Bitwise operator -Compliment -And -Or -Xor -Left shift -Right Shift

In [2]:
```
~12# Compliment
```

Out[2]: -13

In [3]:
```
12&13 #1&1=1 rest of 0
```

Out[3]: 12

In [4]:
```
12|13  #1|0=1,0|1=1,1|1=1,0|0=0
```

Out[4]: 13

In [5]:
```
12^13
```

Out[5]: 1

In [6]:
```
10<<1
```

Out[6]: 20

In [7]:
```
10>>1
```

Out[7]: 5

In [8]:
```
10>>2
```

Out[8]: 2

Math Function

In [10]:
```
import math
```

In [11]:
```
x=math.sqrt(25)
int(x)
```

Out[11]: 5

In [12]:
```
x=math.floor(2.5)
int(x)
```

Out[12]: 2

In [13]:
```
x=math.ceil(2.5)
int(x)
```

Out[13]: 3

In [14]:
```
x=math.e
x
```

Out[14]: 2.718281828459045

In [15]:
```
x=math.pi
x
```

Out[15]: 3.141592653589793

Math Function

In [17]:
```
from math import *
print(round(pow(2,3)))
print(sqrt(25))
```

```
8
5.0
```

In [18]:
```
x=input()
x
y=input()
y
print(x+y)
```

```
54
```

In [19]:
```
type(x)
```

Out[19]: str

In [20]:
```
x2=int(input("Enter a number 1: "))
y2=int(input("Enter a number 1: "))
print(x2+y2)
```

```
30
```

In [21]:
```
st=input("Enter a String")
st
```

Out[21]: 'Arabinda'

In [22]:
```
st=input("Enter a String")[0]
st
```

Out[22]: 'A'

In [23]:
```
st=input("Enter a String")[0:3]
st
```

Out[23]: 'Ara'

In [24]:
```
st=eval(input("Enater a expression"))
st
```

Out[24]:
```
20
```

In [47]:
```
ch=input("Enter character ")
ch
```

Out[47]:
```
'A'
```

In [49]:
```
ch=input("Enter character ")
ch
```

Out[49]:
```
'Arabinda'
```

In [51]:
```
ch[-1]  #Backword slicing
```

Out[51]:
```
'a'
```

In [ ]:
```

```