## 0. Introduction

### 1. Working with text in .NET

### 2. How to access and clone the GitHub code

### 3. Open the cloned solution in Visual Studio

### 4. Work with Git branches

## 1. Overview

### 1. .NET, text, and strings

### 2. What is a string

### 3. Declaring a string variable

### 4. Initialize a string variable

### 5. Immutable - What does that mean in .NET strings

## 2. The Char Type

### 1. The char, the building block of a strong

### 2. Initialize a string from a char array

### 3. Get a char from a string

### 4. Use char methods to analyze Unicode data

## 3. String Basics

### 1. Three ways to work with string literals

### 2. Use escape sequences to add control chars to a string

### 3. Use verbatim string literals

## 4. Composing Strings

### 1. Combine strings with String.Concat

### 2. Combine strings with the concatenation operator

### 3. Composite formatting with placeholders

### 4. Use C# interpolation for simpler composite formatting

### 5. Use StringBuilder to improve performance

### 6. Use Perf Monitor to examine performance improvements

## 5. String Methods

### 1. Splitting strings into an array

### 2. Use LINQ to query string array

### 3. Joining an array into a string

### 4. Trim unwanted content from a string

### 5. Work with a subset of a string

### 6. Verify that string contains search characters

### 7. Replace and remove content

### 8. Challenge - Better substring

### 9. Solution - Better substring

## 6. How Custom Types Format and Parse Strings

### 1. Custom types and string conversions

### 2. How custom types are converted to string with toString()

### 3. How custom types override toString()

### 4. IFormattable and the toString() call order

### 5. Custom formats with IFormattable

### 6. Working with culture-specific formats

### 7. Challenge - Split a string into two numbers

### 8. Solution - Convert a string to the type with parse()

### 9. Add custom filters to parse()

### 10. Prevent exceptions with a tryParse() implementation

## 7. Formatting

### 1. Formatting data with the built-in numeric formats

### 2. Formatting data with custom formatters

### 3. Apply conditional string formats

### 4. Formatting dates

### 5. Challenge - Ordinal formatting

### 6. Solution - Ordinal formatting

## 8. Convert and Parse

### 1. The parse() method and the Convert class

### 2. Use tryParse() for better error handing

## 9. Culture Information

### 1. Why is culture important for strings

### 2. The CultureInfo class

### 3. Use culture when formatting strings

### 4. Invariant culture and strings

## 10. Compare and Sort

### 1. How string comparison works in .NET

### 2. Test for equality with String.Equals

### 3. Compare strings with String.Compare

### 4. Sort strings with Array.Sort and LINQ

## 11. Conclusion

### 1. Next steps