Python 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.

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

o performs 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.

1. When converting **string type** to **int type**, a string must contain integral value only and should be base-10.

2. We can convert any type to `int` type, but we cannot perform **complex** to int type.

Casting float value to an integer

```
pi = 3.14  # float number
print(type(pi))
# Output class 'float'

# converting float integer
num = int(pi)
print("Integer number:", num)
# Output  3
print(type(num))
# Output class 'int'
```

Casting Boolean value to an integer

```
flag_true = True
flag_false = False
print(type(flag_true))
# Output class 'bool'

# converting boolean to integer
num1 = int(flag_true)
num2 = int(flag_false)

print("Integer number 1:", num1)
# Output 1
print(type(num1))
# Output class 'int'

print("Integer number 2:", num2)
# 0
print(type(num2))
# class 'int'
```

Casting a string to an integer

```
string_num = "225"
print(type(string_num))
# Output class 'str'

# converting str to integer
```

```
num1 = int(string_num)

print("Integer number 1:", num1)
# Output 225
print(type(num1))
# Output 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

Example

```
string_num = 'Score is 25'
print(type(string_num))
# Output class 'str'

# ValueError: invalid literal for int() with base 10: 'Score is 25'
num = int(string_num)
print(num)
```

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

```
num = 725
print(type(num))
# Output class 'int'

# converting float to integer
num1 = float(num)

print("Float number:", num1)
# Output 725.0
print(type(num1))
# Output class 'float'
```

Casting Boolean to float

```python
flag_true = True
flag_false = False
print(type(flag_true))
# Output class 'bool'

# converting boolean to float
num1 = float(flag_true)
num2 = float(flag_false)

print("Float number 1:", num1)
# Output 1.0
print(type(num1))
# class 'float'

print("Float number 2:", num2)
# Output 0.0
print(type(num2))
# class 'float'
```

Casting string to float

```python
string_num = "725.535"
print(type(string_num))
# Output class 'str'

# converting str to float
num1 = float(string_num)

print("Float number:", num1)
# Output 725.535
print(type(num1))
# class 'float'
```

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

1. We can convert any type to `float` type, but we cannot cast **complex** to float type.
2. 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

```
r_num = 135
print(type(r_num)) # class 'int'

# converting int to complex(x)
c_num = complex(r_num)

print("Complex number:", c_num)
# Output (135+0j)
print(type(c_num))
# class 'complex'

# converting int to complex(x, y)
r_num, i_num2 = 135, 235
c_num = complex(r_num, i_num2)

print("Complex number:", c_num)
# Output (135+235j)
print(type(c_num))  # class 'complex'
```

Casting float type to complex type

```
r_num = 53.250
print(type(r_num))  # class 'float'

# converting float to complex(x)
```

```
c_num = complex(r_num)

print("Complex number:", c_num)
# Output (53.25+0j)
print(type(c_num))
# class 'complex'

# converting float to complex(x, y)
r_num, i_num2 = 53.250, 350.750
c_num = complex(r_num, i_num2)

print("Complex number:", c_num)
# Output (53.25+350.75j)
print(type(c_num))
# class 'complex'
```

Casting Boolean type to complex type

```
boolean_true = True
print(type(boolean_true))  # class 'bool'

# converting boolean to complex(x)
c_num = complex(boolean_true)

print("Complex number:", c_num)
# Output (1+0j)
print(type(c_num))
# class 'complex'

# converting boolean to complex(x, y)
r_bool, i_bool = False, True
c_num = complex(r_bool, i_bool)

print("Complex number:", c_num)
# Output 1j
print(type(c_num))
# 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

```
num1 = 10
num2 = 0
print(type(num1))  # class 'int'

# Convert into to bool
b1 = bool(num1)
b2 = bool(num2)

print(b1)
# Output True
print(b2)
# Output False

print(type(b1))
# class 'bool'
```

Casting float to Boolean type

```
f_num1 = 25.35
f_num2 = 0.0
print(type(f_num1))  # class 'float'

# Convert float into to bool
b1 = bool(f_num1)
b2 = bool(f_num2)

print(b1)
# Output True

print(b2)
# Output False

print(type(b1))
# Output class 'bool'
```

Casting string to Boolean type

```
s1 = "False"
s2 = "True"
s3 = "812"
s4 = ""
print(type(s1))  # class 'str'

# Convert string into to bool
b1 = bool(s1)
b2 = bool(s2)
b3 = bool(s3)
b4 = bool(s4)

print(b1)  # True
print(b2)  # True
print(b3)  # True
print(b4)  # False
print(type(b1))  # class 'bool'
```

Casting complex type to Boolean type

```
c1 = 33 + 9j
c2 = 0 + 0j
print(type(c1))  # class 'complex'

# Convert complex value into to bool
b1 = bool(c1)
b2 = bool(c2)

print(b1)  # True
print(b2)  # False
print(type(b1))  # 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

```python
num = 15
print(type(num))  # class 'int'

# converting int to str type
s1 = str(num)
print(s1)
# Output '15'
print(type(s1))
# Output class 'str'
```

Casting float type to str type

```python
num = 75.35
print(type(num))  # class 'float'

# converting float to str type
s1 = str(num)
print(s1)
# Output '75.35'
print(type(s1))
# Output class 'str'
```

Casting complex type to str type

```python
complex_num = 15 + 5j
print(type(complex_num))  # class 'complex'

# converting complex to str type
s1 = str(complex_num)
print(s1)
# Output '(15+5j)'

print(type(s1))
# class 'str'
```

Casting bool type to str type

```python
b1 = True
b2 = False
print(type(b1))  # class 'bool'

# converting bool to str type
s1 = str(b1)
s2 = str(b2)
print(s1)
# Output 'True'
print(s2)
# Output 'False'
print(type(s1))  # class 'str'
```