ReactiveCocoa入门到实战

第二周 ReactiveCocoa操作详解

内容大纲

- RACSignal使用基础
- RACSignal各类操作

RACSignal使用基础

RACSignal的使用基础

获得一个信号的方式

- 单元信号

```
RACSignal *signal1 = [RACSignal return:@"Some Value"];
RACSignal *signal2 = [RACSignal error:errorObject];
RACSignal *signal3 = [RACSignal empty];
RACSignal *signal4 = [RACSignal never];
```

- 动态信号

```
RACSignal *signal5 = [RACSignal createSignal:
                      ^RACDisposable *(id<RACSubscriber> subscriber) {
    [subscriber sendNext:@1];
    [subscriber sendNext:@2];
    [subscriber sendError:errorObject];
    [subscriber sendCompleted];
    return [RACDisposable disposableWithBlock:^{

    }];
}];
```

RACSignal的使用基础

获得一个信号的方式

- Cocoa桥接

```
RACSignal *signal6 = [view rac_signalForSelector:@selector(setFrame:)];
RACSignal *signal7 = [view
                            rac_signalForControlEvents:UIControlEventTouchUpInside];
RACSignal *signal8 = [view rac_willDeallocSignal];
RACSignal *signal9 = RACObserve(view, backgroundColor);
```

- 信号变化

```
RACSignal *signal10 = [signal1 map:^id(NSString *value) {
    return [value substringFromIndex:1];
}];
```

- 序列转换

```
RACSignal *signal11 = sequence.signal;
```

RACSignal的使用基础

订阅一个信号的方式

- 订阅方法

```
[signal11 subscribeNext:^(id x) {
    NSLog(@"next value is %@", x);
} error:^(NSError *error) {
    NSLog(@"Ops! Get some error: %@", error);
} completed:^{
    NSLog(@"It finished success");
}];
```

- 绑定

```
RAC(view, backgroundColor) = signal10;
```

- Cocoa桥接

```
[view rac_liftSelector:@selector(convertPoint:toView:)
           withSignals:signal1, signal2, nil];
[view rac_liftSelector:@selector(convertRect:toView:)
  withSignalsFromArray:@[signal3, signal4]];
[view rac_liftSelector:@selector(convertRect:toLayer:)
 withSignalOfArguments:signal5];
```

RACSignal的使用基础

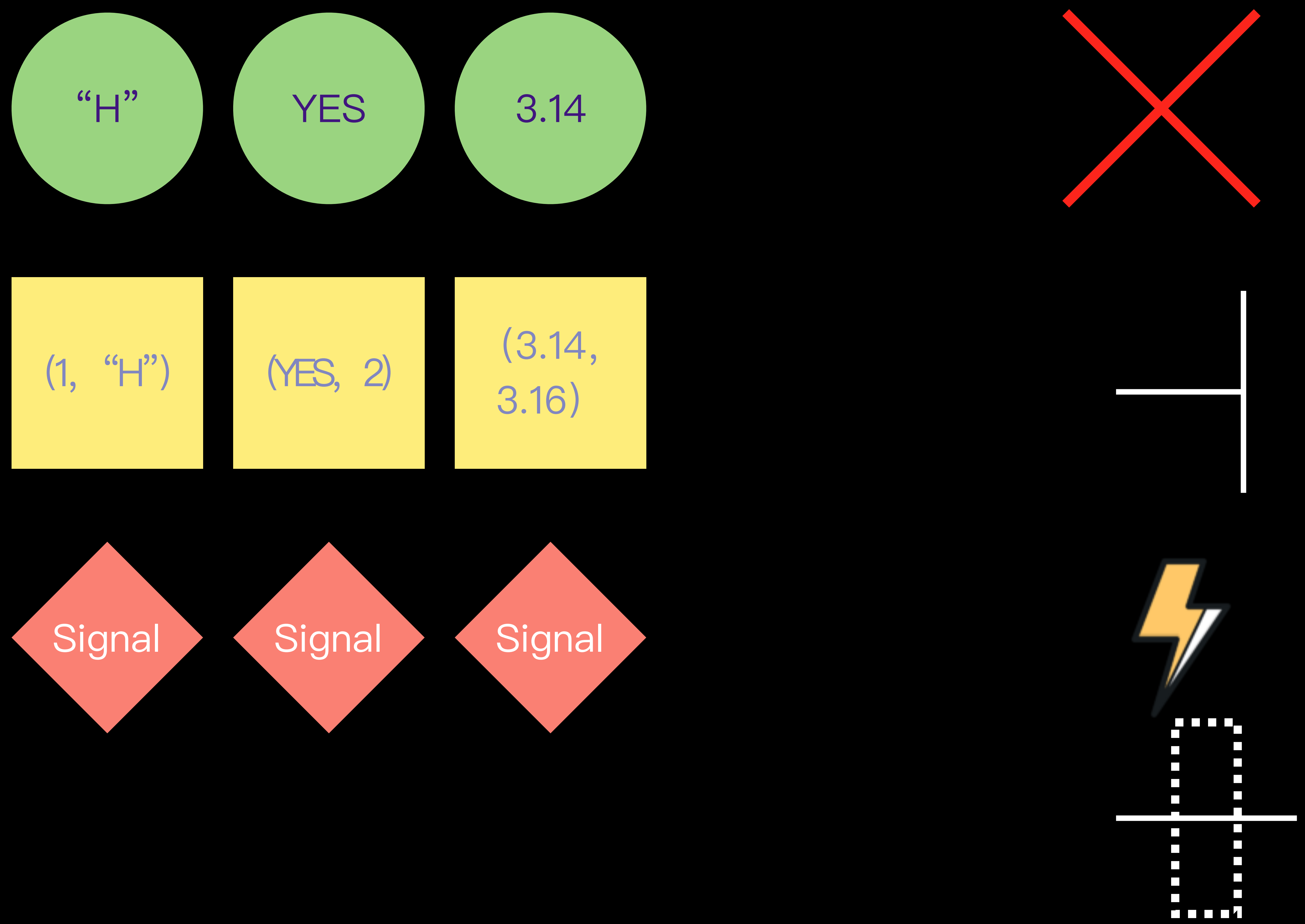
订阅过程

```
RACSignal *signal = [RACSignal createSignal:
                     ^RACDisposable *(id<RACSubscriber> subscriber)
{
    [subscriber sendNext:@1];
    [subscriber sendNext:@2];
    [subscriber sendCompleted];
    return [RACDisposable disposableWithBlock:^{
        NSLog(@"dispose");
    }];
}];

RACDisposable *disposable = [signal subscribeNext:^(id x) {
    NSLog(@"next value is %@", x);
} error:^(NSError *error) {
    NSLog(@"Ops! Get some error: %@", error);
} completed:^{
    NSLog(@"It finished success");
}];

[disposable dispose];
```

RACSignal的使用基础

事件类型 & 图例

- 值
- 错误
- 结束
- 订阅
- 取消订阅



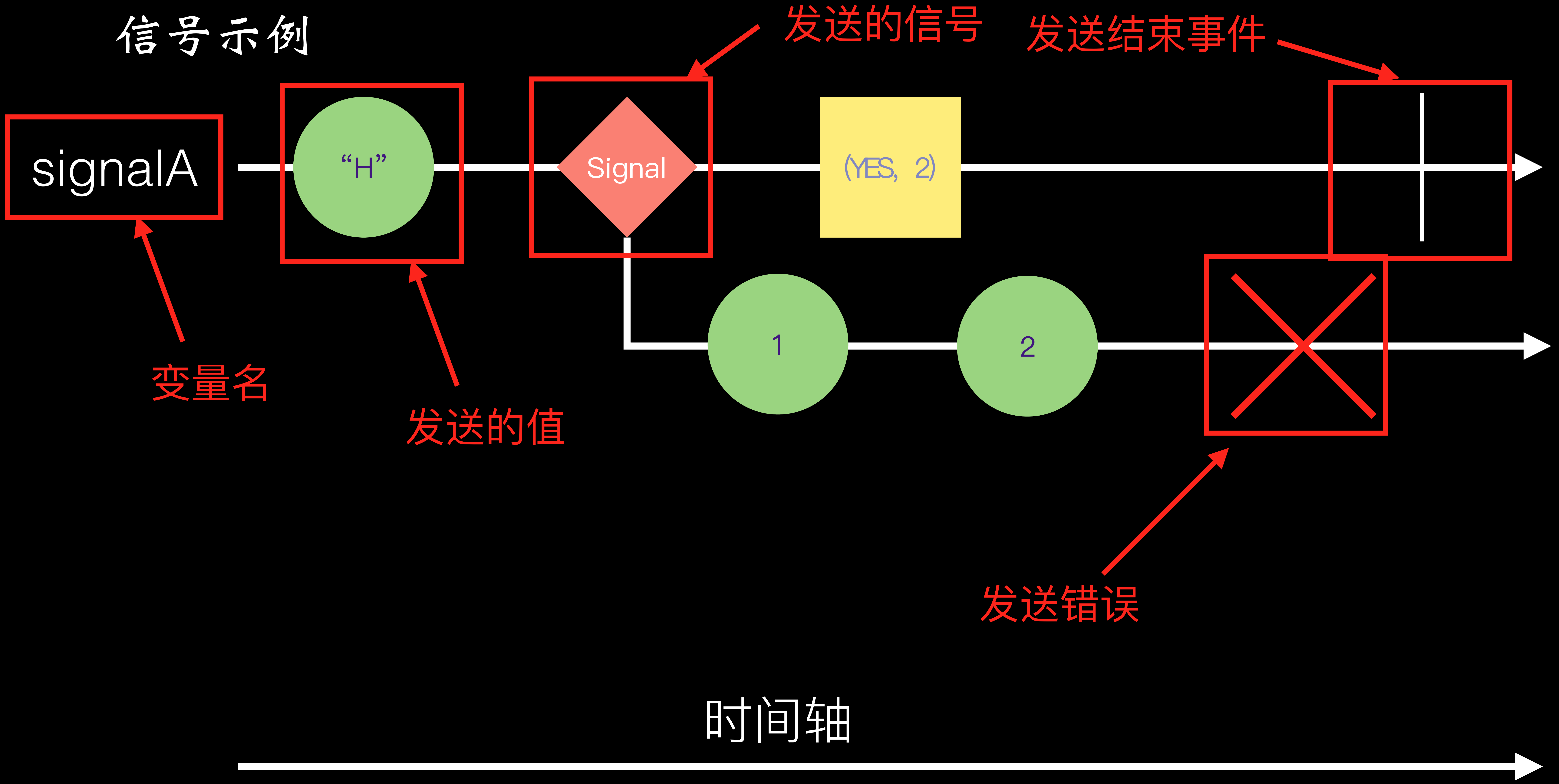
RACSignal的使用基础

元组——RACTuple

- RAC定义的一种数据类型
- NSArray的简化版
- 其他语言中的意义
- 使用

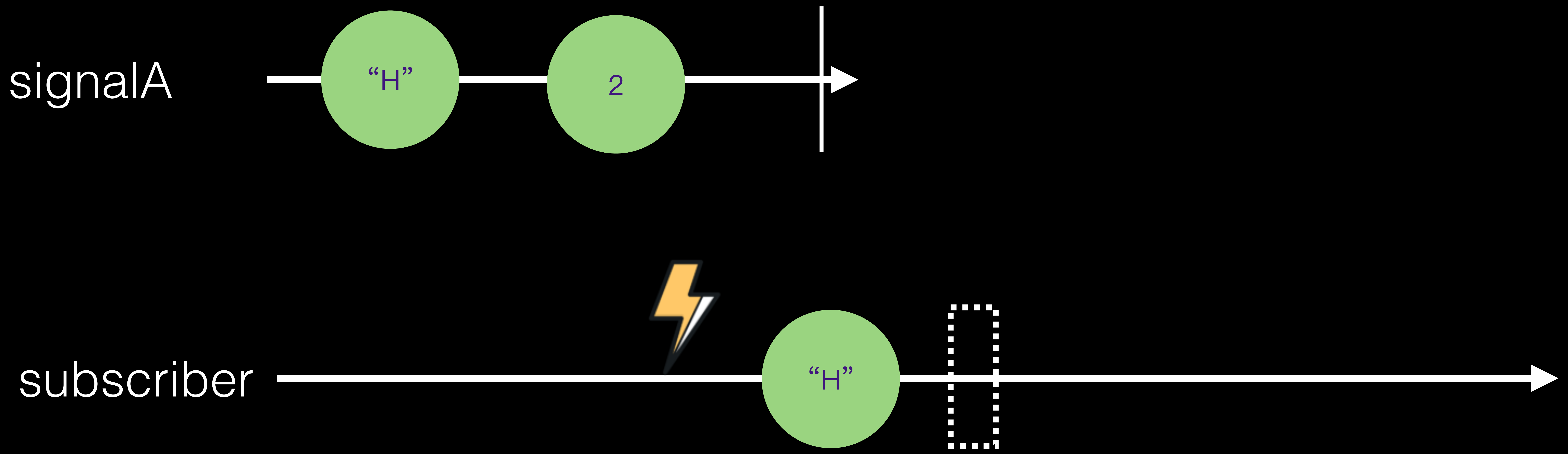
```
RACTuple *tuple = RACTuplePack(@1, @"haha");

id first = tuple.first;
id second = tuple.second;
id last = tuple.last;

id index1 = tuple[1];

RACTupleUnpack(NSNumber *num, NSString *str) = tuple;
```

