

# Casting: Type Conversion and Type Casting

**In Python, we can convert one type of variable to another type. This conversion is called type casting or type conversion.**

In casting, we convert variables declared in specific data types to the different data types.

Python performs the following two types of casting.

1. Implicit casting: The Python interpreter automatically performs an implicit Type conversion, which avoids loss of data.
2. Explicit casting: The explicit type conversion is performed by the user using built-in functions.

To perform a type casting, we are going to use the following built-in functions

1. `int()`: convert any type variable to the integer type.
2. `float()`: convert any type variable to the float type.
3. `complex()`: convert any type variable to the complex type.
4. `bool()`: convert any type variable to the bool type.
5. `str()`: convert any type variable to the string type.

In type casting, data loss may occur because we enforce the object to a specific data type.

## Int type conversion

- In int type conversion, we use the `int()` function to convert variables of other types to int type. Variable can be of any type such as float, string, bool.

While performing int type conversion, we need to remember the following points.

- When converting string type to int type, a string must contain integral value only and should be base-10.
- We can convert any type to int type, but we cannot perform complex to int type.

## Casting float value to an integer

```
In [1]: 1 pi = 3.14 # float number
        2 print(type(pi))
        3
        4 # converting float integer
        5 num = int(pi)
        6 print("Integer number:", num)
        7 print(type(num))
        8
```

```
<class 'float'>
Integer number: 3
<class 'int'>
```

## Casting Boolean value to an integer

```
In [2]: 1 flag_true = True
        2 flag_false = False
        3 print(type(flag_true))
        4
        5 # converting boolean to integer
        6 num1 = int(flag_true)
        7 num2 = int(flag_false)
        8
        9 print("Integer number 1:", num1)
       10 # Output 1
       11 print(type(num1))
       12 # Output class 'int'
       13
       14 print("Integer number 2:", num2)
       15 print(type(num2))
       16
```

```
<class 'bool'>
Integer number 1: 1
<class 'int'>
Integer number 2: 0
<class 'int'>
```

## Casting a string to an integer

In [3]:

```
1 string_num = "225"
2 print(type(string_num))
3
4 # converting str to integer
5 num1 = int(string_num)
6
7 print("Integer number 1:", num1)
8 print(type(num1))
9
```

```
<class 'str'>
Integer number 1: 225
<class 'int'>
```

When converting string type to int type, a string must contain integral value only and should be base-10. If you try to convert

In [4]:

```
1 string_num = 'Score is 25'
2 print(type(string_num))
3
4 # ValueError: invalid literal for int() with base 10: 'Score is 25'
5 num = int(string_num)
6 print(num)
```

```
<class 'str'>
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-4-6ed95d7f5781> in <module>
      3
      4 # ValueError: invalid literal for int() with base 10: 'Score is 25'
----> 5 num = int(string_num)
      6 print(num)
```

```
ValueError: invalid literal for int() with base 10: 'Score is 25'
```

## Float type conversion

In float type conversion we use a built-in function `float()`. This function converts variables of other types to float types.

### Casting integer to float

In [5]:

```
1 num = 725
2 print(type(num))
3
4 # converting float to integer
5 num1 = float(num)
6
7 print("Float number:", num1)
8 print(type(num1))
9
```

```
<class 'int'>
Float number: 725.0
<class 'float'>
```

### Casting Boolean to float

```
In [6]: 1 flag_true = True
2 flag_false = False
3 print(type(flag_true))
4
5 # converting boolean to float
6 num1 = float(flag_true)
7 num2 = float(flag_false)
8
9 print("Float number 1:", num1)
10 print(type(num1))
11
12 print("Float number 2:", num2)
13 print(type(num2))
14
```

```
<class 'bool'>
Float number 1: 1.0
<class 'float'>
Float number 2: 0.0
<class 'float'>
```

## Casting string to float

```
In [7]: 1 string_num = "725.535"
2 print(type(string_num))
3
4 # converting str to float
5 num1 = float(string_num)
6
7 print("Float number:", num1)
8 print(type(num1))
9
```

```
<class 'str'>
Float number: 725.535
<class 'float'>
```

While performing float type conversion, we need to remember some points.

- We can convert any type to float type, but we cannot cast complex to float type.
- While converting string type to float type, a string must contain an integer/decimal value of base-10.

## Complex type conversion

In complex type conversion, we use the built-in function `complex()` to convert values from other types to the complex type. Value can be any type including of int, float, bool, str.

The complex function has the following two forms for conversion.

- `complex(x)`: To convert a value x into a complex type. In this form, the real value is x, and the imaginary value is 0.
- `complex(x, y)`: To convert the value x and y into a complex type. In this form, the real value is x, and the imaginary is y.

