WORK WITH PYTHON NUMBERS

```
In [1]:   5

Out[1]:   5

In [2]:   5+5

Out[2]:   10

In [3]:   -5-5

Out[3]:   -10

In [4]:   5+6-7*3-7

Out[4]:   -17

In [5]:   5+6-7*(3-7)

Out[5]:   39

In [6]:   _

Out[6]:   39

In [7]:   _+1

Out[7]:   40

In [8]:   _=4

In [9]:   _-4

Out[9]:   0

In [10]:  a=3
          b=4

In [11]:  int.__add__(a,b)

Out[11]:  7

In [12]:  a=3
          b=4

In [13]:  int.__sub__(a,b)

Out[13]:  -1

In [14]:  c=suunitha
          d=venu
```

```
int.__add__(c,d)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[14], line 1
----> 1 c=suunitha
      2 d=venu
      3 int.__add__(c,d)

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

```
In [ ]: str.__add__(c,d)
```

```
In [ ]:
```

works with text

```
In [ ]: Naresh IT
```

```
In [ ]: 'Naresh IT'
```

```
In [ ]: "Naresh IT"
```

```
In [ ]: '''Naresh IT'''
```

```
In [ ]: 'Naresh
            Technology.
```

```
In [ ]: "Naresh
           Technology"
```

```
In [ ]: '''Naresh
            Technology'''
```

```
In [ ]:
```

```
In [ ]:
```

28th varibles

```
In [ ]: v=10
        v
```

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

```
In [ ]:
```

```
In [ ]: nit=8
        NIT
```

```
In [ ]:
```

```
In [ ]: 8=nit
```

```
In [ ]:
```

```
In [ ]: 8nit=10
```

```
In [ ]:
```

```
In [ ]: nit8=20
        nit8
```

```
In [ ]:
```

```
In [ ]: nit$=50
        nit$
```

```
In [ ]:
```

```
In [ ]: nit_=78
        nit_
```

```
In [ ]:
```

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

```
In [ ]:
```

```
In [ ]: def=50
        def
```

```
In [ ]:
```

```
In [ ]: DEF=60
        DEF
```

```
In [ ]:
```

```
In [ ]: 3+4
```

```
In [ ]: 3=4
        4=5
```

```
In [ ]: 3+4
        4+5
```

```
In [ ]:
```

```
In [ ]: 3+4
        4+5
        5+6
```

```
In [ ]:

In [ ]: print(3+4)
        print(4+5)
        print(5+6)

In [ ]:

In [ ]:
```

1st Python datatypes

```
In [ ]: i=100
        i

In [ ]: type(i)

In [ ]:

In [ ]:

In [ ]: j=200.56
        j

In [ ]: type(j)

In [ ]:

In [ ]: f1=1e0
        f1

In [ ]: f2=1e1
        f2

In [ ]: f3=1e2
        f3

In [ ]: f4=2e3
        f4

In [ ]:

In [ ]: import keyword
        keyword.kwlist

In [ ]:

In [ ]: b = True
        b

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

```
b + b1
```

In [ ]:

```
print(b+b1)
print(b-b1)
print(b*b1)
```

In [ ]:

```
False /True
```

In [ ]:

```
c=False
c
```

In [ ]:

```
print(True)
```

In [ ]:

```

```

In [ ]:

```
False/True
```

In [ ]:

```
False//True
```

In [ ]:

```
True/False
```

In [ ]:

```

```

complex

In [ ]:

```
c=10+20j
c
```

In [ ]:

```
type(c)
```

In [ ]:

```

```

In [ ]:

```
c.real
```

In [ ]:

```
c.imag
```

In [ ]:

```
False+True
```

In [ ]:

```

```

In [ ]:

```
c1=10+20.3j
c1
```

In [ ]:

```
c2=30+53.8
c2
```

In [ ]:

```
c1+c2
```

In [ ]:

```

```

string

In [ ]:

```
name='sunitha'
name
```

In [ ]:

```
type(name)
```

In [ ]:

```

```

In [ ]:

```
name[1]
```

In [ ]:

```
name[0]
```

In [ ]:

```
name[-1:0]
```

In [ ]:

```
name[0:-1]
```

In [ ]:

```
name[1:-1]
```

In [ ]:

```
name[10]
```

In [ ]:

```
name[-10]
```

In [ ]:

```
name[:]
```

In [ ]:

```
name[2:]
```

In [ ]:

```
name[:-1]
```

In [ ]:

```
name[:5]
```

In [ ]:

```

```

In [ ]:

```
name
```

In [ ]:

```
name[1:-2:3]
```

In [ ]:

```

```

backward index

In [ ]:

```
name[-1]
```

In [ ]:

```
name[-2:]
```

In [ ]:

```
name[-2:0]
```

In [ ]:

```
name[-2:-1]
```

In [ ]:

```

```

In [ ]:

```python
movie = '''Hero: Yash (as Rocky)
Heroine: Srinidhi Shetty
Villain: Garuda
Music: Ravi Basrur

Story in short:

Rocky, born in poverty, grows up to become a powerful gangster in Mumbai.
\He is sent to assassinate Garuda, the ruthless ruler of the Kolar Gold Fields.
The film ends with Rocky killing Garuda, setting the stage for Chapter 2.'''
```

In [ ]:

```python
movie
```

In [ ]:

```

```

In [ ]:

```

```

2nd python typecasting

convert all datatypes to int except complelx and text string

In [ ]:

```python
int(100.5)
```

In [ ]:

```python
int(100.9)
```

In [ ]:

```

```

In [ ]:

```python
int(True)
```

In [ ]:

```python
int(False)
```

In [ ]:

```

```

In [ ]:

```python
int(10+20j)
```

In [ ]:

```

```

In [ ]:

```python
int("10")
```

In [ ]:

```python
int("ten")
```

In [ ]:

```

```

cast other datatypes to float

```
In [ ]: float(25)

In [ ]: float(25,39)

In [ ]:

In [ ]: float(10+20j)

In [ ]:

In [ ]: float("10")

In [ ]: float("ten")

In [ ]:
```

cast other datatypes to complex

```
In [ ]: complex(10)

In [ ]: complex(20,10)

In [ ]: complex(20,10,5)

In [ ]:

In [ ]: complex(2.9)

In [ ]:

In [ ]: complex(2.0+37.8)

In [ ]: complex(True,False)

In [ ]: complex(False,True)

In [ ]:

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

In [ ]:

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

In [ ]:

In [ ]: complex("10")

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

In [ ]:

In [ ]:

```
complex('ten')
```

In [ ]:

cast other datatypes to boolean

In [ ]:

```
bool()
```

In [ ]:

```
bool(10)
```

In [ ]:

```
bool(9.8)
```

In [ ]:

```
bool(0)
```

In [ ]:

```
bool(10+20j)
```

In [ ]:

```
bool("sunitha")
```

In [ ]:

In [ ]:

```
print(10) # 1 arg
print(10,20) # 2 arg
print('python') # string arg
print(10,20,'python') # 3 arg
```

In [ ]:

In [ ]:

```
num1=10
num2=20
add=num1+num2
print('The addition of', num1, 'and', num2, 'is:', add)
```

In [ ]:

In [ ]:

```
num1,num2, num3=10,20,30
add=num1+num2+num3
print('The addition of {} and {} and {} is {}'.format(num1,num2,num3,add))
```

In [ ]:

```
print(f'The addition of {num1} and{num2} and {num3} is {add}')
```

In [ ]:

In [ ]:

```
print('hello')
print('good morning')
```

In [ ]:

In [ ]:

```
print('hello',end='_')
print('good morning')
```

```
In [ ]:

In [ ]: print('*')
        print('**',end="")
        print('****')

In [ ]:

In [ ]: print('hello','hai','how are you',sep='--->')

In [ ]: print('hello','hai','how are you',sep='$-$')

