In [1]:
```
num=100
```

type

In [3]:
```
type(num)
```

Out[3]:
```
int
```

- integer
- float
- string
- Boolean
- complex
- list
- tuple
- dictionary
- set

In [ ]:
```
# Python basic concepts- Part1

- Variables

- Data types

- Data types conversion (type casting)

- print statements

- overview about packages

- Write your first end to end program

- Conditions (if-else)

- Functions

- loops (for-while)


# Python part-2

- strings

- list

- dictionary
```

```
- tuple (assesment)

- sets

- file handling

- lambda functions


# Python part-3: OOPS concepts
```

Integer

- In real time we will use decimal number system
- what ever the input but python will give output in decimal number foramt
- binary
  - binary includes only 0 and 1
  - bi means =2
  - representation is : 0b111, 0B101
  - wrong representation is : 0b123

In [5]:
```
num=111
type(num)
```

Out[5]: int

In [6]:
```
num1=0b111
type(num1)
```

Out[6]: int

In [ ]:
```
- decimal

- binary

- octa

- hexa
```

In [ ]:
```
mars : math binary

how old are you?
```

In [7]:
```
0b111
```

Out[7]: 7

In [8]:
```
0b1111
```

Out[8]: 15

In [9]:
```
0b101
```

Out[9]: 5

In [10]:
```
0b1110
```

Out[10]: 14

In [11]:
```
0b101010
```

Out[11]: 42

- octa
  - octa means =8
  - octa includes only 0 1 2 3 4 5 6 7
  - representation is : 0o123, 0O456
  - wrong representation is : 0o897

In [1]:
```
0o123
```

Out[1]: 83

In [3]:
```
0o567
```

Out[3]: 375

- hexa
  - hexa means =16
  - hexa includes only 0 1 2 3 4 5 6 7 8 9 A B C D E F
  - A=10 B=11 C=12 D=13 E= 14 F=15
  - representation is : 0x123, 0xabc
  - wrong representation is : 0xabz

In [5]:
```
0x123
```

Out[5]: 291

In [6]:
```
0xabc

# 256*10+ 16*11+1*12
```

Out[6]: 2748

In [ ]:
```
2^2    2^1    2^0
-----------------
4       2      1
-----------------
0       0      0   = 0
0       0      1   = 1
0       1      0   = 2
0       1      1   = 3
1       0      0   = 4
1       0      1   = 5
1       1      0   = 6
1       1      1   = 7
```

In [8]:
```
0b110
```

Out[8]: 6

In [9]:
```
0b101010101101
```

Out[9]: 2733

In [ ]:
```
2^3    2^2    2^1    2^0
    -----------------
 8   4       2      1
    -----------------
 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
```

float

In [10]:
```
number_float=123.56
type(number_float)
```

Out[10]: float

In [11]:
```
1e1
```

Out[11]: 10.0

In [12]:
```
1e2
```

Out[12]: 100.0

In [13]:
```
1e3
```

Out[13]: 1000.0

In [14]:
```
1e1,1e2,1e3,1e3
```

Out[14]: (10.0, 100.0, 1000.0, 1000.0)

In [15]:
```
1e+1,1e+2,1e+3,1e+3
```

Out[15]: (10.0, 100.0, 1000.0, 1000.0)

In [17]:
```
1e-1,1e-2,1e-3
# 1/10, 1/100,1000
```

Out[17]: (0.1, 0.01, 0.001)

In [ ]:
```
# You need to check some answer will be zero
# 0.0
# 1e-13
7.74619736e-15
7.74619736/100000000000000000
0.000000000000007746197 ~=0
```

In [ ]:
```
1e-10
1/(10000000000000)= 0.0000000000001
```

Boolean

- True = 1
- False = 0

In [18]:
```
ans=True
type(ans)
```

Out[18]: bool

In [19]:
```
wrong_ans=False
type(wrong_ans)
```

Out[19]: bool

In [20]:
```
true=True
true
```

Out[20]: True

In [ ]:
```
Omkar can we give wrong_ans=False and wrong_ans=5
```

In [21]:
```
wrong_ans=False
wrong_ans=5
wrong_ans
```

Out[21]: 5

In [22]:
```
type(wrong_ans)
# Data type also change
```

Out[22]:
```
int
```

String

string

- string

In [23]:
```
name1='python'
name1
```

Out[23]:
```
'python'
```

In [24]:
```
name2="naresh it"
name2
```

Out[24]:
```
'naresh it'
```

In [25]:
```
type(name1)
```

Out[25]:
```
str
```

In [ ]:
```
integer ==== int
boolean ==== bool
string  ==== str
float   ==== float
```

In [ ]:
```
i like 'python'
```

In [27]:
```
name3='i "like" python'
name3
```

Out[27]:
```
'i "like" python'
```

In [28]:
```
name3="i 'like' python"
name3
```

Out[28]:
```
"i 'like' python"
```

Note

- Entire sentence in single quotes, then highlited word in double quotes
- Entire sentence in double quotes, then highlited word in single quotes

In [35]:
```
str1="""hello how are you
        how do you do
        im good
        how is DS class"""
str1
# \n = new line
# there is multiple lines are there in output
```

Out[35]: 'hello how are you\n   how do you do\n   im good\n   how is DS c
lass'

In [36]:
```
print(str1)
```

```
hello how are you
        how do you do
        im good
        how is DS class
```

- strings can represent in single quotes or double quotes or triple quotes
- When a person writing something in triple quotes : **Doc String**
- Triple quotes are used for conveying information like markdown in Jupyter notebook
- Triple quotes are not used for coding part

In [ ]:
```
Type casting
```