type casting

- we can type cast from all other data type to integer except complex & text string

```
In [1]:  bool(2)

Out[1]:  True
```

```
In [3]:  bool(0)

Out[3]:  False
```

```
In [5]:  bool(3.2)

Out[5]:  True
```

```
In [7]:  bool(1+2j)

Out[7]:  True
```

```
In [9]:  bool(0+0j)

Out[9]:  False
```

```
In [ ]:
```

```
In [11]:  str(7)

Out[11]:  '7'
```

```
In [13]:  str(3.2)

Out[13]:  '3.2'
```

```
In [15]:  str(True)

Out[15]:  'True'
```

type casting is completed

PYTHON OPERATOR

- arithematic operator (+,-,*,/,//)
- relational (-,>,<,>=,<=, ==, !=
- uniary -one operator (-)
- assignment =, +=,-=, /=, //=, *=
- logical = and ,or, not

arithmatic operator

(+ , - , * , /, //, **)

```
In [3]: x1,y1 =5,10
```

```
In [5]: x1+ y1
```

```
Out[5]: 15
```

```
In [7]: x1-y1
```

```
Out[7]: -5
```

```
In [9]: x1/y1
```

```
Out[9]: 0.5
```

```
In [11]: x1//y1
```

```
Out[11]: 0
```

```
In [13]: x1**y1
```

```
Out[13]: 9765625
```

assignment operator

```
In [37]: x = 2
         x
```

```
Out[37]: 2
```

```
In [65]: x = x+2
         x
```

```
Out[65]: 66
```

```
In [45]: x+=4
         x
```

```
Out[45]: 10
```

```
In [49]: x-=2
         x
```

```
Out[49]: 6
```

```
In [53]: x*=3
         x
```

Out[53]:
```
54
```

In [69]:
```
x/= 2
x
```

Out[69]:
```
33.0
```

In [71]:
```
x//=2
x
```

Out[71]:
```
16.0
```

unary operator

In [84]:
```
n = 7 #negattion
n
```

Out[84]:
```
7
```

In [92]:
```
m =-(n)
m
```

Out[92]:
```
-7
```

In [94]:
```
n
```

Out[94]:
```
7
```

In [96]:
```
-n
```

Out[96]:
```
-7
```

Relational operator

In [99]:
```
r1 =5
r2 = 6
```

In [101…
```
r1> r2
```

Out[101…
```
False
```

In [103…
```
r1<r2
```

Out[103…
```
True
```

In [105…
```
r1 ==r2
```

Out[105…
```
False
```

In [107…
```
r1!=r2
```

Out[107…
```
True
```

In [109…

```
r1
```

Out[109…

```
5
```

In [111…

```
r2
```

Out[111…

```
6
```

In [119…

```
r3 = 6
r3
```

Out[119…

```
6
```

In [121…

```
r1==r3
```

Out[121…

```
False
```

In [123…

```
r2==r3
```

Out[123…

```
True
```

In [125…

```
print(r1)
print(r2)
print(r3)
```

```
5
6
6
```

In [127…

```
r3<=r2
```

Out[127…

```
True
```

In [129…

```
r2>=r3
```

Out[129…

```
True
```

logical operator

In [132…

```
 x = False
x
```

Out[132…

```
False
```

In [134…

```
not x
```

Out[134…

```
True
```

In [136…

```
y =True
y
```

Out[136…

```
True
```

In [138…

```
not y
```

```
Out[138… False
```

```
In [ ]:
```