

Type Casting ¶

- integer
- float
- string
- boolean
- complex
- int to float, int to str, int to bool, int to complex
- it is called type casting

```
In [1]: number = 10  
type(number)
```

Out[1]: int

```
In [2]: #int to float  
float(number)  
#int to string  
str(number)  
#int to bool  
bool(number)  
#int to complex  
complex(number)
```

Out[2]: (10+0j)

```
In [3]: #int to float  
float(number)
```

Out[3]: 10.0

```
In [4]: #int to string  
str(number)
```

Out[4]: '10'

```
In [5]: bool(number)  
#int to bool
```

Out[5]: True

```
In [6]: #int to complex  
complex(number)
```

Out[6]: (10+0j)

```
In [7]: print(float(-10))
        print(str(-10))
        print(bool(-10))
        print(complex(-10))
```

```
-10.0
-10
True
(-10+0j)
```

```
In [8]: print(bool(0))
```

```
False
```

- For zero value boolean conversion will give False
- Remaining all are true

```
In [9]: complex(20),complex(20,50)
        #take your cursor inside the bracket apply shift +tab at the same time
```

```
Out[9]: ((20+0j), (20+50j))
```

```
In [11]: import random
         random.randint()
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[11], line 2
      1 import random
----> 2 random.randint()

TypeError: Random.randint() missing 2 required positional arguments: 'a' and 'b'
```

```
In [12]: complex()
```

```
Out[12]: 0j
```

```
In [ ]: #Float to other
```

```
In [13]: print(int(10.5)) #10
         print(str(10.5)) #'10.5'
         print(bool(10.5)) #true
         print(complex(10.5)) #10.5+0j
```

```
10
10.5
True
(10.5+0j)
```

```
In [14]: #String to other
print(int('python')) #error
print(float('python')) #'error'
print(bool('python')) #true
print(complex('python')) #error
```

ValueError Traceback (most recent call last)

Cell In[14], line 2

```
1 #String to other
----> 2 print(int('python')) #error
      3 print(float('python')) #'error'
      4 print(bool('python')) #true
```

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

```
In [15]: #String to other
print(int('10')) #error
print(float('10')) #'error'
print(bool('10')) #true
print(complex('10')) #error
```

```
10
10.0
True
(10+0j)
```

```
In [ ]: - '10'=== can be converted to int to float but '10.5' cannot be converted to f
float is the boss, integer conversion of float value having quotes
```