5.Yield 关键字

我们在之前学习了 IEnumerable 和 IEnumerator ，算是掌握了 C# 的迭代器模式的实现。

今天我们学习 yield 关键字。

yield 实质是一个语法糖，它让程序员能够更方便的去使用迭代器，通过 yield 你可以直接使用迭代器操作而不需要去实现 IEnumerable 和 IEnumerator，也不需要一个临时的 Collection 来完成迭代。

yield 有两种格式声明

```
yield return <expression>;
yield break;
```

我们通过实现与上堂课功能一样的示例代码，来了解下 yield 工作流程

```
/************************************************************************
 * http://sikiedu.com liangxie
 ************************************************************************/

using System.Collections;
using UnityEngine;

namespace UniRxLesson
{
    public class YieldExample : MonoBehaviour
    {
        private void Start()
        {
            foreach (var empty in FiveTimes())
            {
                Debug.Log("A");
            }
        }
```

```
        private  IEnumerable FiveTimes()
        {
            for (var i = 0; i < 5; i++)
            {
                yield return string.Empty;
            }
        }
    }
}
```

输出结果为

```
A
A
A
A
A
```

与上一堂课结果一样。

每一次 foreach 的循环中都会调用迭代器方法，当 yield return 被执行到时，表达式的值会被返回，同时当前的函数的上下文信息被保存下来。下一次循环执行之时会重新从上一次停止的位置继续执行。你也可以使用 yield break 来终止迭代过程。

yield 关键字除了支持返回 IEnumerable 之外，还支持返回 IEnumerator。

```
/*************************************************************************
 * http://sikiedu.com liangxie
 *************************************************************************/

using System.Collections;
using UnityEngine;

namespace UniRxLesson
{
    public class YieldExample : MonoBehaviour
```

```
    {
        private void Start()
        {
            foreach (var empty in FiveTimes())
            {
                Debug.Log("A");
            }

            var fourTimes = FourTimes();

            while (fourTimes.MoveNext())
            {
                Debug.Log("B");
            }
        }

        private IEnumerable FiveTimes()
        {
            for (var i = 0; i < 5; i++)
            {
                yield return string.Empty;
            }
        }

        private IEnumerator FourTimes()
        {
            for (var i = 0; i < 4; i++)
            {
                yield return string.Empty;
            }
        }
    }
}
```

输出结果为

```
A
A
A
```

```
A
A
B
B
B
B
```

今天的内容就这些。