Introduction

CsTools is a C# utility library written in .NET 8. It provides common utilites of all types - strings, math tools, and others. For example, you can change and retrieve a casing mode for a string, or map numbers of any numeric type that exists (because of the generic INumber interface) from an input range to an output range.

Example

```
myStr.IsNullOrEmpty();
myStr2.IsNullOrWhiteSpace();
myIntString.TryParseNumber<int>(out int val);

char[] chars = ['t', 'e', 's', 't'];
Console.WriteLine(chars.GetString()); // --> test

if ("Some Text".GetCasing() == StringCasing.Title)
{
  Console.WriteLine("Is title case!");
}

Console.WriteLine("RandomTest".SetCasing(StringCasing.Camel)); // --> randomTest

"Bob".IsPalindrome();

Console.WriteLine(MathTools.Map(5, new Range<int>(1, 10), new Range<int>(11, 20))); // --
> 15

MathTools.Mod(10, 2);

MathTools.IsEven(6);
MathTools.IsOdd(3);
```

Getting Started

To start using CsTools, you need to import the DLL into your Visual Studio project. To import it, follow these steps:

Import the DLL

1. Download the latest DLL from the Releases page.
2. In your Visual Studio project, right-click **References** and select **Add Project Reference**.
3. In the dialog, select the **Browse** tab from the sidebar.
4. Click the **Browse...** button in the bottom-right corner.
5. In the file dialog, browse to the DLL you downloaded and select it.
6. Back in the **Reference Manager** dialog, make sure the DLL name is checked.
7. Finally, press **OK**.

Usage

Now that the DLL is imported into our project, we can start using the CsTools library. To start, add `using FireBlade.CsUtils` at the top of your script. Optionally, you can also add `using FireBlade.CsUtils.Numbers` if you want to get the math components as well. Now that we're set, we can start using the library. For example, you can use some string extension methods:

```
string s = textBox1.Text;

// shortcut
if (s.IsNotNullOrEmpty())
{
        // do something...
}
```

Or change the casing of a string:

```
string str = "hello, world!";
string newStr = str.SetCasing(StringCasing.Title);

Console.WriteLine(newStr.GetCasing().ToString()); // --> Title
Console.WriteLine(newStr);
```

If you have the math components imported, you can use some of the math utilites as well:

```
var range1 = new Range<int>(1, 10);
var range2 = new Range<int>(11, 20);
```

```
var mapped = MathTools.Map<int>(5, range1, range2);

Console.WriteLine(mapped); // --> 15
```

Note: CsUtils can handle any number type, because it uses the generic INumber interface, so you can use int, double, float or any other type you want.