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))

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)

## 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.

### **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.

*# Type Conversion mean that converting from one data type to another data type  
# example convert from int to float , float to int , int to string , string to int  
# Implicit type conversion  
# Explicit type conversion*num1 = 10  
num2 = 23.40  
print(type(num1))  
print(type(num2))  
*# Implicit type conversion*num3 = num1 + num2  
print(num3)  
print(type(num3))  
  
*# Explicit type conversion*num4 = num1 + int(num2)  
print(num4)  
print(type(num4))  
  
*# If inside string if it contains only numbers it can be converted to int  
# if it contains alphabets it will throw error it can not convert*myname= **"123"**print(type(myname))  
num5 = int(myname)  
print(num5)  
print(type(num5))  
  
*# The below code will give error because it contains alphabets  
# myname= "ABC"  
# print(type(myname))  
# num5 = int(myname)  
  
# Convert from numbers to string*num6 =456  
num7 = str(num6)  
print(num7)