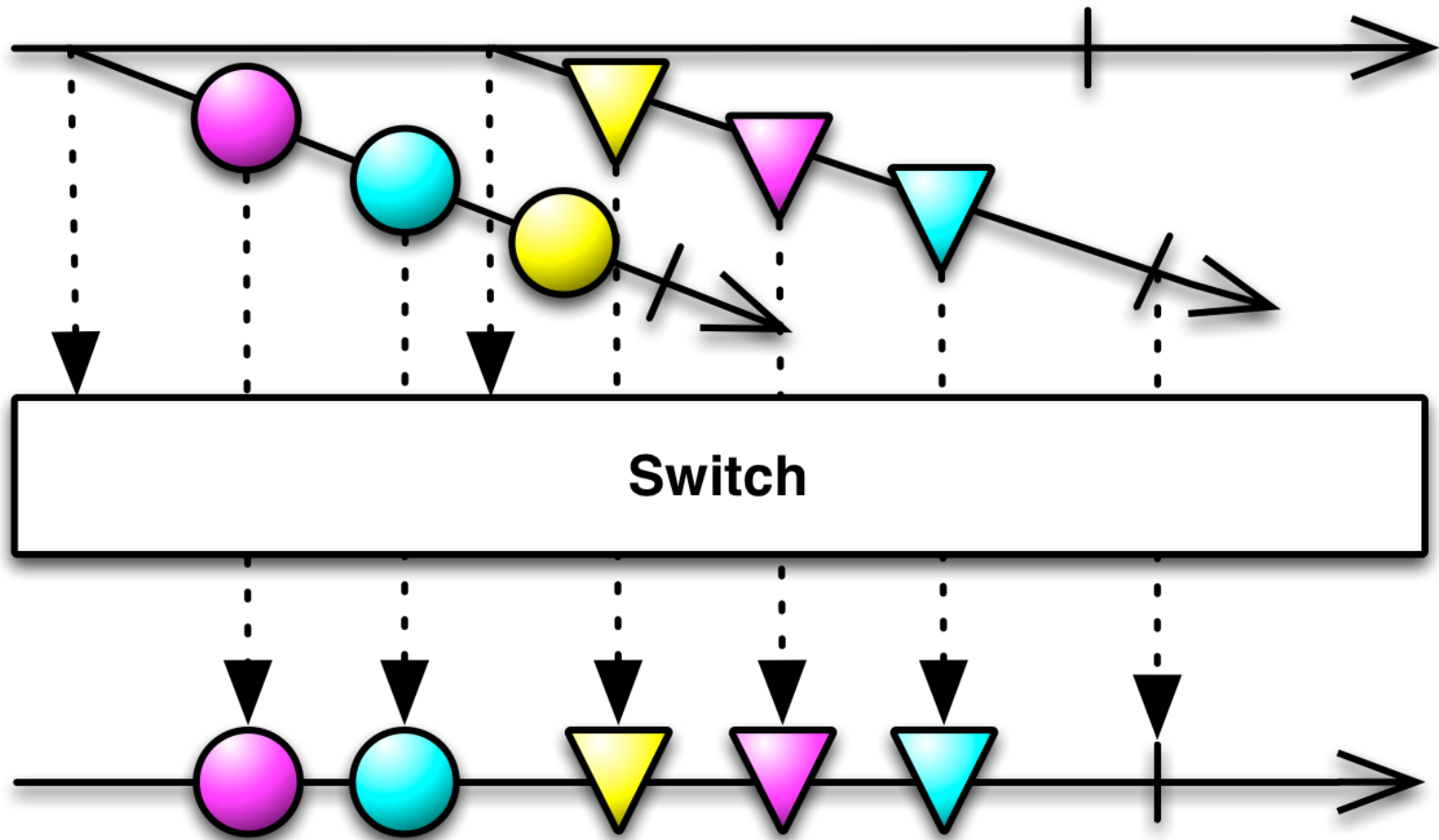


# 17.Switch

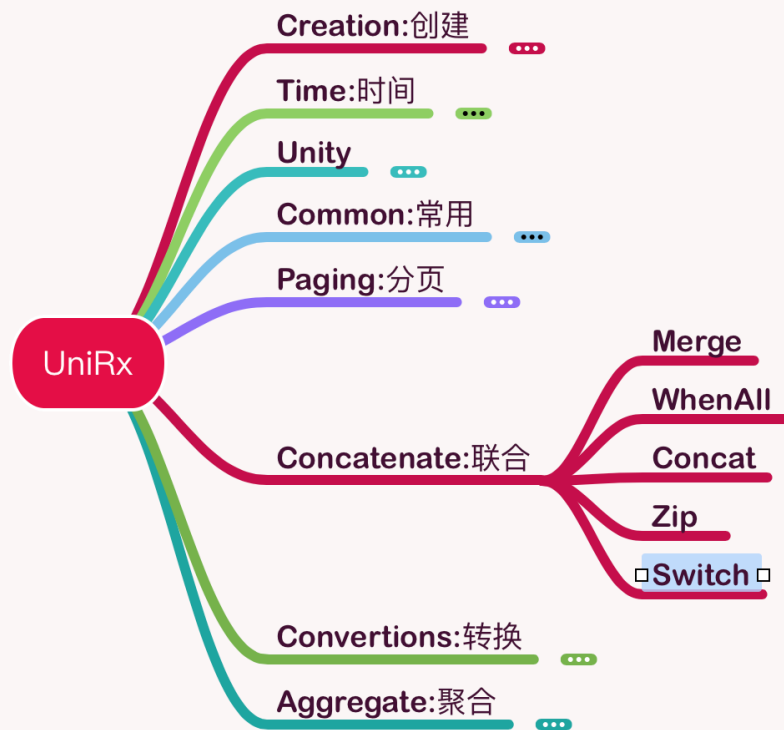
## Switch 示意图

将一个发射多个 Observables 的 Observable 转换成另一个单独的 Observable，后者发射那些 Observables 最近发射的数据项



Switch 订阅一个发射多个 Observables 的 Observable。它每次观察那些 Observables 中的一个，Switch 返回的这个 Observable 取消订阅前一个发射数据的 Observable，开始发射最近的 Observable 发射的数据。注意：当原始 Observable 发射了一个新的 Observable 时（不是这个新的 Observable 发射了一条数据时），它将取消订阅之前的那个 Observable。这意味着，在后来那个 Observable 产生之后到它开始发射数据之前的这段时间里，前一个 Observable 发射的数据将被丢弃（就像图例上的那个黄色圆圈一样）。

# Switch 所在知识地图的位置



# Switch 代码示例

```
/******  
 * http://sikiedu.com liangxie  
******/  
  
using UniRx;  
using UnityEngine;  
  
namespace UniRxLesson  
{  
    public class SwitchExample : MonoBehaviour  
    {  
        private void Start()  
        {  
            var buttonDownStream = Observable.EveryUpdate().Where(_ =>  
Input.GetMouseButtonDown(0));
```

```
        var buttonStream = Observable.Updates().Where(_ =>
Input.GetMouseButtonUp(0));

        buttonDownStream.Select(_ =>
        {
            Debug.Log("mouse button down");
            return buttonStream;
        })
        .Switch()
        .Subscribe(_ => { Debug.Log("mouse button up"); });
    }
}
```

执行结果为，当按下鼠标时输出 “mouse button down” 抬起之后输出 “mouse button up”。

今天的内容就这些