In [3]:

```
#@author: Bhanu Prakash

num = 2.5

print(type(num)) #number which is of float type
```

```
<class 'float'>
```

In [5]:

```
num = 5

print(type(num)) #number which is of integer type
```

```
<class 'int'>
```

In [7]:

```
num = 6+9j #complex number we will represent it as 'a+bi' but here we will represent it as 'a+bj'

print(type(num)) #number which is of complex type
```

```
<class 'complex'>
```

In [9]:

```
a = 5.6 #the number it is of type float

b = int(a) #where as I want 'b' as integer value of 'a' So I can mention it as int(a) then it will be converted into

print(b)

print(type(b)) #number which is of integer type
```

```
5
<class 'int'>
```

In [12]:

```
a = 5.6 #the number it is of type float

b = int(a) #where as I want 'b' as integer value of 'a' So I can mention it as int(a) then it will be converted into

print(b)

print(type(b)) #number which is of integer type

k = float(b) #where as I want 'k' as float value of 'b' So I can mention it as float(b) then it will be converted int

print(k)
print(type(k)) #number which is of float type
```

```
5
<class 'int'>
5.0
<class 'float'>
```

In [14]:

```
a = 5

b = 7

c = complex(a,b) #converting normal number into complex number by using complex()

print(c)
```

```
(5+7j)
```

In [16]:

```
a = 10

b = 5

c = a < b

print(c) #bool type is a 'True' or 'False'

print(type(c)) #it is of type 'bool' which will tell's 'Ture' or 'False'
```

```
False
<class 'bool'>
```

In [21]:

```
a = 10

b = 5

c = a < b

d = a > b

print(c) #bool type is a 'True' or 'False'

print(d)
print(type(c)) #it is of type 'bool' which will tell's 'Ture' or 'False'

print(type(d))

print(int(True)) #if we want to convert 'True' to integer value in python it is of '1'

print(int(False)) #if we want to convert 'False' to integer value in python it is of '0'
```

```
False
True
<class 'bool'>
<class 'bool'>
1
0
```

In [26]:

```
lst = [20,21,21,44,33,55,66]

print(lst)

print(type(lst)) #it is of type list
```

```
[20, 21, 21, 44, 33, 55, 66]
<class 'list'>
```

In [27]:

```
s = {20,21,21,44,66,8,9,99,99}

print(s)

print(type(s)) #it is of type set
```

```
{66, 99, 8, 9, 44, 20, 21}
<class 'set'>
```

In [28]:

```
t = (20,21,21,44,33,55,66,77,77)

print(t)

print(type(t)) #it is of type tuple
```

```
(20, 21, 21, 44, 33, 55, 66, 77, 77)
<class 'tuple'>
```

In [29]:

```
str = 'bhanu'
```

```
print(str)

print(type(str)) #it is of type string
```

```
bhanu
<class 'str'>
```

In [30]:

```
st = 'k'

print(st)

print(type(st)) #for a single character which would be 'char' type but in python it of string type for a single or 'n
```

```
k
<class 'str'>
```

In [40]:

```
range(0,10) #here we can get range of values from '0' to 10 values (if we want '10' values from '0' to '10' then we w

list(range(10)) #converting range into list
```

Out[40]:

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

In [42]:

```
range(2,10,2) #here we can get range of values from '2' to 10 values (if we want even values from '0' to '10' then we

list(range(2,10,2)) #converting range into list
```

Out[42]:

```
[2, 4, 6, 8]
```

In [43]:

```
range(2,10,2) #here we can get range of values from '2' to 10 values (if we want even values from '0' to '10' then we

list(range(2,10,2)) #converting range into list

print(type(range(2,10,2))) #it is of type range
```

```
<class 'range'>
```

In [53]:

```
x = {'bhanu':'googlepixel','kavya':'oneplus','navya':'iphone'}

print(x)

print(x.keys()) #if we have dictionary keys should be unique
print(x.values()) #for to fetch values

print(x['bhanu']) #we will get value by key

print(x.get('navya')) #one more way fetch a value by using (variable.get())
```

```
{'bhanu': 'googlepixel', 'kavya': 'oneplus', 'navya': 'iphone'}
dict_keys(['bhanu', 'kavya', 'navya'])
dict_values(['googlepixel', 'oneplus', 'iphone'])
googlepixel
iphone
```

In [ ]: