Working with Strings

Gill Cleeren
CTO Xpirit Belgium

@gillcleeren | xpirit.com/gill

Agenda

Working with strings

Comparing strings

Parsing from strings to other types

Demo

A small recap on strings

Working with Strings

`int l = myString.Length;`	◀ Get the length of the string
`string upper = myString.ToUpper();`	◀ Set the string to uppercase
`string lower = myString.ToLower();`	◀ Set the string to lowercase
`bool b = myString.Contains("Hello");`	◀ Check if a string contains “Hello”, return bool
`string s = myString.Replace("a","b");`	◀ Replace “a” with “b” in the string
`string sub = myString.Substring(1, 3);`	◀ Get a part of the string (zero-based)

```
string s1 = "Learning C# ";//notice the extra space at the end
string s2 = "is awesome";
string s3 = s1 + s2;
//Output: "Learning C# is awesome"
```

Concatenating Multiple Strings

```csharp
string s1 = "Learning C# ";//notice the extra space at the end
string s2 = "is awesome";
string s3 = String.Concat(s1, s2);
//Output: "Learning C# is awesome"
```

Using String.Concat

```
string employeeName = "Bethany";
int age = 34;
string greetingText = "Hello " + employeeName + ", you are " + age + " years";
//Output: Hello Bethany, you are 34 years
```

Less-readable String Concatenation

```
string employeeName = "Bethany";
int age = 34;
string greetingText =
    string.Format("Hello {0}, you are {1} years", employeeName, age);
//Output: Hello Bethany, you are 34 years
```

Using string.Format to Concatenate Strings

```
string employeeName = "Bethany";
int age = 34;
string greetingText = $"Hello {employeeName}, you are {age} years";
//Output: Hello Bethany, you are 34 years
```

String Interpolation

Often better and easier to read

Demo

Manipulating strings

Concatenating strings

Using string interpolation

```
Console.WriteLine("Here are the employee details:\nBethany\tSmith");
```

Adding Escape Characters

Always start with a \

```
string escapedFilePath = "C:\\Documents\\readme.txt";
```

Representing a File Path

```
string escapedFilePath = "C:\\Documents\\readme.txt";
string verbatimFilePath = @"C:\Documents\readme.txt";
```

Using Verbatim Strings

Used when text contains \ as part of the content
Improves readability

Demo

Escaping text

Using verbatim strings

Testing Strings for Equality

```
string firstName  = "Bethany";
bool b1 = firstName == "Bethany";//true
bool b2 = firstName == "bethany";//false
bool b3 = firstName.Equals("Bethany");//true
```

Comparing Two Strings

```
bool b = firstName.ToUpper() == anotherString.ToUpper();
```

Comparing Strings Case-insensitive

Demo

Comparing strings

Parsing from Strings to Other Types

```
string w = Console.ReadLine();
double wage = double.Parse(w);

bool active = bool.Parse("true");
```

Use Parsing to Generate a Value from a String

Can cause issues though

```
string enteredText = "true";
if (bool.TryParse(enteredText, out bool b))
{
    Console.WriteLine($"The value is {b}");
}
```

Using TryParse

The out keyword will be covered in the next module

Demo

Parsing strings into other types

Using TryParse

Summary

Strings are a very important concept

Many methods available

- Concatenation
- Parsing

Up next:

Working with classes and objects