In [ ]:
```

3rd python datastructure

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

In [ ]: type(l)

In [ ]: len(l)

In [ ]: l

In [ ]: l.append(10)
        l

In [ ]:

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

In [ ]: l

In [ ]: l2=l.copy()
        l2

In [ ]:

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

In [ ]:

In [ ]: l.append(20.3, 'hi', 20+10j, True)

In [ ]:
```

In [ ]:

```
l.append(20.3)
l.append('hi')
l.append(20+10j)
l.append(True)
```

In [ ]:

```
l
```

In [ ]:

```
 print(l)
print(l2)
```

In [ ]:

In [ ]:

```
l2.clear()
```

In [ ]:

```
l2
```

In [ ]:

```
del l2
```

In [ ]:

```
l2
```

In [ ]:

```
l
```

In [ ]:

```
l.count(10)
```

In [ ]:

```
l.count(20)
```

In [ ]:

```
l[:]
```

In [ ]:

```
l[1:]
```

In [ ]:

```
l[5:]
```

In [ ]:

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

In [ ]:

In [ ]:

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

In [ ]:

```
l.index('hi')
```

In [ ]:

```
l[8]
```

In [ ]:

In [ ]:

```
l
```

In [ ]:

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

In [ ]:

```
l[::2]
```

In [ ]:

In [ ]:
```
l[3:14:7]
```

In [ ]:
```

```

9th list

In [ ]:
```
l=[10,20,30,50,390,230]
l
```

In [ ]:
```
l2=['sunitha', 'venu', 'rushik','moksha']
l2
```

In [ ]:
```

```

In [ ]:
```
l.append(101)
```

In [ ]:
```
l
```

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

In [ ]:
```
l[2]=202
l
```

In [ ]:
```
l2[-1]=1000
l2
```

In [ ]:
```

```

In [ ]:
```
l3=[]
l3
```

In [ ]:
```
l3.extend(l2)
l3
```

In [ ]:
```
l2
```

In [ ]:
```
l3.extend(l)
l3
```

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

In [ ]:
```

```

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

In [ ]:
```
l1
```

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

In [ ]:

```

```

In [ ]:

```
print(l)
print(l2)
print(l3)
```

In [ ]:

```
print(l1)
```

In [ ]:

```

```

In [ ]:

```
l1.index('sunitha')
```

In [ ]:

```

```

In [ ]:

```
l1.insert(2,'deepu')
```

In [ ]:

```
l1
```

In [ ]:

```
l
```

In [ ]:

```
l.insert(0,0)
```

In [ ]:

```
l
```

In [ ]:

```

```

In [ ]:

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

In [ ]:

```

```

In [ ]:

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

In [ ]:

```

```

In [ ]:

```
my_list = ['apple', 'banana', 'cherry']

# Using enumerate with default start (0)
for index, item in enumerate(my_list):
    print(f"Index: {index}, Item: {item}")

# Using enumerate with a custom start (1)
for index, item in enumerate(my_list, start=1):
    print(f"Position: {index}, Fruit: {item}")
```

In [ ]:

```

```

In [ ]:

```
l
```

In [ ]:

```
l.pop()
```

In [ ]:

```
l
```

In [ ]:
```
l.pop(1)
l
```

In [ ]:
```

```

In [ ]:
```
del[l[2]]
```

In [ ]:
```
l
```

In [ ]:
```
l.remove(390)
```

In [ ]:
```
l
```

In [ ]:
```
l1
```

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

In [ ]:
```
l1
```

In [ ]:
```
l1[::-1]
l1
```

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

In [ ]:
```
l1.remove(1000)
```

In [ ]:
```
l1
```

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

In [ ]:
```

```

In [ ]:
```
l2
```

In [ ]:
```
l3
```

In [ ]:
```
l1
```

In [ ]:
```
l
```

In [ ]:
```
l4=['m','l', 'M','d']
l4
```

In [ ]:
```
l4.sort()
l4
```

In [ ]:
```
l5=[34,24,456,342]
l5
```

```
In [ ]: l5.sort(reverse=True)
        l5
```

```
In [ ]:
```

```
In [ ]: l5
```

```
In [ ]: 200 in l5
```

```
In [ ]:
```

```
In [ ]: all(l5)
```

```
In [ ]: any(l5)
```

```
In [ ]: l5.append(0)
        l5
```

```
In [ ]: all(l5)
```

```
In [ ]: any(l5)
```

```
In [ ]: l5.remove(0)
        l5
```

```
In [ ]: any(l5)
```

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

```
In [ ]: any(l5)
```

```
In [ ]:
```

Tuple

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

```
In [ ]: t
```

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

```
In [ ]: t=(10,20,30)
        t
```

```
In [ ]: t.remove(30)
```

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

```
In [ ]: t.count(2)
```

In [ ]:
```
t[::]
```

In [ ]:
```
t[:2]
```

In [ ]:
```
t.index(2)
```

In [ ]:
```
t
```

In [ ]:
```
t.index(10)
```

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

In [ ]:
```
t.count(30)
```

In [ ]:

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

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

In [ ]:

In [ ]:
```
t1= ([1,2,3,], 3, True, 1+2j, 'hi')
t1
```

In [ ]:

10th set

In [ ]:
```
s={10,2,23,54,342,2}
s
```

In [ ]:
```
s1={'a', 'sunitha',10+20j,True,[1,2,3],'b','r'}
s1
```

In [ ]:

In [ ]:
```
s2={'a', 'sunitha',10+20j,True,'b','r'}
s2
```

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

In [ ]:
```
print(id(s))
print(id(s2))
```

In [ ]:
```
s.add(10)
s
```

In [ ]:
```
s.add(100)
s
```

In [ ]:
```
s2.add('venu')
```

In [ ]:
```
s2
```

In [ ]:
```
s2.add(20+10j)
s2
```

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

In [ ]:
```
s2.add('e')
```

In [ ]:
```
s2
```

In [ ]:
```

```

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

In [ ]:
```
s==s2
```

In [ ]:
```
s!=s2
```

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

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

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

In [ ]:
```
s2
```

In [ ]:
```
s2.pop()
```

In [ ]:
```
s2.pop()
```

In [ ]:
```
s2.pop()
```

In [ ]:
```

```

In [15]:
```
s
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[15], line 1
----> 1 s

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

```
In [ ]:
```

```
In [16]: s2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[16], line 1
----> 1 s2

NameError: name 's2' is not defined
```

```
In [ ]:
```

```
In [17]: s={1,3,434,32798,132,4,23,23}
```

```
In [18]: s
```

```
Out[18]: {1, 3, 4, 23, 132, 434, 32798}
```

```
In [19]: s2={132,'venu','sunitha','sd'}
```

```
In [20]: s2
```

```
Out[20]: {132, 'sd', 'sunitha', 'venu'}
```

```
In [21]: s.remove(434)
         s
```

```
Out[21]: {1, 3, 4, 23, 132, 32798}
```

```
In [22]: s.discard(23)
```

```
In [23]: s
```

```
Out[23]: {1, 3, 4, 132, 32798}
```

```
In [24]: s.remove(432)
         s
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[24], line 1
----> 1 s.remove(432)
      2 s

KeyError: 432
```

```
In [25]: s.discard(100)
         s
```

```
Out[25]: {1, 3, 4, 132, 32798}
```

```
In [26]: s.discard(432)
```

```
In [27]: s
```

```
Out[27]: {1, 3, 4, 132, 32798}

In [ ]:

In [28]: s3={1,2,3,4,5,6}
         s4={4,5,6,7,8}
         s={8,9,10}

In [33]: print(s3)
         print(s4)
         print(s)

        {1, 2, 3, 4, 5, 6}
        {4, 5, 6, 7, 8}
        {1, 3, 132, 4, 32798}

In [32]: print(s6)

        ---------------------------------------------------------------------------
        NameError                                 Traceback (most recent call last)
        Cell In[32], line 1
        ----> 1 print(s6)

