In [1]:
```
#@author: Bhanu Prakash

num=5

id(num) #address of num will be write it as "id"
```

Out[1]:
```
140720228870048
```

In [2]:
```
name='bhanu'

id(name) #address of name will be write it as "id"
```

Out[2]:
```
4387608240
```

In [24]:
```
a = 10

b = a

a

print(a)

b

print(b)

id(a)

print(id(a))

id(b) #when ever we create multiple variables in python if we have same date then we will get same address that why i

print(id(b))
```

```
10
10
140720228870208
140720228870208
```

In [19]:
```
a = 10
```

```
b = a

a

print(a)

b

print(b)

id(10) #address is based on the box it means object which is of data '10' and this is the one which assigned for 'a'

print(id(10))
```

```
10
10
140720228870208
```

In [20]:

```
a = 10

b = a

a

print(a)

b

print(b)

k = 10

id(k)

print(id(k)) #here we are tagging the variable 'k' for a data '10' So it will also allocated to same object it means
```

```
10
10
140720228870208
```

In [21]:

```
a = 10
```

```
b = a

a

print(a)

b

print(b)

id(a)

print(id(a))

id(b) #when ever we create multiple variables in python if we have same date then we will get same address that why i

print(id(b))

a = 11

id(a)

print(id(a)) #data has been changed so address also changes
```

```
10
10
140720228870208
140720228870208
140720228870240
```

In [22]:

```
a = 10

b = a

a

print(a)

b

print(b)

id(a)
```

```
print(id(a))

id(b) #when ever we create multiple variables in python if we have same date then we will get same address that why i

print(id(b))

a = 11

id(a)

print(id(a)) #data has been changed so address also changes

print(id(b)) #if we change the data for 'a' then also it won't effect the address of 'b' where as it was allocated to
```

```
10
10
140720228870208
140720228870208
140720228870240
140720228870208
```

In [23]:

```
a = 10

b = a

a

print(a)

b

print(b)

id(a)

print(id(a))

id(b) #when ever we create multiple variables in python if we have same date then we will get same address that why i

print(id(b))

a = 11
```

```
id(a)

print(id(a)) #data has been changed so address also changes

print(id(b)) #if we change the data for 'a' then also it won't effect the address of 'b' where as it was allocated to

k = a

id(k)

print(id(k)) #when ever we create multiple variables in python if we have same date then we will get same address tha
```

```
10
10
140720228870208
140720228870208
140720228870240
140720228870208
140720228870240
```

In [25]:
```
PI = 3.14 #in python we cannot make it constant so we will refer it as capital letters so it indicates that it should

print(PI)

print(type(PI)) #we can show the type of variable in python
```

```
3.14
<class 'float'>
```

In [ ]:
```

```