11)
```

```
In [ ]: s11.issubset(s12)
```

```
In [ ]: s11.issuperset(s12)
```

```
In [ ]: s11 = {1,2,3,4,5,6,7,8,9}
        s12 = {3,4,5,6,7,8}
        s13 = {10,20,30,40}
```

```
In [ ]: s13.isdisjoint(s12)
```

```
In [ ]: s13.isdisjoint(s11)
```

```
In [ ]: s12 = {1,2,3,4,5}
        s13 = {10,20,30}
        s14 = {15,25,35}
```

```
In [ ]: s13.issubset(s12)
```

```
In [ ]: s12.issuperset(s13)
```

```
In [ ]: s14.isdisjoint(s12)
```

```
In [ ]: s14.isdisjoint(s13)
```

```
In [3]: s15 = {1,2,3,4,5,6}
        s16 = {4,5,6}
        s17 = {10,20}
```

```
In [4]: s16.issubset(s15)
```

```
Out[4]: True
```

```
In [5]: s17.isdisjoint(s15)
```

```
Out[5]: True
```

```
In [6]: s17.isdisjoint(s16)
```

```
Out[6]: True
```

```
In [8]: s15
```

```
Out[8]: {1, 2, 3, 4, 5, 6}
```

In [9]:

```
for i in s15:
    print(i)
```

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

In [10]:

```
for i in enumerate(s15):
    print(i)
```

```
(0, 1)
(1, 2)
(2, 3)
(3, 4)
(4, 5)
(5, 6)
```

In [12]:

```
s15
```

Out[12]:

```
{1, 2, 3, 4, 5, 6}
```

In [13]:

```
sum(s15)
```

Out[13]:

```
21
```

In [14]:

```
min(s15)
```

Out[14]:

```
1
```

set is completed

dictionary

In [17]:

```
d = {}
d
```

Out[17]:

```
{}
```

In [18]:

```
type(d)
```

Out[18]:

```
dict
```

In [20]:

```
d1 = {1 : 'one', 2 : 'two', 3: 'three'}
d1
```

Out[20]:

```
{1: 'one', 2: 'two', 3: 'three'}
```

In [22]:

```
d1.keys()
```

Out[22]:

```
dict_keys([1, 2, 3])
```

In [23]:
```
d1.values()
```

Out[23]:
```
dict_values(['one', 'two', 'three'])
```

In [24]:
```
d2 = d1.copy()
d2
```

Out[24]:
```
{1: 'one', 2: 'two', 3: 'three'}
```

In [25]:
```
d1.items()
```

Out[25]:
```
dict_items([(1, 'one'), (2, 'two'), (3, 'three')])
```

In [26]:
```
d1[1]
```

Out[26]:
```
'one'
```

In [29]:
```
keys = {'ram' , 'b' , 'c' , 'd'}
value = [10,20,30]
mydict3 = dict.fromkeys(keys , value) # Create a dictionary from a sequence of
mydict3
```

Out[29]:
```
{'c': [10, 20, 30], 'd': [10, 20, 30], 'ram': [10, 20, 30], 'b': [10, 20, 30]}
```

In [30]:
```
value.append(50)
mydict3
```

Out[30]:
```
{'c': [10, 20, 30, 50],
 'd': [10, 20, 30, 50],
 'ram': [10, 20, 30, 50],
 'b': [10, 20, 30, 50]}
```

In [31]:
```
range(10)
```

Out[31]:
```
range(0, 10)
```

In [32]:
```
list(range(0,10))
```

Out[32]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [33]:
```
list(range(10,20))
```

Out[33]:
```
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

In [34]:
```
list(range(10,20,3))
```

Out[34]:
```
[10, 13, 16, 19]
```

In [35]:
```
list(range(10,20,3,4))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[35], line 1
----> 1 list(range(10,20,3,4))

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

```
In [36]: r = range(1,10)
         r
```

```
Out[36]: range(1, 10)
```

```
In [37]: for i in r:
             print(i)
```

```
1
2
3
4
5
6
7
8
9
```

```
In [ ]:
```