

yield

yield向编译器指示它所在的方法是迭代器块。

yield并不是.NET中的关键字，所以我们可以将它用于变量名。不过yield return和yield break就是关键词组了。

### yield return

yield return 是迭代器（Iterator Pattern）模式的一种实现，能够将本身不是可迭代集合的对象做成可迭代集合。

```
static void Main(string[] args)
{
    foreach (var item in getNums())
    {
        Console.WriteLine(item);
    }
}
static IEnumerable getNums()
{
    yield return "start";
    yield return 1;
    yield return 2;
    yield return 3;
    yield return 4;
    yield return "end";
}
```

注意事项：

- yield return后面要跟一个表达式，该表达式必须能转换为迭代器所使用的类型；

- yield return只能用于迭代器块中；这个迭代器块即可以放在方法体中，也可以放在运算函数中，还可以用在访问器中（即属性方法）；

- yield return 不能在不安全块中（unsafe）；

- 方法、运算符或访问器的参数不能是ref或out；

- yield return不能出现在catch语句中，也不能出现在带有一个或多个catch子句的try块中；

- yield 语句不能出现在匿名方法中。

### yield break

由于某种原因，需要在最后的元素返回之前就结束掉可以用yield break关键词组。

```
static void Main(string[] args)
{
    foreach (var item in getNums())
    {
        Console.WriteLine(item);
    }
}
static IEnumerable getNums()
{
    yield return "start";
    yield return 1;
    yield return 2;
    yield break;
    yield return 3;
    yield return 4;
    yield return "end";
}
```

```
}
```

这样就不会输出2之下的信息了。

## 总结

关键词组yield return和yield break是在.NET 2.0中引入的。Yield return用于创建一个自动生成的、类型化的枚举器，而yield break则用于达到最终元素之前终止迭代。它们在LINQ查询上也非常有用，LINQ的延迟机制的关键就是在yield上。