

**Python Type Conversion and Type Casting** In this, you will learn about the Type conversion and uses

of type conversion.

Before learning Type Conversion in Python, you should have knowledge about [Python Data Types](#).

## Type Conversion

The process of converting the value of one data type (integer, string, float, etc.) to another data type is called type conversion. Python has two types of type conversion.

1. Implicit Type Conversion
2. Explicit Type Conversion

## Implicit Type Conversion

In Implicit type conversion, Python automatically converts one data type to another data type. This process doesn't need any user involvement.

Let's see an example where Python promotes the conversion of the lower data type (integer) to the higher data type (float) to avoid data loss.

### Example 1: Converting integer to float

```
num_int = 123
```

```
num_flo = 1.23
```

```
num_new = num_int + num_flo
```

```
print("datatype of num_int:",type(num_int))
```

```
print("datatype of num_flo:",type(num_flo))
```

```
print("Value of num_new:",num_new)
```

```
print("datatype of num_new:",type(num_new))
```

When we run the above program, the output will be:

```
datatype of num_int: <class 'int'>
```

```
datatype of num_flo: <class 'float'>
```

```
Value of num_new: 124.23
```

```
datatype of num_new: <class 'float'>
```

In the above program,

- We add two variables num\_int and num\_flo, storing the value in num\_new.
- We will look at the data type of all three objects respectively.
- In the output, we can see the data type of num\_int is an integer while the data type

of num\_flo is a float.

- Also, we can see the num\_new has a float data type because Python always converts smaller data types to larger data types to avoid the loss of data.

Now, let's try adding a string and an integer, and see how Python deals with it.

### **Example 2: Addition of string(higher) data type and integer(lower) datatype**

```
num_int = 123
num_str = "456"

print("Data type of num_int:",type(num_int))
print("Data type of num_str:",type(num_str))

print(num_int+num_str)
```

When we run the above program, the output will be:

Data type of num\_int: <class 'int'>

Data type of num\_str: <class 'str'>

Traceback (most recent call last):

File "python", line 7, in <module>

TypeError: unsupported operand type(s) for +: 'int' and 'str'

In the above program,

- We add two variables num\_int and num\_str.
- As we can see from the output, we got TypeError. Python is not able to use Implicit

Conversion in such conditions.

- However, Python has a solution for these types of situations which is known as Explicit

Conversion.

## **Explicit Type Conversion**

In Explicit Type Conversion, users convert the data type of an object to required data type. We use the predefined functions like int(), float(), str(), etc to perform explicit type conversion.

This type of conversion is also called typecasting because the user casts (changes) the data type of the

objects.

Syntax :

<required\_datatype>(expression)

Typecasting can be done by assigning the required data type function to the expression.

### **Example 3: Addition of string and integer using explicit conversion**

```
num_int = 123
```

```
num_str = "456"
```

```
print("Data type of num_int:",type(num_int))
```

```
print("Data type of num_str before Type Casting:",type(num_str))
```

```
num_str = int(num_str)
```

```
print("Data type of num_str after Type Casting:",type(num_str))
```

```
num_sum = num_int + num_str
```

```
print("Sum of num_int and num_str:",num_sum)
```

```
print("Data type of the sum:",type(num_sum))
```

When we run the above program, the output will be:

Data type of num\_int: <class 'int'>

Data type of num\_str before Type Casting: <class 'str'>

Data type of num\_str after Type Casting: <class 'int'>

Sum of num\_int and num\_str: 579

Data type of the sum: <class 'int'>

In the above program,

- We add num\_str and num\_int variable.
- We converted num\_str from string(higher) to integer(lower) type using int() function to perform the addition.

- After converting num\_str to an integer value, Python is able to add these two variables.
- We got the num\_sum value and data type to be an integer.

### **Key Points to Remember**

1. Type Conversion is the conversion of object from one data type to another data type.
2. Implicit Type Conversion is automatically performed by the Python interpreter.
3. Python avoids the loss of data in Implicit Type Conversion.
4. Explicit Type Conversion is also called Type Casting, the data types of objects are converted using predefined functions by the user.
5. In Type Casting, loss of data may occur as we enforce the object to a specific data type.