In [18]:
```
anmol = 100
#Run a Code using Shift + Enter
```

In [19]:
```
anmol
```

Out[19]: 100

In [20]:
```
anjli = 45.8
```

In [21]:
```
anjli
```

Out[21]: 45.8

In [22]:
```
book_name = "Top Visionaries"
```

In [23]:
```
book_name
```

Out[23]: 'Top Visionaries'

In [24]:
```
one_hp = 100
```

In [25]:
```
onehp = 100
```

- Python is a Dymnically Types Language

Statically Type -

int abc = 10

float xyz = 18.5

char name = "LetsUpgrade"

Dynamically Types Langauiage

abc = 1000

xyz = 10.4

name = "LetsUpgrade"

In [26]:
```
#Code

a = 1
b = 2

a+b
```

Out[26]: 3

Markdown for notes ->

Hey Guys there are our Notes

if abc == 2: pass

Heading tag for our notes and code

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Objects and Data Sctructures

Objects and Data Structres are different kind of containers in which we can store different kind of data, which can e used for our programming purpose

These are kind of different jars to store differnt things

Integer => Whole Numbers like 1,2,3,1000, 39001033

In [27]:
```
num1 = 1000

num2 = 100
```

In [28]:
```
type(num1)
```

Out[28]:
```
int
```

In [29]:
```
type(num2)
```

Out[29]:
```
int
```

In [30]:
```
print(num2)
```

```
100
```

Float => Decimal Numbers

12.3, 56.1936193, 67.7402645, 867.284629

In [31]:
```
num3 = 12.3
```

In [32]:
```
type(num3)
```

Out[32]:
```
float
```

String => Words and Sentences

"Saikiran" "LetsUpgrade" "Python" "Niren"

In [36]:
```
name1 = 'Saikiran'
```

In [37]:
```
name1
```

Out[37]:
```
'Saikiran'
```

In [38]:
```
type(name1)
```

Out[38]:
```
str
```

In [42]:
```
name1
```

Out[42]:
```
'Saikiran'
```

In [46]:
```
# Write . after the variable name and press tab key
# this is a example of using By Default function in python

name1.lower()
```

Out[46]:
```
'saikiran'
```

In [50]:
```
name1.count("i")
```

Out[50]:
```
2
```

List ?

- Milk 1 Liter
- Floor 1 Kg
- Diary Milk Silk 10

We need one container which can store collection of different data types inside it, and those data types are accessable

list

- 12
- 45.2
- "LetsUpgrade"
- sub list
- - 34
- - 45

In [54]:
```
lst = [12,45.2,"LetsUpgrade",[34,45]]
```

In [52]:
```
lst
```

Out[52]:
```
[12, 45.2, 'LetsUpgrade', [34, 45]]
```

In [53]:
```
type(lst)
```

Out[53]:
```
list
```

In [55]:
```
lst[2]
```

Out[55]:
```
'LetsUpgrade'
```

In [58]:
```
lst[3][1]
```

Out[58]:
```
45
```

In [59]:
```
lst.append("This is gettting added in the end")
```

In [60]:
```
lst
```

Out[60]:
```
[12, 45.2, 'LetsUpgrade', [34, 45], 'This is gettting added in the end']
```

In [61]:
```
lst.pop()
```

Out[61]:
```
'This is gettting added in the end'
```

In [62]:
```
lst
```

Out[62]:
```
[12, 45.2, 'LetsUpgrade', [34, 45]]
```

Dictionaries

- unordered Key Value pairs
- just like our normal dictionary have word and some value attached to that word,
- the same way we will have a key attachted to a value
- We will access the values using those key
- those keys will also act as our house numbers

In [63]:
```
dit = {"Key1":100, "key2":56,"key3":78}
```

In [70]:
```
dit["Key9"]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-70-6fd54d49ae28> in <module>
----> 1 dit["Key9"]

KeyError: 'Key9'
```

In [72]:
```
dit.get("Key1")
```

Out[72]:
```
100
```

In [73]:
```
dit_person_details = {"name":"sai","email":"sai@letsupgrade.in",
                      "mobile":"000000000","age":"5"}
```

In [74]:
```
dit_person_details.get("name")
```

Out[74]:
```
'sai'
```

In [75]:
```
dit_person_details.get("mobile")
```

Out[75]:
```
'000000000'
```

In [76]:
```
dit.keys()
```

Out[76]:
```
dict_keys(['Key1', 'key2', 'key3'])
```

In [77]:
```
dit.items()
```

Out[77]:
```
dict_items([('Key1', 100), ('key2', 56), ('key3', 78)])
```

In [78]:
```
dit.values()
```

Out[78]:
```
dict_values([100, 56, 78])
```

Sets

Unordered collection of unique objects,

- Remeber ur 8th - 12th class maths ?
- You had set in that ?
- which used to have all the unique values from all the collections
- the same way, python has a data type which stores all the uniques values from the given collection of data types
- Sets are been used for user auth, and while working with excel data to remove duplicates from the data

In [79]:
```
sts = {1,2,3,4,5,1,3,1,3,45,1,1,2,3,4,45,"sai","sai"}
```

In [80]:
```
sts
```

Out[80]:
```
{1, 2, 3, 4, 45, 5, 'sai'}
```

Tuples

Ordered immutable sequence of objects: (10,"hello",200.3)

A container which store a collection of different data types which onces delcared are never changed

these are immuntable in the programming world

- This kind of data type is been used for storing your private keys while developing python based api and softwares

In [81]:
```
tup = (1,2,3,4,4)
```

In [82]:
```
tup
```

Out[82]:
```
(1, 2, 3, 4, 4)
```

In [83]:
```
tup.count(4)
```

Out[83]:
```
2
```

tup.index(3)

In [85]:
```
tup
```

Out[85]:
```
(1, 2, 3, 4, 4)
```

Boolean

A Data Contrainer which stores -> True or False

Like switch which store 1 or 0

In [90]:
```
isClassThere = True
```

In [91]:
```
isClassThere
```

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

In [93]:
```
isClassThere
```

Out[93]:
```
True
```

In [94]:
```
lst.append(3)
```

In [95]:
```
lst.pop()
```

Out[95]:
```
3
```

In [96]:
```
lst.reverse()
```

In [97]:
```
lst.insert(4,1)
```

In [99]:
```
lst.clear()
```

In [ ]:
```

```