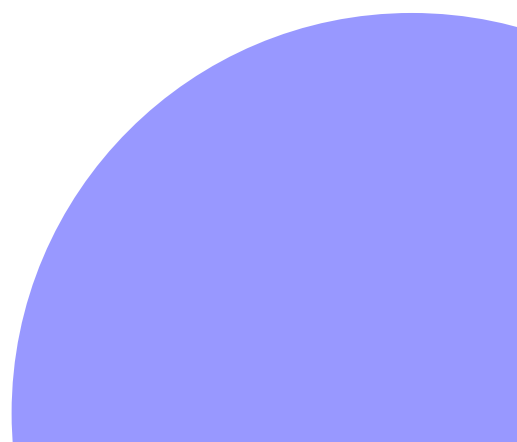


# Strings

2025-01-10



## I. Introduction

Strings in C# are objects that represent sequences of characters. They are immutable, meaning once created, their contents cannot be changed. C# provides a rich set of methods and properties for working with strings efficiently.

### I.1. Declaration and Initialization

### I.2. Creating Strings

You can create strings in several ways:

```
// Using string literals
string greeting = "Hello, World!";

// Using string concatenation
string fullName = "John" + " " + "Doe";

// Using string interpolation (C# 6.0+)
string message = $"Welcome, {fullName}!";
```

== Accessing Elements  
Access array elements using their index:

### I.3. String Properties and Methods

C# strings come with useful properties and methods:

```
string text = "Example Text";

int length = text.Length; // Returns 12

string upper = text.ToUpper(); // Returns "EXAMPLE TEXT"
string lower = text.ToLower(); // Returns "example text"

bool contains = text.Contains("amp"); // Returns true
string trimmed = " Hello ".Trim(); // Returns "Hello"

int index = trimmed.IndexOf('e'); // Returns 1
```

Strings are kind of special. They hold an array of chars internally. You can access characters by its 0-based index:

```
var str = "hello";
int index = str[0]; // Returns 'h'
```

### I.4. String Manipulation

You can manipulate strings using various methods:

```
string original = "Hello, World!";

string substring = original.Substring(0, 5); // Returns "Hello"
string replaced = original.Replace("World", "C#"); // Returns "Hello, C#!"
string[] split = original.Split(','); // Returns ["Hello", " World!"]
```

## Strings

Remember that string operations that appear to modify a string actually create a new string object.

## II. Exercises

The amount of 🌶️ determines how hard the exercises are.

1. Create a program that takes a sentence as input and counts the number of words in it. Assume words are separated by spaces. Print the total word count. 🌶️
2. Create a program that takes an input and prints the string character by character in reversed order in new lines. 🌶️ 🌶️
3. Caesar chiffré: Write a program that can print a string as caesar encrypted value. The user enters a number as encryption key and the program displays every character shifted by that key in the alphabet. You can ignore the casing. 🌶️ 🌶️ 🌶️

### Samples:

Input: Hello  
Key: 7  
Output: olssv

Input: The zoo is big  
Key: 3  
Output: Wkh crr lv elj

4. Do the same as in 3. but decode the chiffré. 🌶️ 🌶️ 🌶️