Float data Type

```
In [3]: a=36e3
        print(a)
        print(type(a))
```

```
36000.0
<class 'float'>
```

sequence

1) list

- To Store Multiple items in a single variable
- Ordered
- changeable and allow duplicate values indexed
- Square bracket

```
In [4]: l=["Apple",'Banna']
        print (l)
        print (type(l))
```

```
['Apple', 'Banna']
<class 'list'>
```

2) Tuple

- To Store Multiple items in a single variable
- Ordered
- Unchangeable and allow duplicate values indexed
- Round bracket

```
In [5]: t=("Apple",'Banna',"Apple","apple")
        print (t)
        print (type(t))
```

```
('Apple', 'Banna', 'Apple', 'apple')
<class 'tuple'>
```

3) Range

- range(0,5,2): (Starting point , end (not included) , step)
- in range(0,5,-2): no Output , not an error

```
In [6]: x=range(5)
        print (x)
        print(type(x))
        for i in range(5):
         print(i)
```

```
range(0, 5)
<class 'range'>
0
1
2
```

```
3
4
```

Mapping Type

- Ordered
- Changeable
- Does not allow duplicate
- Key : Value
- Curly Bracket

In [7]:
```
d={1:"A",2:"B",3:"C",4:"D", 4:"F"}
print(d)
print(type(d))
print(d[3])
print(type(d[3]))
```

```
{1: 'A', 2: 'B', 3: 'C', 4: 'F'}
<class 'dict'>
C
<class 'str'>
```

Set Type

1) Set

- UnOrdered
- Unindexed
- Does not allow duplicate
- Curly Bracket

In [8]:
```
d={"A","B","C","D","F"}
print(d)
print(type(d))
print(d[3])
```

```
{'C', 'D', 'B', 'F', 'A'}
<class 'set'>
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-8-637efac07743> in <module>
      2 print(d)
      3 print(type(d))
----> 4 print(d[3])

TypeError: 'set' object is not subscriptable
```

2) Frozen Set

Boolean

bool

In [9]:
```
print(bool(1))
```
```
True
```

In [10]:
```
print(bool(0))
```
```
False
```

In [11]:
```
print(bool(25>6))
```
```
True
```

In [12]:
```
print(bool(6>25))
```
```
False
```

In [13]:
```
print(bool(" "))
```
```
True
```

In [14]:
```
print(bool(""))
```
```
False
```

In [15]:
```
print(bool(" "))
```
```
True
```

Global variable VS Local variable

In [16]:
```
a='python'
def test():
    a='java'
    print(a)

test()
print (a)
```
```
java
python
```

In [17]:
```
a='python'
def test():
    global a
    a='java'
    print(a)

test()
print (a)
```
```
java
java
```

- Comments

In [18]:
```
# this is a comments
a=100
b=200
print(a+b) #total
```

```
'''' multiline '''
print("Hello")
```

```
300
Hello
```

- Reading input from user

In [ ]:
```
user=input("Enter UserName ")
print("The name is :",user)
```

Typecasting

1) int

In [22]:
```
print(int(123.789))
print(int(True))
print(int(False))
print(int("100"))
```

```
123
1
0
100
```

In [23]:
```
# all error
print(int(0B111))
print(int("110.5")) # 2) float ma allow karse
print(int("ten"))
print(int("0B111"))
```

```
7
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-23-01826b40655e> in <module>
      1 # all error
      2 print(int(0B111))
----> 3 print(int("110.5"))
      4 print(int("ten"))
      5 print(int("0B111"))

ValueError: invalid literal for int() with base 10: '110.5'
```

3) boolean

In [31]:
```
print(bool(0))
print(bool(1))
print(bool(10.5))
```

```
False
True
True
```

In [32]:
```
print(bool(0.625))
print(bool(" "))
print(bool("abd"))
```

```
True
True
True
```

In [33]:
```
print(bool("True"))
print(bool("False"))
print(bool(""))
```

```
True
True
False
```

4) String

In [37]:
```
print(str(10))
print(str(10.5))
print(str(True))
print(str(oyyy))
```

```
10
10.5
True
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-37-a730287217c5> in <module>
      2 print(str(10.5))
      3 print(str(True))
----> 4 print(str(oyyy))

NameError: name 'oyyy' is not defined
```

