**Part 7 - C# Tutorial - Datatype conversions**

**In this video, we will discuss**  
**1.** Implicit conversions  
**2.** Explicit Conversions  
**3.** Difference between Parse() and TryParse()

**Implicit conversion is done by the compiler:**  
**1.** When there is no loss of information if the conversion is done  
**2.** If there is no possibility of throwing exceptions during the conversion

**Example:**Converting an **int** to a **float** will not loose any data and no exception will be thrown, hence an implicit conversion can be done.   
  
Where as when converting a **float** to an **int**, we loose the fractional part and also a possibility of overflow exception. Hence, in this case an explicit conversion is required. For explicit conversion we can use cast operator or the convert class in c#.  
  
**Implicit Conversion Example**  
using System;  
class Program  
{  
    public static void Main()  
    {  
        int i = 100;  
  
        // float is bigger datatype than int. So, no loss of  
        // data and exceptions. Hence implicit conversion  
        float f = i;  
  
        Console.WriteLine(f);  
    }  
}  
  
**Explicit Conversion Example**  
using System;  
class Program  
{  
    public static void Main()  
    {  
        float f = 100.25F;  
  
        // Cannot implicitly convert float to int.  
        // Fractional part will be lost. Float is a  
        // bigger datatype than int, so there is  
        // also a possiblity of overflow exception  
        // int i = f;  
  
        // Use explicit conversion using cast () operator  
        int i = (int)f;  
  
        // OR use Convert class  
        // int i = Convert.ToInt32(f);  
  
        Console.WriteLine(i);  
    }  
}  
  
**Difference between Parse and TryParse**  
**1.** If the number is in a string format you have 2 options - Parse() and TryParse()   
**2.** Parse() method throws an exception if it cannot parse the value, whereas TryParse() returns a bool indicating whether it succeeded or failed.

**3.** Use Parse() if you are sure the value will be valid, otherwise use TryParse()