Data types

- in python type declare not required
- python will automatically takes the type based on provided value
- we have many data types available
- The main data types are
  - integer
  - float
  - Boolean
  - String
- List
- Dictionary
- tuple
- Complex

integer

In [1]:
```
n=100
type(n)
```

Out[1]:
```
int
```

- binary representation
- octa representation
- hexa represenation

In [ ]:
```
100000
100
200
```

Binary

- bi mean 2
- so it requires 2 digits
- Generally we have 0 1 2 3 4 5 6 7 8 9
- So it will represent with only 0 and 1

- representation is 0b10,0b01111, 0B111
- wrong representation is 0b102, 0b007

In [3]:
```
0b101
```

Out[3]: 5

In [4]:
```
0b1001
```

Out[4]: 9

In [5]:
```
0b101
```

Out[5]: 5

In [1]:
```
import os
```

In [2]:
```
os.getcwd()

# get current working directory
# It wil give the path
# where our python file located
```

Out[2]: 'C:\\Users\\omkar\\OneDrive\\Documents\\Data science\\Naresh IT\\Naresh IT\\Data science\\Batches\\Batch-9_June\\Python'

In [3]:
```
# Binary

0b1111
```

Out[3]: 15

In [ ]:
```
8  4  2   1    num
0   0  0   0     0
0   0  0   1     1
0   0  1   0     2
0   0  1   1     3
0   1  0   0     4
0   1  0   1     5
0   1  1   0     6
0   1  1   1     7
1   0  0   0     8
1   0  0   1     9
1   0  1   0     10
1   0  1   1     11
1   1  0   0     12
1   1  0   1     13
1   1  1   0     14
1   1  1   1     15

8  4   2   1
```

In [4]:
```
0b11101
```

Out[4]: 29

In [5]:
```
0b111011
```

Out[5]: 59

octal

- octa mean 8
- so it requires 8 digits
- Generally we have 0 1 2 3 4 5 6 7 8 9
- So it will represent with only 0 1 2 3 4 5 6 7
- representation is 0o123,0O012345, 0o7654321
- wrong representation is 0o108, 0o987

In [6]:
```
0o123
```

Out[6]: 83

In [7]:
```
0o1234
```

Out[7]: 668

hexa

- hexa mean 16
- so it requires 16 digits
- Generally we have 0 1 2 3 4 5 6 7 8 9 A(10) B(11) C(12) D(13) E(14) F(15)
- So it will represent with only 0-9 A-F
- representation is 0xF9,0X0123abc, 0xabc
- wrong representation is 0xGH, 0xABCDEFG

In [10]:
```
0xabc
```

Out[10]: 2748

In [9]:
```
0x123f  # 4671
```

Out[9]: 4671

float

In [12]:
```
n1=1005.00
type(n1)
```

Out[12]: float

In [13]:
```
n2=1005
type(n2)
```

Out[13]: int

In [17]:
```
1e1,1e2,1e3,1e4
# kep=k*10^p
```

Out[17]: (10.0, 100.0, 1000.0, 10000.0)

In [18]:
```
24e3
# 24*10^3
```

Out[18]: 24000.0

In [19]:
```
1e+1,1e+2,1e+3,1e+4
```

Out[19]: (10.0, 100.0, 1000.0, 10000.0)

In [20]:
```
24e-3  # 24/1000= 0.024
```

Out[20]: 0.024

- postive sign means multiply
- negative means divide

strings

- strings are used to represnt the english charcters generally
- strings will represent in
  - single quotes
  - double quotes
  - triple quotes

In [22]:
```
name='python'
type(name)
```

Out[22]: str

In [23]:
```
name1="omkar"
type(name1)
```

Out[23]: str

In [24]:
```
name2='10'
type(name2)
```

Out[24]: str

In [25]:
```
email_id='naresh123@gamil.com'
type(email_id)
```

Out[25]: str

RED COLOUR MEANS STRINGS

GREEN COLOUR MEANS KEYWORDS

BLACK COLOUR MEANS VARIABLES

In [26]:
```
name='python'
name1="python"
```

In [27]:
```
name
```

Out[27]: 'python'

In [28]:
```
name1
```

Out[28]: 'python'

- output always in single quotes

In [29]:
```
print(name)
```

```
python
```

- whenever you print quotes will not visible

In [32]:
```
name="i like 'python'"
print(name)
```

```
i like 'python'
```

In [33]:
```
str='i like "python"'
print(str)
```

```
i like "python"
```

In [34]:
```
name
```

Out[34]: "i like 'python'"

- Entire string in double quotes the highilted word in single quote vice versa

*__Triple quotes__

- Triple quotes is not used for coding
- Triple quotes is used to provide the information to the user
- In order to provide the information in jupyter notebook we will Markdown

- In vscode or pycharm there is no markdown option
- Thatswhy the world start used to write the information in triple quotes
- This entire process is called **Doc String**

In [35]:
```
"""
hello
im writing python code
this python
"""
```

Out[35]:
```
'\nhello\nim writing python code\nthis python\n'
```

Boolean

In [36]:
```
value=True
type(value)
```

Out[36]:
```
bool
```

In [37]:
```
value1=False
type(value1)
```

Out[37]:
```
bool
```

In [ ]:
```
true="True"
false=False

# true and false are variables
# type true is String
# type false is boolean
```