

你确实应该学习并使用的 10 个 C# 特性 - 文章



1) async / await

使用 async / await 模式，可以在执行代码块操作的时候不会阻塞 UI 或者当前的线程。即使该操作被某些执行动作延迟了（比如一个 web 请求），async / await 模式也会继续执行后续的代码。

更多关于 async / await 模式的内容，请见：<https://msdn.microsoft.com/zh-cn/library/hh191443.aspx>

2) 对象 / 数组 / 集合的初始值设定项 (initializers)

通过使用对象、数组、集合的初始值设定项，可以很容易地创建类、数组和集合的实例：

C#// 示例类

```
public class Employee {  
    public string Name {get; set;}  
    public DateTime StartDate {get; set;}  
}
```

// 使用初始值设定项创建员工实例

```
Employee emp = new Employee {Name="John Smith", StartDate=DateTime.Now()};
```

上述示例中的代码在单元测试中可能非常有帮助，不过在有些情况下也应该避免使用它，比如应该通过构造函数来进行类的实例化的时候。

更多关于初始值设定项的内容，请见：<https://msdn.microsoft.com/zh-cn/library/bb384062.aspx>

3) Lambda 表达式、谓词委托 (predicates)、委托 (delegates) 和闭包 (closures)

这些特性在很多情况下都是非常必要的（比如使用 Linq 的时候），请务必学习在何时以及如何使用它们。

更多关于 Lambda 表达式、谓词委托、委托和闭包的内容，请

见：<http://www.codeaddiction.net/articles/13/lambda-expressions-delegates-predicates-and-closures-in-c>

4) ?? - null 合并运算符 (Null coalescing operator)

当表达式左侧不为 null 的时候，?? 运算符返回其左侧的值，否则返回其右侧的值：

C#

```
public static IEnumerable<int> Power(int number, int exponent)
{
    int result = 1;
    for (int i = 0; i < exponent; i++)
    {
        result = result * number;
        yield return result;
    }
}
```

如果使用得当的话，yield 会变得非常强大。它使得你延迟生成序列中的对象，比如当系统不需要枚举整个集合的时候，可以按需停止。

（译者注：本文中的专有词汇基本都使用了中文版 MSDN 中的翻译）

加入伯乐在线专栏作者。扩大知名度，还能得赞赏！详见《[招募专栏作者](#)》

打赏支持译者翻译更多好文章，谢谢！

2 赞 2 收藏 [评论](#)

关于作者：[Erucy](#)



曾经的SharePoint喵星程序猿（暂时还挂着微软MVP的名头），现在的Azure/.Net/MongoDB/Cordova/前端程序猿，偶尔写小说

合作联系

Email: bd@jobbole.com

QQ: 2302462408 （加好友请注明来意）

更多频道

[小组](#) - 好的话题、有启发的回复、值得信赖的圈子

[头条](#) - 分享和发现有价值的内容与观点

[相亲](#) - 为IT单身男女服务的征婚传播平台

[资源](#) - 优秀的工具资源导航

[翻译](#) - 翻译传播优秀的外文文章

[文章](#) - 国内外的精选文章

[设计](#) - UI, 网页, 交互和用户体验

[iOS](#) - 专注iOS技术分享

[安卓](#) - 专注Android技术分享

[前端](#) - JavaScript, HTML5, CSS

[Java](#) - 专注Java技术分享

[Python](#) - 专注Python技术分享