        NameError: name 's6' is not defined

In [34]: s7={3,67,87}

In [35]: s7

Out[35]: {3, 67, 87}

In [37]: print(s3)
         print(s4)
         print(s)

        {1, 2, 3, 4, 5, 6}
        {4, 5, 6, 7, 8}
        {1, 3, 132, 4, 32798}

In [38]: s3.union(s4)

Out[38]: {1, 2, 3, 4, 5, 6, 7, 8}

In [39]: s3|s4

Out[39]: {1, 2, 3, 4, 5, 6, 7, 8}

In [40]: s|s3|s4

Out[40]: {1, 2, 3, 4, 5, 6, 7, 8, 132, 32798}

In [42]: s.intersection(s3)

Out[42]: {1, 3, 4}

In [43]: s & s3
```

```
Out[43]: {1, 3, 4}
```

```
In [44]: s&s4
```

```
Out[44]: {4}
```

```
In [ ]:
```

```
In [45]: s3.difference(s4)
```

```
Out[45]: {1, 2, 3}
```

```
In [ ]:
```

```
In [46]: s3-s4
```

```
Out[46]: {1, 2, 3}
```

```
In [ ]:
```

```
In [47]: s4-s3
```

```
Out[47]: {7, 8}
```

```
In [48]: s
```

```
Out[48]: {1, 3, 4, 132, 32798}
```

```
In [49]: s4
```

```
Out[49]: {4, 5, 6, 7, 8}
```

```
In [50]: s.symmetric_difference(s4)
```

```
Out[50]: {1, 3, 5, 6, 7, 8, 132, 32798}
```

```
In [55]: sa={1,2,3,5}
         sa
```

```
Out[55]: {1, 2, 3, 5}
```

```
In [53]: sb={1,2,3,4}
         sb
```

```
Out[53]: {1, 2, 3, 4}
```

```
In [56]: sa.symmetric_difference(sb)
```

```
Out[56]: {4, 5}
```

```
In [57]: sa-sb
```

```
Out[57]: {5}
```

```
In [58]: sb-sa
```

```
Out[58]:  {4}

In [59]:  sa^sb

Out[59]:  {4, 5}

In [60]:  sb^sa

Out[60]:  {4, 5}

In [61]:  len(sa)

Out[61]:  4

In [62]:  len(s4)

Out[62]:  5

In [63]:  sa.difference_update(sb)

In [67]:  sb.difference_update(sa)
          sb

Out[67]:  {1, 2, 3, 4}

In [65]:  sa

Out[65]:  {5}

In [66]:  sb

Out[66]:  {1, 2, 3, 4}

In [68]:  s

Out[68]:  {1, 3, 4, 132, 32798}

In [69]:  s3

Out[69]:  {1, 2, 3, 4, 5, 6}

In [71]:  s.difference_update(s3)
          s

Out[71]:  {132, 32798}

In [73]:  s3.difference_update(s)
          s3

Out[73]:  {1, 2, 3, 4, 5, 6}


In [76]:  A={1,2,3,4,5,6,7,8}
          A
```

```
Out[76]: {1, 2, 3, 4, 5, 6, 7, 8}
```

```
In [87]: B={1,2,3,4}
         B
```

```
Out[87]: {1, 2, 3, 4}
```

```
In [79]: C={10,33,32,578,78}
         C
```

```
Out[79]: {10, 32, 33, 78, 578}
```

```
In [81]: A.issubset(B)
```

```
Out[81]: False
```

```
In [88]: B.issubset(A)
```

```
Out[88]: True
```

```
In [89]: B.isdisjoint(A)
```

```
Out[89]: False
```

```
In [90]: C.isdisjoint(A)
```

```
Out[90]: True
```

```
In [91]: A.issuperset(B)
```

```
Out[91]: True
```

```
In [93]: A.update(B)
         A
```

```
Out[93]: {1, 2, 3, 4, 5, 6, 7, 8}
```

```
In [94]: A.update(C)
```

```
In [95]: A
```

```
Out[95]: {1, 2, 3, 4, 5, 6, 7, 8, 10, 32, 33, 78, 578}
```

```
In [ ]:
```

Dict()