Python Operation

1) Arithmetic Operation

- +add
- -sub
- *multiply
- / divide **always in float**
- % modulo
- // floor division
- ** Exponensial

In [46]:
```
a=int(input("Enter a number 1: "))
b=int(input("Enter a number 2: "))
print("Addition : ",a+b)
print("subtraction : ",a-b)
print("multiplication : ",a*b)
print("division : ",a/b)
print("modulo : ",a%b)
print("floor : ",a//b)
print("power : ",a**b)
```

```
Enter a number 1: 4
Enter a number 2: 2
Addition :  6
subtraction :  2
multiplication :  8
division :  2.0
modulo :  0
```

```
floor :  2
power :  16
```

In [47]:
```
"dixit"+38
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-47-d03906dc5f44> in <module>
----> 1 "dixit"+38

TypeError: can only concatenate str (not "int") to str
```

In [50]:
```
"dixit "+"38"
```

Out[50]:
```
'dixit38'
```

In [61]:
```
"abc" * "xyz"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-61-1973b9071316> in <module>
----> 1 "abc" * "xyz"

TypeError: can't multiply sequence by non-int of type 'str'
```

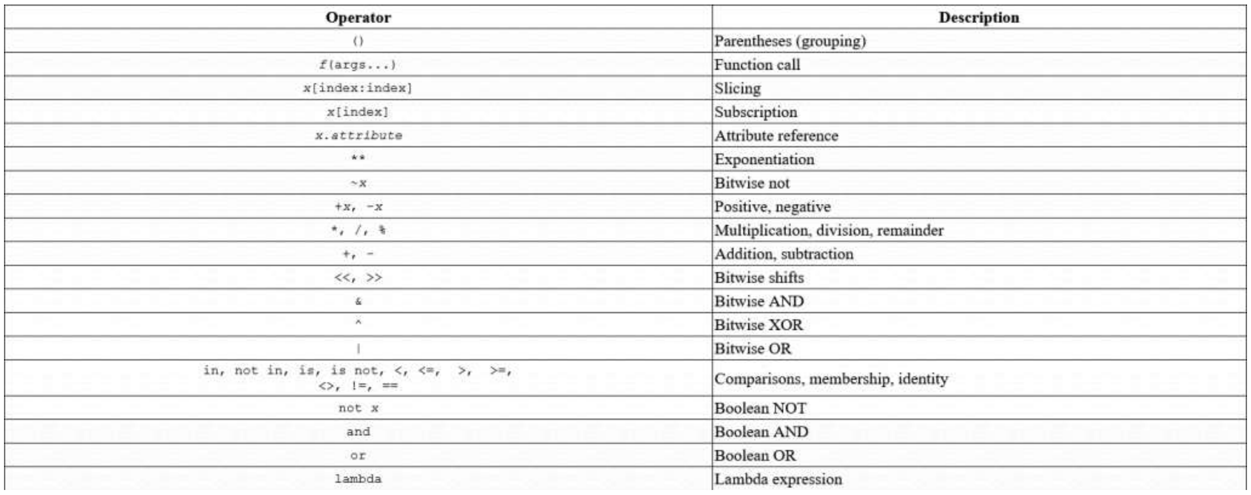
In [53]:
```
"Patel , "* 5
```

Out[53]:
```
'Patel , Patel , Patel , Patel , Patel , '
```

In [58]:
```
12.9//5 # if any one no is float then the ans will be in float
```

Out[58]:
```
2.0
```

operator precedence in python



Operator	Description
()	Parentheses (grouping)
f(args...)	Function call
x[index:index]	Slicing
x[index]	Subscription
x.attribute	Attribute reference
**	Exponentiation
~x	Bitwise not
+x, -x	Positive, negative
*, /, %	Multiplication, division, remainder
+, -	Addition, subtraction
<<, >>	Bitwise shifts
&	Bitwise AND
^	Bitwise XOR
|	Bitwise OR
in, not in, is, is not, <, <=, >, >=, <>, !=, ==	Comparisons, membership, identity
not x	Boolean NOT
and	Boolean AND
or	Boolean OR
lambda	Lambda expression

In [65]:
```
print((6+4) * (2+8))
print(6+4*2+8)
```

```
100
22
```

In [66]:
```
3**2*2**3
```
Out[66]:
```
72
```
In [ ]: