

```
In [3]: #python numbers
x = 99
y = 2.356
z = 7j
print(type(x))
print(type(y))
print(type(z))

<class 'int'>
<class 'float'>
<class 'complex'>
```

```
In [ ]: #int numbers
a = 22
b = 45
c = 65
d = 23
e = 77
f = 62
g = 147
h = 25
i = 79
j = 485
```

```
In [ ]: #float numbers
aa = 2.09
bb = 3.89
cc = 9.8989
dd = 44.89
ee = 2390.666
ff = 75.6
gg = 300.9
hh = 12.8999
ii = 3.6666666
jj = 5.503
```

```
In [ ]: #complex numbers
_a = 2+1j
_b = 4+67j
ab = 2+6.89j
bc = 89j
cd = 3j
de = 6+29.7878j
ef = 2.08+1j
fg = 4+4.22222j
gh = 893421+47j
hi = 72+333j
```

```
In [9]: #type conversion
#int to float
a1 = 3
b1 = float(a1)
print(b1)
a2 = 45
b2 = float(a2)
print(b2)
a3 = 78
b3 = float(a3)
print(b3)
a4 = 9045
b4 = float(a4)
print(b4)
a5 = 748781
b5 = float(a5)
print(b5)
a6 = 23
b6 = float(a6)
print(b6)
a7 = 46989
b7 = float(a7)
print(b7)
a8 = 134789
b8 = float(a8)
print(b8)
a9 = 10000
b9 = float(a9)
print(b9)
a10 = 6740146002
b10 = float(a10)
print(b10)
```

```
3.0
45.0
78.0
9045.0
748781.0
23.0
46989.0
134789.0
10000.0
6740146002.0
```

```
In [11]: #float to int
c1 = 3.99
d1 = int(c1)
print(d1)
c2 = 56.9779
d2 = int(c2)
print(d2)
c3 = 45.98
d3 = int(c3)
print(d3)
c4 = 9.56
d4 = int(c4)
print(d4)
c5 = 2.67
d5 = int(c5)
print(d5)
c6 = 3.9969999
d6 = int(c6)
print(d6)
c7 = 1.7890686
d7 = int(c7)
print(d7)
```

```
3
56
45
9
2
3
1
```

```
In [5]: #int to complex
e1 = 15
f1 = complex(e1)
print(f1)
e2 = 5.90
f2 = complex(e2)
print(f2)
e3 = 9.89999
f3 = complex(e3)
print(f3)
e4 = -0.555
f4 = complex(e4)
print(f4)
e5 = -7898
f5 = complex(e5)
print(f5)
e6 = 65.9090233
f6 = complex(e6)
print(f6)
```

```
(15+0j)
(5.9+0j)
(9.89999+0j)
(-0.555+0j)
(-7898+0j)
(65.9090233+0j)
```

```
In [11]: #complex to int or float
x = 12j
c = float(x)
print(c)
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[11], line 3
      1 #complex to int or float
      2 x = 12j
----> 3 c = float(x)
      4 print(c)

TypeError: float() argument must be a string or a real number, not 'complex'
```

```
In [13]: #python booleans
print(6<9)
print(12==3)
print(4!=2)
print(2==2)
print(46>34)
```

```
True
False
True
True
True
```

```
In [15]: #Evaluate values & variables
print(bool(0))
print(bool(34))
print(bool(3.67))
print(bool(0.45))
```

```
False
True
```