```
In [96]: l=[1,2,3,4,5,6,7,8]
         l
```

```
Out[96]: [1, 2, 3, 4, 5, 6, 7, 8]
```

```
In [97]: l[::1]
```

```
Out[97]:  [1, 2, 3, 4, 5, 6, 7, 8]
```

```
In [98]:  l[::-1]
```

```
Out[98]:  [8, 7, 6, 5, 4, 3, 2, 1]
```

```
In [99]:  l[::-2]
```

```
Out[99]:  [8, 6, 4, 2]
```

```
In [100…  l[::2]
```

```
Out[100…  [1, 3, 5, 7]
```

```
In [ ]:
```

```
In [101…  d={}
          d
```

```
Out[101…  {}
```

```
In [102…  type(d)
```

```
Out[102…  dict
```

```
In [103…  s11={12,23,34,45,56}
          s11
```

```
Out[103…  {12, 23, 34, 45, 56}
```

```
In [104…  type(s11)
```

```
Out[104…  set
```

```
In [105…  d={}
          d
```

```
Out[105…  {}
```

```
In [108…  d1={'one':1,'two':2,'three':3,'four':4,'five':5,1+2j:'complex'}
          d1
```

```
Out[108…  {'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5, (1+2j): 'complex'}
```

```
In [109…  d1['ten']=10
          d1
```

```
Out[109…  {'one': 1,
           'two': 2,
           'three': 3,
           'four': 4,
           'five': 5,
           (1+2j): 'complex',
           'ten': 10}
```

```
In [110…  d1.keys()
```

Out[110…

```
dict_keys(['one', 'two', 'three', 'four', 'five', (1+2j), 'ten'])
```

In [111…

```
d1.values()
```

Out[111…

```
dict_values([1, 2, 3, 4, 5, 'complex', 10])
```

In [112…

```
d1.items()
```

Out[112…

```
dict_items([('one', 1), ('two', 2), ('three', 3), ('four', 4), ('five', 5), ((1
+2j), 'complex'), ('ten', 10)])
```

In [ ]:

In [113…

```
keys={'a','b','c','d'}
mydict3=dict.fromkeys(keys)
mydict3
```

Out[113…

```
{'d': None, 'c': None, 'b': None, 'a': None}
```

In [114…

```
keys = {'a' , 'b' , 'c' , 'd'}
mydict3 = dict.fromkeys(keys) # Create a dictionary from a sequence of keys
mydict3
```

Out[114…

```
{'d': None, 'c': None, 'b': None, 'a': None}
```

In [ ]:

In [115…

```
for i in (d1):
 print(i)
```

```
one
two
three
four
five
(1+2j)
ten
```

In [116…

```
for i in (d1):
    print(i,':',d1[i])
```

```
one : 1
two : 2
three : 3
four : 4
five : 5
(1+2j) : complex
ten : 10
```

In [117…

```
d1
```

Out[117…

```
{'one': 1,
 'two': 2,
 'three': 3,
 'four': 4,
 'five': 5,
 (1+2j): 'complex',
 'ten': 10}
```

```
In [ ]:
```

Range

```
In [119…  range(10)

Out[119…  range(0, 10)

In [120…  range(10,20)

Out[120…  range(10, 20)

In [121…  range(10,100,2)

Out[121…  range(10, 100, 2)

In [122…  range(10,100,2,3)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[122], line 1
----> 1 range(10,100,2,3)

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

```
In [126…  list(range(10))

Out[126…  [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

In [127…  list(range(10,20))

Out[127…  [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]

In [128…  r=list(range(10,20))
          r

Out[128…  [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]

In [131…  for i in (r):
           print(i)
```

```
10
11
12
13
14
15
16
17
18
19
```

```
In [ ]:
```