

Values or Data Types : int,float,string,complex, Boolean

```
In [1]: i = 25  
i
```

```
Out[1]: 25
```

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

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

```
In [5]: print(type(i)) #inbuild class
```

```
<class 'int'>
```

```
In [7]: electricitybill=1450.67  
electricitybill
```

```
Out[7]: 1450.67
```

```
In [9]: type(electricitybill)
```

```
Out[9]: float
```

```
In [11]: color = True  
color
```

```
Out[11]: True
```

```
In [13]: type(color)
```

```
Out[13]: bool
```

```
In [15]: True + False
```

```
Out[15]: 1
```

```
In [17]: True-True
```

```
Out[17]: 0
```

```
In [19]: False+True
```

```
Out[19]: 1
```

```
In [21]: False-False
```

```
Out[21]: 0
```

```
In [23]: True*False
```

Out [23]: 0

In [25]: False/True

Out [25]: 0.0

In [27]: True/False

```
-----  
ZeroDivisionError                                Traceback (most recent call last)  
Cell In[27], line 1  
----> 1 True/False  
  
ZeroDivisionError: division by zero
```

In [29]: False//True

Out [29]: 0

Complex = a+bj ( a is real part, b is imaginary part & j is squareroot of -1)

In [31]: c1= 20+30j  
c1

Out [31]: (20+30j)

In [33]: type(c1)

Out [33]: complex

In [35]: c1.real # . is compulsory

Out [35]: 20.0

In [37]: c1.imag

Out [37]: 30.0

In [39]: c2=10+40j  
c2

Out [39]: (10+40j)

In [41]: c1+c2

Out [41]: (30+70j)

In [43]: c1-c2

Out [43]: (10-10j)

In [45]: c2-c1

Out[45]:  $(-10+10j)$

```
In [47]: print(c1)
         print(c2)
```

$(20+30j)$   
 $(10+40j)$

```
In [49]: c1*c2
```

Out[49]:  $(-1000+1100j)$

```
In [51]: b='bhavani'
         b
```

Out[51]: 'bhavani'

```
In [53]: b1="bowlu"
         b1
```

Out[53]: 'bowlu'

```
In [55]: b2='' extended name
         in another line'' #multiline string used in NLP
         b2
```

Out[55]: ' extended name \nin another line'

Slicing [:]

```
In [1]: name='powerrangers'
         name
```

Out[1]: 'powerrangers'

```
In [4]: name[2:7]
```

Out[4]: 'werra'

```
In [6]: name[10]
```

Out[6]: 'r'