

16.Scan

Scan 示意图

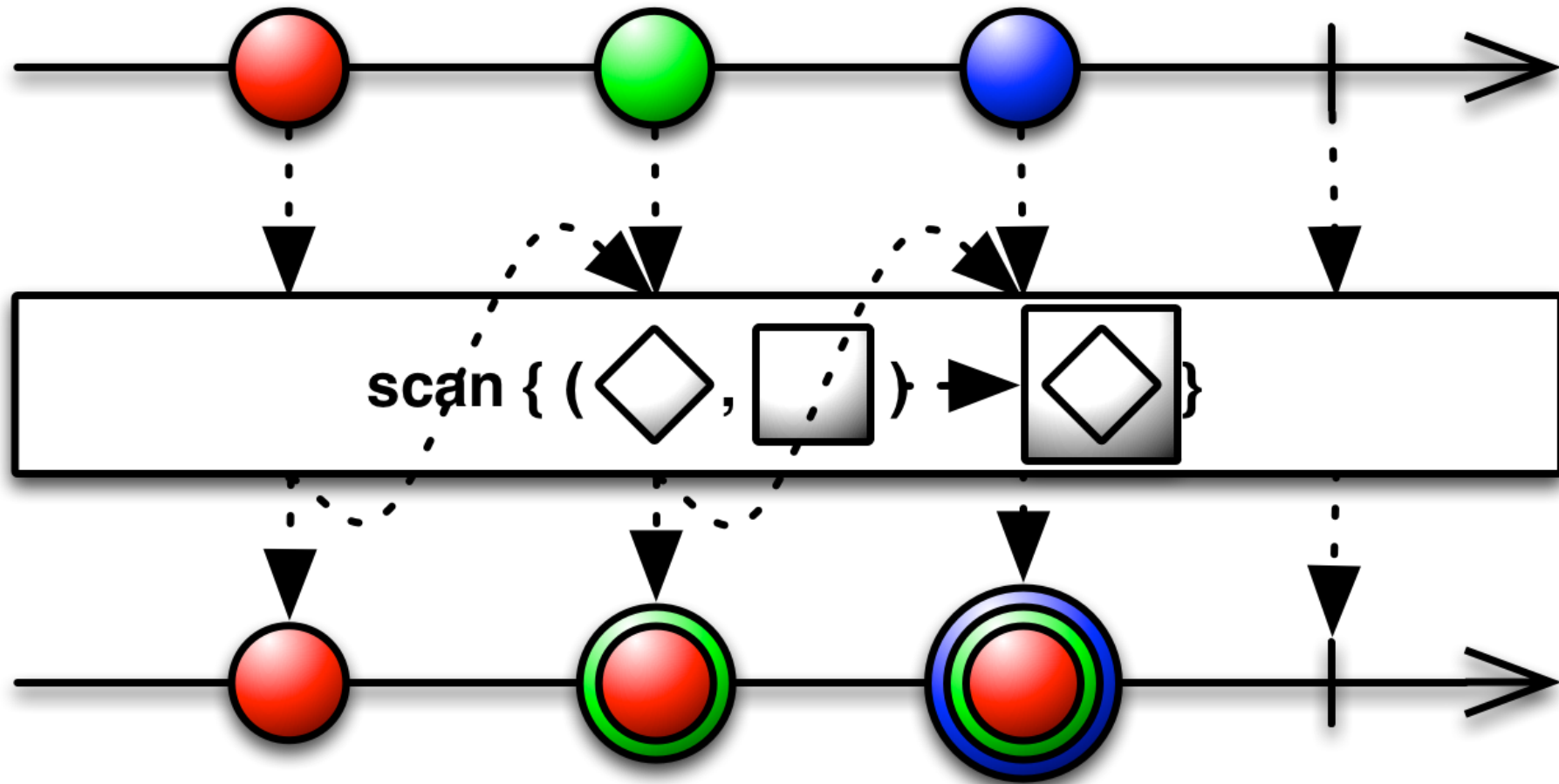
连续地对数据序列的每一项应用一个函数，然后连续发射结果



`scan((x, y) => x + y)`

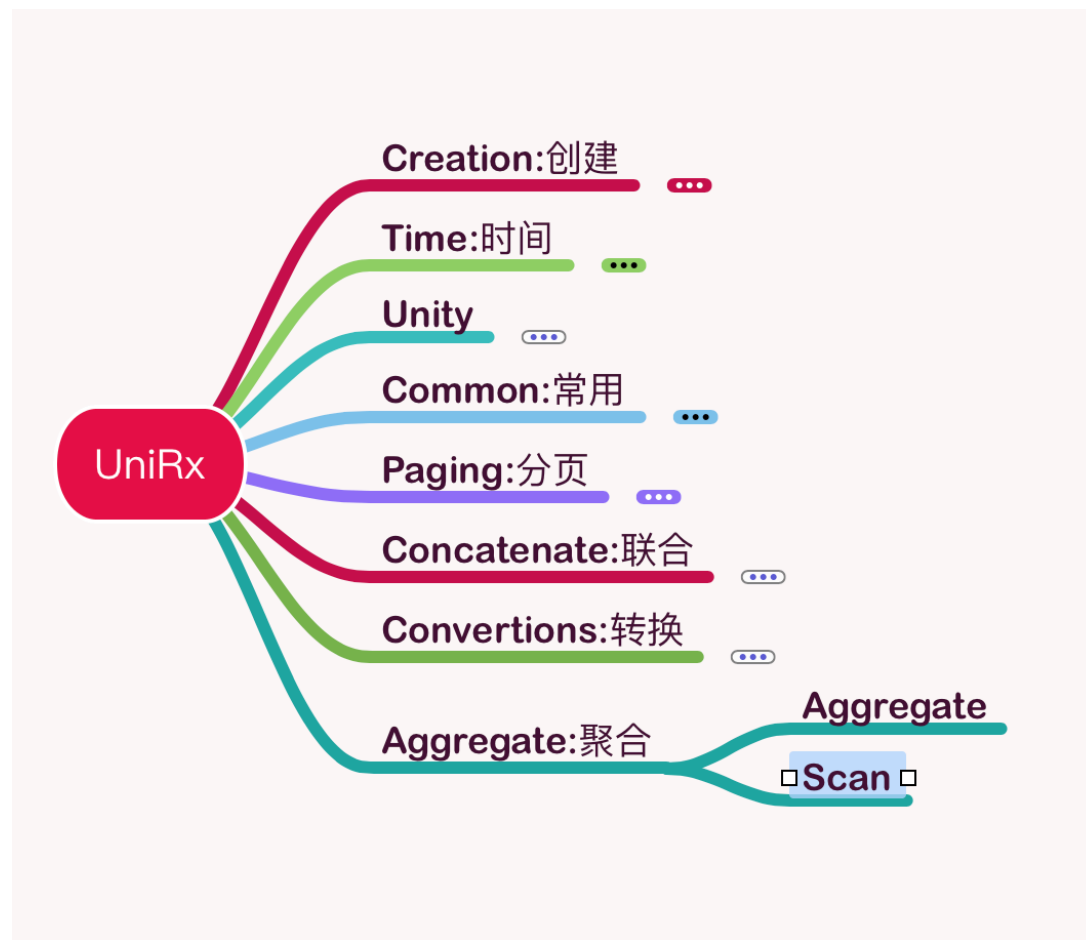


Scan操作符对原始Observable发射的第一项数据应用一个函数，然后将那个函数的结果作为自己的第一项数据发射。它将函数的结果同第二项数据一起填充给这个函数来产生它自己的第二项数据。它持续进行这个过程来产生剩余的数据序列。这个操作符在某些情况下被叫做accumulator。



与 Aggregate 类似，但是 Scan 是每次进行输出，而 Aggregate 则是结束计算后进行输出。

Scan 在知识地图中的位置



Scan 代码示例

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

```
using UnityEngine;  
using UnityEngine;
```

```
namespace UniRxLesson  
{  
    public class ScanExample : MonoBehaviour  
    {  
        private void Start()  
        {  
            Observable.Range(0, 8)  
                .Scan(0, (acc, currentValue) => acc + 5)  
                .Subscribe(xx =>
```



```
    {  
        Debug.Log(xx);  
    });  
}  
}
```

输出结果为

```
5  
10  
15  
20  
25  
30  
35  
40
```

今天的内容就这些