RACSignal的使用基础



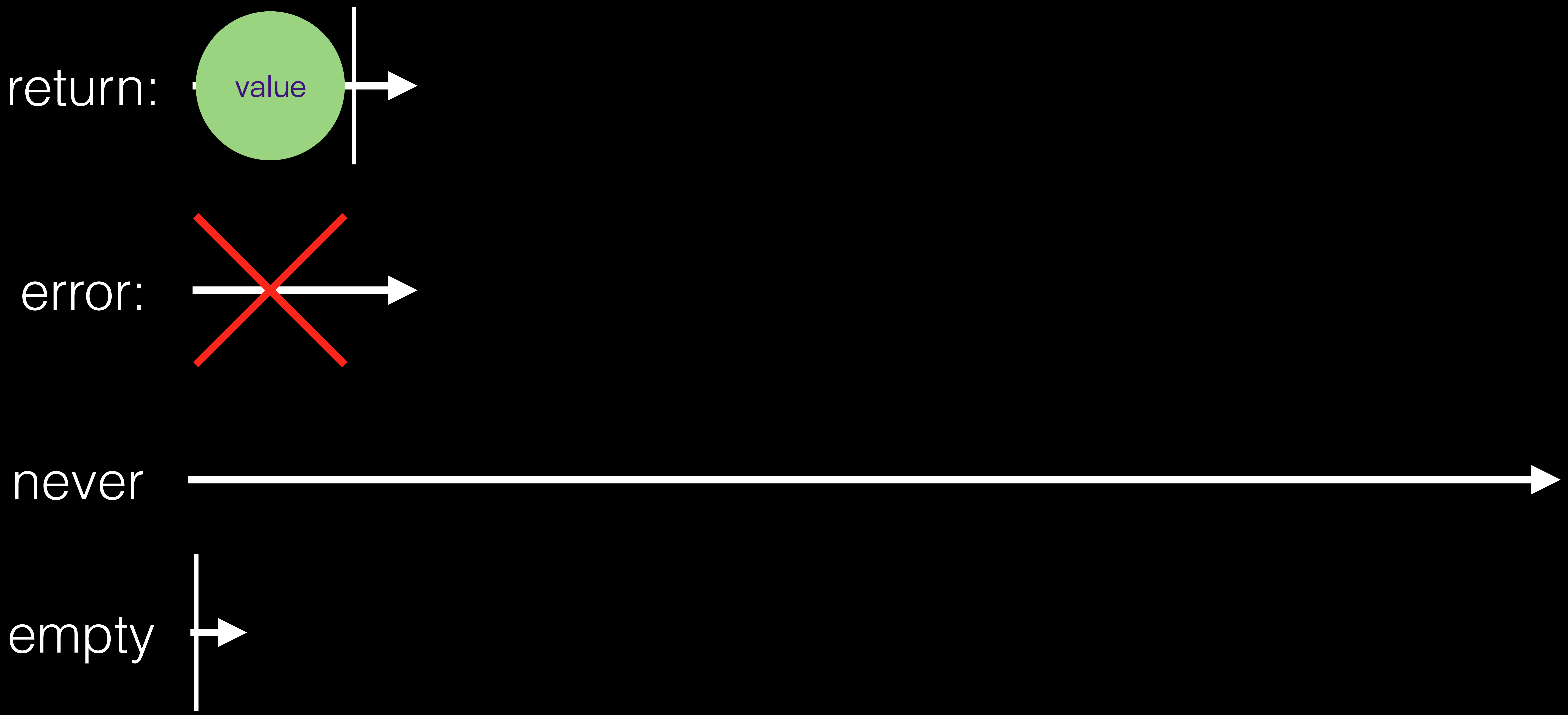
RACSignal的使用基础

