String handling in C#.NET

1. **Creating Strings**

```
string name = "Alice";
string greeting = "Hello, " + name + "!";
string multiline = @"This is
a multiline
string.";
```

> The `@` symbol before a string literal means **verbatim string**, which preserves line breaks and backslashes.

2. **String Interpolation**

Introduced in C# 6.0 — a cleaner way to build strings.

```
string name = "Alice";
int age = 30;
string message = $"My name is {name}, and I am {age} years old.";
```

3. **Common String Methods**

```
string text = "  Hello World!  ";

text.Length;              // 15
text.ToUpper();           // "  HELLO WORLD!  "
text.ToLower();           // "  hello world!  "
text.Trim();              // "Hello World!"
text.StartsWith("  He"); // true
text.EndsWith("!  ");     // true
text.Contains("World");   // true
text.Replace("World", "C#"); // "  Hello C#!  "
```

4. **Substring and Indexing**

```
string phrase = "Programming";
string sub = phrase.Substring(3, 4);  // "gram"
```

```
char letter = phrase[0];           // 'P'
int index = phrase.IndexOf("gram");   // 3
```

---

5. **Splitting and Joining**

```
string csv = "apple,banana,cherry";
string[] fruits = csv.Split(',');

foreach (var fruit in fruits)
    Console.WriteLine(fruit); // apple, banana, cherry

string joined = string.Join(" | ", fruits);
// "apple | banana | cherry"
```

---

6. **StringBuilder for Performance**

When doing **many concatenations**, use `StringBuilder` to avoid creating multiple string objects.

```
using System.Text;

StringBuilder sb = new StringBuilder();
sb.Append("Hello");
sb.Append(" ");
sb.Append("World!");
string result = sb.ToString();  // "Hello World!"
```

---

7. **String Comparison**

```
string a = "hello";
string b = "HELLO";

bool equal = string.Equals(a, b, StringComparison.OrdinalIgnoreCase); // true
```

---

8. **Checking for Null or Empty**

```
string empty = "";
string nullString = null;

string.IsNullOrEmpty(empty);        // true
string.IsNullOrWhiteSpace("   ");   // true
```

---

9. **Formatting Strings**

```
string result = string.Format("Name: {0}, Age: {1}", "Alice", 30);
// "Name: Alice, Age: 30"
```

Or, prefer modern interpolation:

```
string result = $"Name: {"Alice"}, Age: {30}";
```

---

10. **Escape Sequences**

```
string path = "C:\\Users\\Admin\\Documents";
string quoted = "He said, \"Hello!\"";
```

Or use verbatim strings:

```
string path = @"C:\Users\Admin\Documents";
```
