1. Built-in Functions

Built-in functions are the **ready-made methods** that come with a programming language or framework. We don't need to write them from scratch because they are already provided. For example, in C# methods like ToString(), Length, or Substring() are built-in. They save time and make coding easier.

---

2. Dynamic

dynamic in C# is a special type that lets us **store any kind of data** without knowing the type at compile time. The compiler skips type checking until runtime. This gives flexibility, but also means errors may appear only when the program runs. It is useful when working with data from unknown or flexible sources (like JSON or external libraries).

---

3. Create, Delete, Type – When?

- **Create** → When we want to make a new object, array, or variable.
- **Delete** → When we want to remove something that we no longer need (in C#, the *garbage collector* usually handles memory cleanup).
- **Type** → Choosing the right data type when creating variables or objects (int, string, bool, or even custom classes) is important because it defines what kind of values can be stored.

---

4. Yielding

The yield keyword in C# is used when we want to **return items one by one** instead of all at once. It is useful in loops and iterators. For example, if we have a big list, yield return lets us send each element step by step without creating a temporary collection, saving memory.

---

5. Lazy Loading

Lazy Loading means **loading data only when it is actually needed**. Instead of bringing all data at once (which might be heavy and slow), the program waits until the specific data is requested. For example, in databases or Entity Framework, related objects are not loaded until we access them. This improves performance and saves resources.