

**Extension methods** در **C#** اجازه می‌دهند که به یک کلاس جدیدی عملیاتی اضافه کنید بدون اینکه کلاس اصلی را تغییر دهید. این امکان اجازه می‌دهد که کدها را سازماندهی و مدیریت بهتری داشته باشید. در زیر چند مثال از عملیات‌هایی که می‌توانید با استفاده از **extension methods** انجام دهید را مشاهده می‌کنید:

### 1. تبدیل رشته به **int**

ممکن است نیاز داشته باشید که یک رشته را به عدد **integer** تبدیل کنید. این کار با استفاده از یک **extension method** انجام می‌شود:

```
public static class StringExtensions
{
    public static int ToInt(this string str)
    {
        ;int result
        ;int.TryParse(str, out result)
        ;return result
    }
}
```

حالا می‌توانید از این **extension method** به صورت زیر استفاده کنید:

```
string numberString = "123";
int number = numberString.ToInt();
```

تبدیل رشته به DateTime:

مشابه مثال قبلی، می‌توانید یک رشته را به یک شیء DateTime تبدیل کنید:

```
public static class StringExtensions
{
    public static DateTime ToDateTime(this string str)
    {
        DateTime result;
        DateTime.TryParse(str, out result);
        return result;
    }
}
```

مثال استفاده:

```
string dateString = "2024-05-11";
DateTime date = dateString.ToDateTime();
```

### 3. تبدیل Enum به String :

برای تبدیل یک Enum به String، می‌توانید یک extension method به شکل زیر بنویسید:

```
public static class EnumExtensions
{
    public static string ToDescriptionString(this Enum value)
    {
        var field = value.GetType().GetField(value.ToString());
        var attribute =
            (DescriptionAttribute)Attribute.GetCustomAttribute(field,
            typeof(DescriptionAttribute));
        return attribute == null ? value.ToString() : attribute.Description;
    }
}
```

موارد استفاده مانند:

```
enum WeekDays
{
    [Description("شنبه")]
    Sunday,
    [Description("یکشنبه")]
    Monday,
}

WeekDays day = WeekDays.Sunday;
string dayString = day.ToDescriptionString();
```

#### 4. فیلتر کردن لیست:

می‌توانید یک **extension method** برای فیلتر کردن لیست ایجاد کنید:

```
public static class ListExtensions
{
    public static List<T> Filter<T>(this List<T> list, Func<T, bool>
predicate)
    {
        var result = new List<T>();
        foreach (var item in list)
        {
            if (predicate(item))
            {
                result.Add(item);
            }
        }
        return result;
    }
}
```

موارد استفاده مانند:

```
List<int> numbers = new List<int> { 1, 2, 3, 4, 5 };
```

```
List<int> evenNumbers = numbers.Filter(x => x % 2 == 0); // [2, 4]
```

این نمونه‌ها فقط چند مثال از استفاده از **extension methods** هستند. از آنها برای اضافه کردن عملیات به کلاس‌ها و سازماندهی کد خود استفاده کنید.