

Python Type Conversion and Type Casting

Anoop S Babu

Faculty Associate

Dept. of Computer Science & Engineering

bsanoop@am.amrita.edu

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.
 - Implicit Type Conversion
 - Explicit Type Conversion

Implicit Type Conversion

- In Implicit type conversion, Python **automatically** converts one data type to another data type.

Example 1: Converting integer to float

```
integerNumber = 123
floatNumber = 1.23
resultNumber = integerNumber + floatNumber
print("Datatype of integerNumber:", type(integerNumber))
print("Datatype of floatNumber:", type(floatNumber))
print("Value of resultNumber:", resultNumber)
print("Datatype of resultNumber:", type(resultNumber))
```

Output

```
Datatype of integerNumber: <class 'int'>
Datatype of floatNumber: <class 'float'>
Value of resultNumber: 124.23
Datatype of resultNumber: <class 'float'>
```

- Python always converts **smaller data types to larger data types** to avoid the loss of data.

Implicit Type Conversion

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

```
integerNumber = 123
stringNumber = "456"
print("Data type of integerNumber:", type(integerNumber))
print("Data type of stringNumber:", type(stringNumber))
print(integerNumber+stringNumber)
```

Output

```
Data type of integerNumber: <class 'int'>
Data type of stringNumber: <class 'str'>
Traceback (most recent call last):
  File "<string>", line 5, in <module>
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

- Python is not able to use Implicit Conversion
- 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**.
- Uses predefined functions like **int()**, **float()**, **str()**, **complex()** etc.
- This type of conversion is also called **typecasting** because the user casts (changes) the data type of the objects.

Syntax

```
<required_datatype>(expression)
```

Example

```
x = 1      # int
y = 2.8    # float
z = 1j     # complex
a = float(x)    #convert from int to float:
b = int(y)      #convert from float to int:
c = complex(x)  #convert from int to complex:
```

Explicit Type Conversion

Example 3: Addition of string and integer using explicit conversion

```
integerNumber = 123
stringNumber = "456"
print("Data type of integerNumber:",type(integerNumber))
print("Data type of stringNumber before Type Casting:",type(stringNumber))
stringNumber = int(stringNumber)
print("Data type of stringNumber after Type Casting:",type(stringNumber))
result = integerNumber + stringNumber
print("Sum of integerNumber and stringNumber:",result)
print("Data type of the sum:",type(result))
```

Output

```
Data type of integerNumber: <class 'int'>
Data type of stringNumber before Type Casting: <class 'str'>
Data type of stringNumber after Type Casting: <class 'int'>
Sum of integerNumber and stringNumber: 579
Data type of the sum: <class 'int'>
```

Key Points to Remember

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