C# TYPE CONVERSION

Type conversion is converting one type of data to another type. It is also known as Type Casting. In C#, type casting has two forms:

Implicit conversions: No special syntax is required because the conversion is type safe and no data will be lost. Examples include conversions from smaller to larger integral types.

Explicit conversions (casts): Explicit conversions require a cast operator. Casting is required when information might be lost in the conversion, or when the conversion might not succeed for other reasons. Typical examples include numeric conversion to a type that has a smaller range

In the following example, the compiler **implicitly converts** the value type int to a type long:

```
// Implicit conversion. num long can
// hold any value an int can hold, and more!
int num = 2147483647;
long bigNum = num;
```

In above conversion no data will be lost.

The following example shows an explicit type conversion:

```
using System;
namespace TypeConversionApplication
{
   class ExplicitConversion
   {
      static void Main(string[] args)
      {
         double d = 5673.74;
         int i;

         // cast double to int.
         i = (int)d;
         Console.WriteLine(i);
         Console.ReadKey();
      }
   }
}
```

When the above code is compiled and executed, it produces the following result:

```
5673
```

C# Type Conversion Methods

C# provides the following built-in type conversion methods:

Sr.No	Methods & Description
1	**ToBoolean:** Converts a type to a Boolean value, where possible.
2	**ToByte:** Converts a type to a byte.
3	**ToChar:** Converts a type to a single Unicode character, where possible.
4	**ToDateTime:** Converts a type *integerorstringtype* to date-time structures.
5	**ToDecimal:** Converts a floating point or integer type to a decimal type.
6	**ToDouble:** Converts a type to a double type.
7	**ToInt16:** Converts a type to a 16-bit integer.
8	**ToInt32:** Converts a type to a 32-bit integer.
9	**ToInt64:** Converts a type to a 64-bit integer.
10	**ToSbyte:** Converts a type to a signed byte type.
11	**ToSingle:** Converts a type to a small floating point number.
12	**ToString:** Converts a type to a string.
13	**ToType:** Converts a type to a specified type.
14	**ToUInt16:** Converts a type to an unsigned int type.
15	**ToUInt32:** Converts a type to an unsigned long type.
16	**ToUInt64:** Converts a type to an unsigned big integer.

The following example converts various value types to string type:

```
using System;
namespace TypeConversionApplication
{
   class StringConversion
   {
      static void Main(string[] args)
      {
         int i = 75;
         float f = 53.005f;
         double d = 2345.7652;
         bool b = true;

         Console.WriteLine(i.ToString());
         Console.WriteLine(f.ToString());
         Console.WriteLine(d.ToString());
         Console.WriteLine(b.ToString());
         Console.ReadKey();

      }
   }
}
```

When the above code is compiled and executed, it produces the following result:

```
75
53.005
2345.7652
True
```