## Casting integer type to complex type

```
In [8]: 1 r_num = 135
        2 print(type(r_num)) # class 'int'
        3
        4 # converting int to complex(x)
        5 c_num = complex(r_num)
        6
        7 print("Complex number:", c_num)
        8 print(type(c_num))
        9
       10 # converting int to complex(x, y)
       11 r_num, i_num2 = 135, 235
       12 c_num = complex(r_num, i_num2)
       13
       14 print("Complex number:", c_num)
       15 print(type(c_num)) # class 'complex'
```

```
<class 'int'>
Complex number: (135+0j)
<class 'complex'>
Complex number: (135+235j)
<class 'complex'>
```

## Casting float type to complex type

In [9]:

```
1 r_num = 53.250
2 print(type(r_num)) # class 'float'
3
4 # converting float to complex(x)
5 c_num = complex(r_num)
6
7 print("Complex number:", c_num)
8 print(type(c_num))
9
10 # converting float to complex(x, y)
11 r_num, i_num2 = 53.250, 350.750
12 c_num = complex(r_num, i_num2)
13
14 print("Complex number:", c_num)
15 print(type(c_num))
16
```

```
<class 'float'>
Complex number: (53.25+0j)
<class 'complex'>
Complex number: (53.25+350.75j)
<class 'complex'>
```

## Casting Boolean type to complex type

```
In [10]: 1 boolean_true = True
2 print(type(boolean_true)) # class 'bool'
3
4 # converting boolean to complex(x)
5 c_num = complex(boolean_true)
6
7 print("Complex number:", c_num)
8 # Output (1+0j)
9 print(type(c_num))
10 # class 'complex'
11
12 # converting boolean to complex(x, y)
13 r_bool, i_bool = False, True
14 c_num = complex(r_bool, i_bool)
15
16 print("Complex number:", c_num)
17 # Output 1j
18 print(type(c_num))
19 # class 'complex'
```

```
<class 'bool'>
Complex number: (1+0j)
<class 'complex'>
Complex number: 1j
<class 'complex'>
```

## Bool type conversion

We use the built-in function `bool()` to convert values of other types to bool types. This function returns two values, either True or False.

We can convert any type of values to bool type, and the output for all values will be True, Except 0, which is False. If you convert an empty string to a boolean it will be converted to boolean False.

The bool True is 1 and False is 0. Every non-zero value is treated as True.

## Casting integer to Boolean type



```
In [11]: 1 num1 = 10
          2 num2 = 0
          3 print(type(num1)) # class 'int'
          4
          5 # Convert into to bool
          6 b1 = bool(num1)
          7 b2 = bool(num2)
          8
          9 print(b1)
         10 print(b2)
         11
         12 print(type(b1))
         13
```

```
<class 'int'>
True
False
<class 'bool'>
```

## Casting float to Boolean type

In [12]:

```
1 f_num1 = 25.35
2 f_num2 = 0.0
3 print(type(f_num1)) # class 'float'
4
5 # Convert float into to bool
6 b1 = bool(f_num1)
7 b2 = bool(f_num2)
8
9 print(b1)
10
11 print(b2)
12
13 print(type(b1))
14
```

&lt;class 'float'&gt;

True

False

&lt;class 'bool'&gt;

## Casting string to Boolean type

```
In [13]: 1 s1 = "False"
2 s2 = "True"
3 s3 = "812"
4 s4 = ""
5 print(type(s1)) # class 'str'
6
7 # Convert string into to bool
8 b1 = bool(s1)
9 b2 = bool(s2)
10 b3 = bool(s3)
11 b4 = bool(s4)
12
13 print(b1) # True
14 print(b2) # True
15 print(b3) # True
16 print(b4) # False
17 print(type(b1)) # class 'bool'
```

```
<class 'str'>
```

```
True
```

```
True
```

```
True
```

```
False
```

```
<class 'bool'>
```

## Casting complex type to Boolean type

```
In [14]: 1 c1 = 33 + 9j
2 c2 = 0 + 0j
3 print(type(c1)) # class 'complex'
4
5 # Convert complex value into to bool
6 b1 = bool(c1)
7 b2 = bool(c2)
8
9 print(b1) # True
10 print(b2) # False
11 print(type(b1)) # class 'bool'
```

```
<class 'complex'>
True
False
<class 'bool'>
```

## String type conversion

In str type conversion, we use the built-in function `str()` to convert converts variables of other types to a string type. This function returns the string type of object (value).

## Casting int to str type

```
In [15]: 1 num = 15
2 print(type(num)) # class 'int'
3
4 # converting int to str type
5 s1 = str(num)
6 print(s1)
7 print(type(s1))
8 # Output class 'str'
```

```
<class 'int'>
15
<class 'str'>
```

## Casting float type to str type

In [16]:

```
1 num = 75.35
2 print(type(num)) # class 'float'
3
4 # converting float to str type
5 s1 = str(num)
6 print(s1)
7 print(type(s1))
8
```

```
<class 'float'>
75.35
<class 'str'>
```

## Casting complex type to str type

In [17]:

```
1 complex_num = 15 + 5j
2 print(type(complex_num)) # class 'complex'
3
4 # converting complex to str type
5 s1 = str(complex_num)
6 print(s1)
7
8 print(type(s1))
9 # class 'str'
```

```
<class 'complex'>
(15+5j)
<class 'str'>
```

## Casting bool type to str type

```
In [18]: 1 b1 = True
          2 b2 = False
          3 print(type(b1)) # class 'bool'
          4
          5 # converting bool to str type
          6 s1 = str(b1)
          7 s2 = str(b2)
          8 print(s1)
          9 print(s2)
         10 print(type(s1)) # class 'str'
```

```
<class 'bool'>
True
False
<class 'str'>
```

```
In [ ]:
```

```
1
```