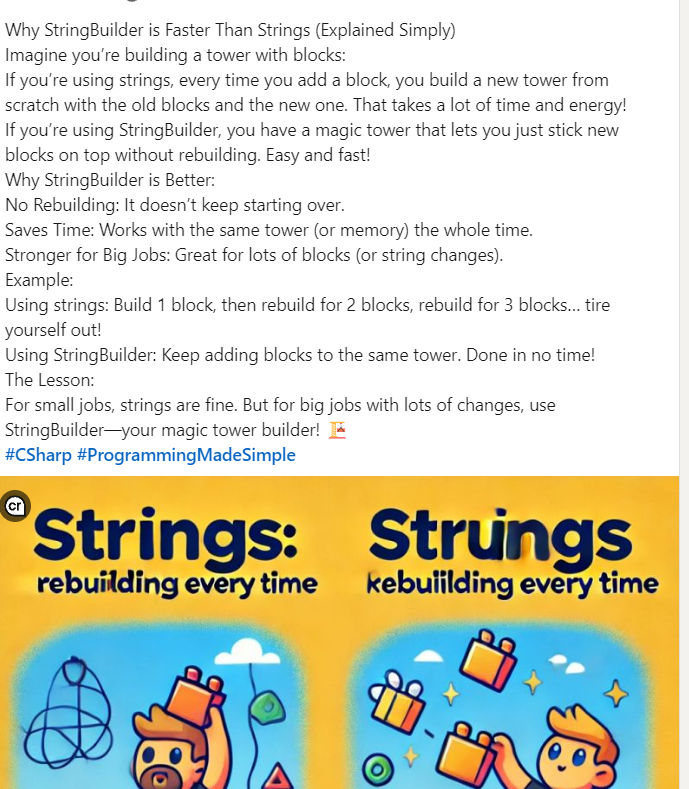
LinkedIn article about string immutability

The link : <https://www.linkedin.com/posts/ibrahim-elmahy-12b968265_csharp-programmingmadesimple-activity-7267968154574745601-Roj8?utm_source=share&utm_medium=member_desktop>



**1. What is an Enum, and When is it Used?**

An **enum** (short for enumeration) is a value type in C# that defines a set of named constants. It provides a way to represent a collection of related values with meaningful names, making code more readable and maintainable.

**When to Use an Enum:**

* To represent a fixed set of related constants (e.g., days of the week, directions, or user roles).
* When you want to make code more expressive and less error-prone by using meaningful names instead of arbitrary values.

**Syntax Example:**

enum DaysOfWeek

{

Sunday,

Monday,

Tuesday,

Wednesday,

Thursday,

Friday,

Saturday

}

Usage:

DaysOfWeek today = DaysOfWeek.Monday;

Console.WriteLine($"Today is {today}.");

**Three Common Built-in Enums:**

1. **DayOfWeek**: Represents days of the week.
2. **ConsoleColor**: Defines colors for console text/background.
3. **StringComparison**: Specifies how strings are compared (e.g., case-insensitive).

**2. Scenarios to Use String vs. StringBuilder**

**When to Use Strings:**

* **Simple and infrequent modifications**: Strings are easy to use and efficient for a small number of operations.
  + Example: Assigning values, comparing, or performing one or two concatenations.

string greeting = "Hello";

greeting += " World!"; // Fine for small tasks

* **When immutability is important**: Strings are thread-safe and prevent accidental changes.

**When to Use StringBuilder:**

* **Frequent or large-scale string modifications**: Ideal for operations inside loops or where performance is critical.
  + Example: Appending, inserting, or modifying strings repeatedly.

StringBuilder sb = new StringBuilder();

for (int i = 0; i < 1000; i++)

{

sb.Append($"Line {i}\n");

}

* **Performance-critical applications**: StringBuilder avoids creating multiple intermediate string objects, reducing memory usage and improving speed.

**Summary Table for Strings vs. StringBuilder:**

| **Feature** | **String** | **StringBuilder** |
| --- | --- | --- |
| **Mutability** | Immutable (creates new objects) | Mutable (modifies buffer in place) |
| **Use Case** | Simple, infrequent changes | Frequent, large-scale modifications |
| **Performance** | Slower for repeated changes | Faster for heavy modifications |
| **Memory Efficiency** | More memory-intensive | More efficient for repeated edits |