Type Conversion in Python

It converts the one data type data into another form of data. It is a conversion technique. Implicit type translation and explicit type converter are Python's two basic categories of type conversion procedures. In Python, there are two kinds of type conversion, these are –

1. Explicit Type Conversion-The programmer must perform this task manually.
2. Implicit Type Conversion-By the Python program automatically.

Implicit Type Conversion

Implicit character data conversion is used in Python when a data type conversion occurs, whether during compilation or runtime.

Ex.

```
a = 15
print("Data type of a:",type(a))
b = 7.6
print("Data type of b:",type(b))
c = a + b
print("The value of c:", c)
print("Data type of c:",type(c))
```

Explicit Type Conversion:

Explicit type conversion is performed by the user by explicitly using type conversion functions in the program code.

```
a = '4'
# printing and converting a character to an integer
b = ord(a)
print ("After converting character into integer : ",end="")
print (b)
# printing integer converting to hexadecimal string
b = hex(56)
print ("After converting 56 to hexadecimal string : ",end="")
print (b)
# printing the integer converting into octal string
b = oct(56)
print ("After converting 56 into octal string : ",end="")
print (b)
```

Python Operators

Operators are used to perform operations on variables and values.

In the example below, we use the + operator to add together two values:

```
print(10 + 5)
```

Python divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators
- Identity operators
- Membership operators
- Bitwise operators

- **Arithmetic operators**

Operator	Description	Syntax	
Addition Operator	+	Addition: adds two operands	x + y
Subtraction Operator	–	Subtraction: subtracts two operands	x – y
Multiplication Operator	*	Multiplication: multiplies two operands	x * y

Division Operator	/	Division (float): divides the first operand by the second	x / y
Floor Division Operator	//	Division (floor): divides the first operand by the second	x // y
Modulus Operator	%	Modulus: returns the remainder when the first operand is divided by the second	x % y
Exponentiation Operator	**	Power (Exponent): Returns first raised to power second	x ** y

Ex.

```
val1 = 3

val2 = 2


# using the floor division

res = val1 // val2

print(res)
```

Ex. val1 = 2

```
val2 = 3
```

```
# using the exponentiation operator

res = val1 ** val2

print(res)
```

Assignment Operators in Python

Operators	Sign	Description	Syntax
Assignment Operator	=	Assign the value of the right side of the expression to the left side operand	c = a + b
Addition Assignment Operator	+=	Add right side operand with left side operand and then assign the result to left operand	a += b
Subtraction Assignment Operator	-=	Subtract right side operand from left side operand and then assign the result to left operand	a -= b
Multiplication Assignment Operator	*=	Multiply right operand with left operand and then assign the result to the left operand	a *= b
Division Assignment Operator	/=	Divide left operand with right operand and then assign the result to the left operand	a /= b
Modulus Assignment Operator	%=	Divides the left operand with the right operand and then assign the remainder to the left operand	a %= b