

"Override" در سی شارپ به معنای تغییر دادن رفتار متدی است که در کلاس پدر تعریف

شده است. به زبان ساده‌تر، این یعنی شما می‌توانید رفتار پیش فرض یک متد را تغییر دهید

برای پاسخ به نیازهای خاص خود.

چرا باید "Override" را یاد بگیریم؟

توجه داشته باشید که در دنیای برنامه‌نویسی، زمان بسیار ارزشمند است. هرچقدر بتوانید کد

خود را به صورت کارآمدتر بنویسید، زمان بیشتری برای حل مشکلات دیگر خواهید داشت .

"Override" یکی از این ابزارهای کارآمد در سی شارپ است .

نحوه استفاده از "Override"

برای مثال، فرض کنید یک کلاس Animal دارید که دارای متد Speak () است. اکنون

فرض کنید که شما یک کلاس فرزند با نام Dog از کلاس Animal می‌سازید. اگر

می‌خواهید که کلاس Dog به صورت خاص خود حرکت کند، می‌توانید متد Speak () را

"Override" کنید.

```
public class Animal
```

```

{
    public virtual void Speak()
    {
        Console.WriteLine("The animal speaks.");
    }
}

public class Dog : Animal
{
    public override void Speak()
    {
        Console.WriteLine("The dog barks.");
    }
}

```

C#

در اینجا، ما متد `Speak()` را در کلاس `Animal` با کلمه کلیدی `virtual` تعریف کردیم.

این یعنی این متد می‌تواند در کلاس‌های فرزند `"Override"` شود. سپس ما

در کلاس `Dog` متد `Speak()` را با کلمه کلیدی `override` تغییر دادیم.

علت استفاده از متدهای `override` شده :

متدهای `override` شده به سی‌شارپ اجازه می‌دهند تا از ویژگی `runtime polymorphism` بهره ببرد.

`Polymorphism` توانایی ساخت متدهایی است که با توجه به موقعیت، می‌توانند اجرای متفاوتی داشته باشند.

Polymorphism به این دلیل برای برنامه‌نویسی شی‌گرا اهمیت دارد که به یک کلاس کلی، اجازه می‌دهد متدهایی داشته باشد که در همه‌ی کلاس‌های مشتق شده از آن کلاس، مشترک هستند.

این درحالی است که به `derived class` ها این اجازه را می‌دهد تا هرطور که می‌خواهند آن متدها را اجرا کنند.

در صورت نیاز، نحوه‌ی اجرای آن متدها را تغییر دهند.

متدهای `override` شده، روش دیگری برای اجرای این جنبه از `polymorphism` که می‌گوید “one interface, multiple methods” هستند.