

بـص يا سيدي، في C# أي **string** لما تكتبه أو تعمله تعديل، هو في الحقيقة **ما بيتعدلش!** هو بيعمل **نُسخة جديدة** بالنص الجديد ويسيب القديمة زي ما هي.

مثال بسيط:

```

string x = "Hello";
x += " World"; // هنا ما عدلش

```

جديدة خالص String ده عمل ،x هنا ما عدلش

يعني كل مرة "تزود" على string أو "تعديلها"، **بيخلق واحدة جديدة** وده ممكن ببقى ثقيل على الميموري لو عملت كده كتير.

طيب ليه كده؟ إيه الفائدة؟

- ✔️ **أمان أكثر:** لو في string فيها كلمة سر أو عنوان، محدش يقدر يغيرها فجأة.
- ✔️ **آمن في البرامج اللي بتشتغل بكذا خيط (Thread):** عشان مفيش حد هيغيرها وانت مش واخد بالك.
- ✔️ **الهاش كود مبيتلعبش فيه:** مهم جدًا لما تستخدم string كمفتاح في Dictionary.

طيب أعمل إيه لو عندي تعديلات كتير على string؟

لو بتعدل في النص كتير (خصوصًا في لوب أو ملف كبير)، ساعتها **استخدم `StringBuilder`** ، لأنه بيعدل في نفس النسخة من غير ما يخلق واحدة جديدة كل شوية.

An **enum** (short for *enumeration*) is a **special value type** in C# that lets you define a set of **named constants** to represent **fixed choices**.

Use **enum** when:

- ✦ You want to improve **code readability** with meaningful names instead of numbers.
- ✦ You need to represent a **fixed set of related values**.
- ✦ You want to **avoid magic numbers** and **make your code self-documenting**.

Three Common Built-in Enums:

Enum	Namespace	Description
<code>DayOfWeek</code>	<code>System</code>	Represents days (Sunday to Saturday)
<code>ConsoleColor</code>	<code>System</code>	Used to set console text/background colors
<code>DateTimeKind</code>	<code>System</code>	Specifies whether a <code>DateTime</code> is local, UTC, or unspecified

Use Case	Use <code>String</code>	Use <code>StringBuilder</code>
Small, few modifications	✔️ When you're doing 1–2 concatenations	❌ Not necessary
Readability & simplicity	✔️ For clear, simple formatting (e.g., interpolation)	❌ <code>StringBuilder</code> adds complexity
Repeated concatenation in loops	❌ Causes many temporary string objects → slow	✔️ Efficient with frequent changes
Large-scale string manipulations	❌ Inefficient and memory-heavy	✔️ Ideal for building large text blocks
Performance-critical scenarios	❌ May impact performance	✔️ Better speed and memory handling

