ToString() METHOD

- This method is used to translate a numeric value into a **string**.

```
int age = 16;       //start with double
MessageBox.Show("My age is" + age.ToString());
```

- You can convert a numeric value to a string and format it in the same step by using the ToString() method with some special parameters.

- Here is the list of possible ToString() parameters that we can use on this value.
  - **Fx** – Format the number to "x" number of decimal places
  - **C** – Format the number to a currency value
  - **Dx** – Format the number to "x" number of digits. If the number has more digits than the "x" number, the digits will be truncated. If the number has less digits, the number will be padded with zeros at the beginning
  - **P** – Format the number to a percentage value

- Let's assume that we are working with the following variable:

```
double myNumber = 12.5648;
```

- Here is an example of these method parameters:

```
newString = myNumber.ToString("F2");   //newString = "12.56"
newString = myNumber.ToString("C");    //newString = "$12.56"
newString = myNumber.ToString("D3");   //newString = "012"
newString = myNumber.ToString("P");    //newString = "12%"
```

Parse() Method

- If we want to convert a string value to a double or an int value, things can get a bit complicated. This is not an easy operation to perform by the compiler, and not all strings can be converted to numerical values. For this, we have a special method called Parse.
- Here is an example of how to use this special method:

```
int result = int.Parse("10");
double otherResult = double.Parse(txtAnswer.text);
```

TryParse() Method

- If, for any reason, the string cannot be converted to the data type that you have chosen, an error will occur. If you are not sure the input string can be converted correctly you should consider using the **TryParse()** method instead.
- **TryParse()** syntax looks like this:

```
bool <data type>.TryParse(<input string>, out <variable to update>);
```

EXAMPLE

```
int result;
bool success1 = int.TryParse("8675304", out result);

//success1 will be set to true and result will contain 5675304

int result2;
bool success2 = int.TryParse("true", out result2);

//success2 will be set to false and result2 will not be changed
```