

4. Select 操作符

在上一堂课，介绍 Where 操作符的时候，不管是 UniRx 还是 LINQ，其查询表达式都多出了一个 select。

```
var oldStudents = from oldStudent in students
                  where oldStudent.Age > 45
                  select oldStudent;
```

但是在链式的写法中，却没有使用其对应的 Select 操作符，这是为什么呢？

```
var oldStudents = students.Where(student => student.Age > 45);
```

原因是，在 查询式写法中，编译器要求必须要在最后 select 一下。

而链式写法，我们可以理解成省略掉了。

不省略的版本如下：

```
var oldStudents = students
    .Where(student => student.Age > 45)
    .Select(student => student);
```

其实就是选择了一下自己。

除了选择自己，其实还可以选择一个其他类型的变量作为结果。

比如：

```
var oldStudentNames = students
    .Where(student => student.Age > 45)
    .Select(student => student.Name);
```

既然提到了 Select 我们就今天就正式学习一下 Select 操作符。

LINQ Select 简介

LINQ 中的 Select 操作符 与 SQL 命令中的 Select 作用相似，但是位置不同，查询表达式中的 select 及所接子句是放在最后并把子句中的变量也就是结果返回回来。

LINQ Select 示例代码

```
/* *****  
 * http://sikiedu.com liangxie  
 * ***** */  
  
using System.Collections.Generic;  
using System.Linq;  
using UnityEngine;  
  
namespace UniRxLesson  
{  
    public class LINQSelectExample : MonoBehaviour  
    {  
        class Student  
        {  
            public string Name;  
  
            public int Age;  
        }  
  
        private void Start()  
        {  
            var students = new List<Student>()  
            {  
                new Student() {Name = "张三", Age = 50},  
                new Student() {Name = "张三", Age = 50},  
                new Student() {Name = "李四", Age = 40}  
            };  
  
            var oldStudentNames = students.Where(student => student.Age > 45)  
                .Select(student => student.Name);  
  
            foreach (var studentName in oldStudentNames)  
            {  
                Debug.Log(studentName);  
            }  
        }  
    }  
}
```

输出结果为:

张三

张三

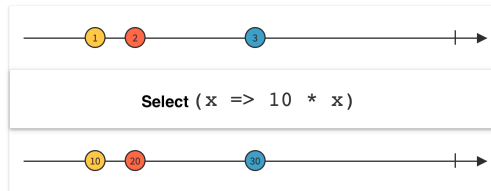
LINQ Select 查询式

```
/* *****  
 * http://sikiedu.com liangxie  
 * ***** */  
  
using System.Collections.Generic;  
using System.Linq;  
using UnityEngine;  
  
namespace UniRxLesson  
{  
    public class LINQSelectQueryExample : MonoBehaviour  
    {  
        class Student  
        {  
            public string Name;  
  
            public int Age;  
        }  
  
        private void Start()  
        {  
            var students = new List<Student>()  
            {  
                new Student() {Name = "张三", Age = 50},  
                new Student() {Name = "张三", Age = 50},  
                new Student() {Name = "李四", Age = 40}  
            };  
  
            var oldStudentNames = from student in students  
                                  where student.Age > 45  
                                  select student.Name;  
  
            foreach (var studentName in oldStudentNames)  
            {  
                Debug.Log(studentName);  
            }  
        }  
    }  
}
```

输出结果为:

张三
张三

UniRx Select 示意图



UniRx Select 在知识地图中的位置



它是非常常用的操作符

UniRx Select 示例代码

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

using UniRx;
using UnityEngine;

namespace UniRxLesson
{
    public class UniRxSelectExample : MonoBehaviour
    {
        void Start()
        {
            Observable.EveryUpdate()
                .Where(_ => Input.GetMouseButtonUp(0))
                .Select(_ => "mouse up")
        }
    }
}
```

```

        .Subscribe(Debug.Log)
        .AddTo(this);
    }
}
}

```

当每次抬起鼠标左键，则输出 mouse up。

UniRx Select 查询式代码

```

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

using UniRx;
using UnityEngine;

namespace UniRxLesson
{
    public class UniRxSelectQueryExample : MonoBehaviour
    {
        void Start()
        {
            var mouseUpEventSteam = from updateEvent in Observable.EveryUpdate()
                                   where Input.GetMouseButtonUp(0)
                                   select "mouse up";

            mouseUpEventSteam
                .Subscribe(Debug.Log)
                .AddTo(this);
        }
    }
}

```

功能与链式的一致。

今天的内容就这些。