信号定义 && 信号订阅



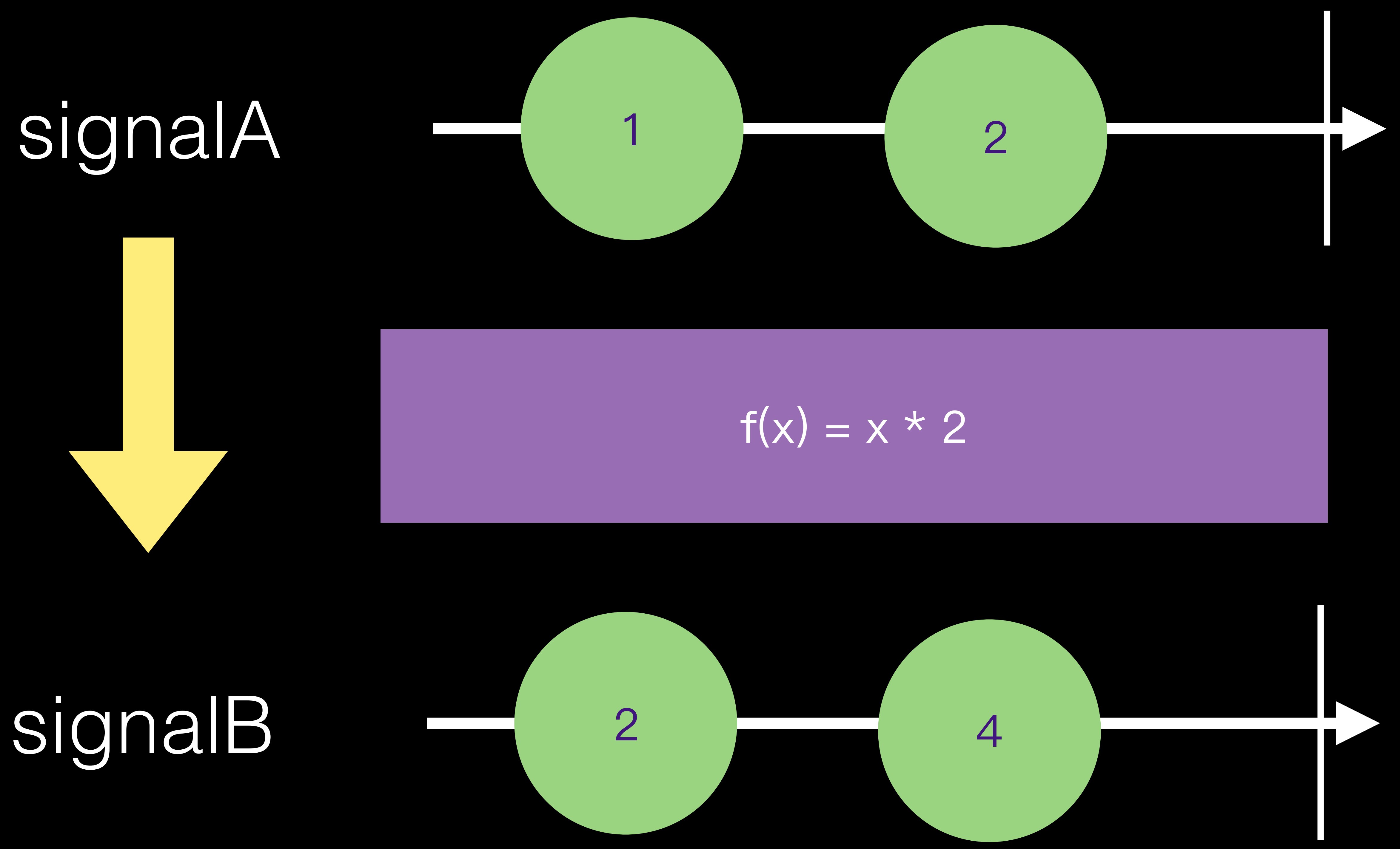
RACSignal的使用基础

单元信号



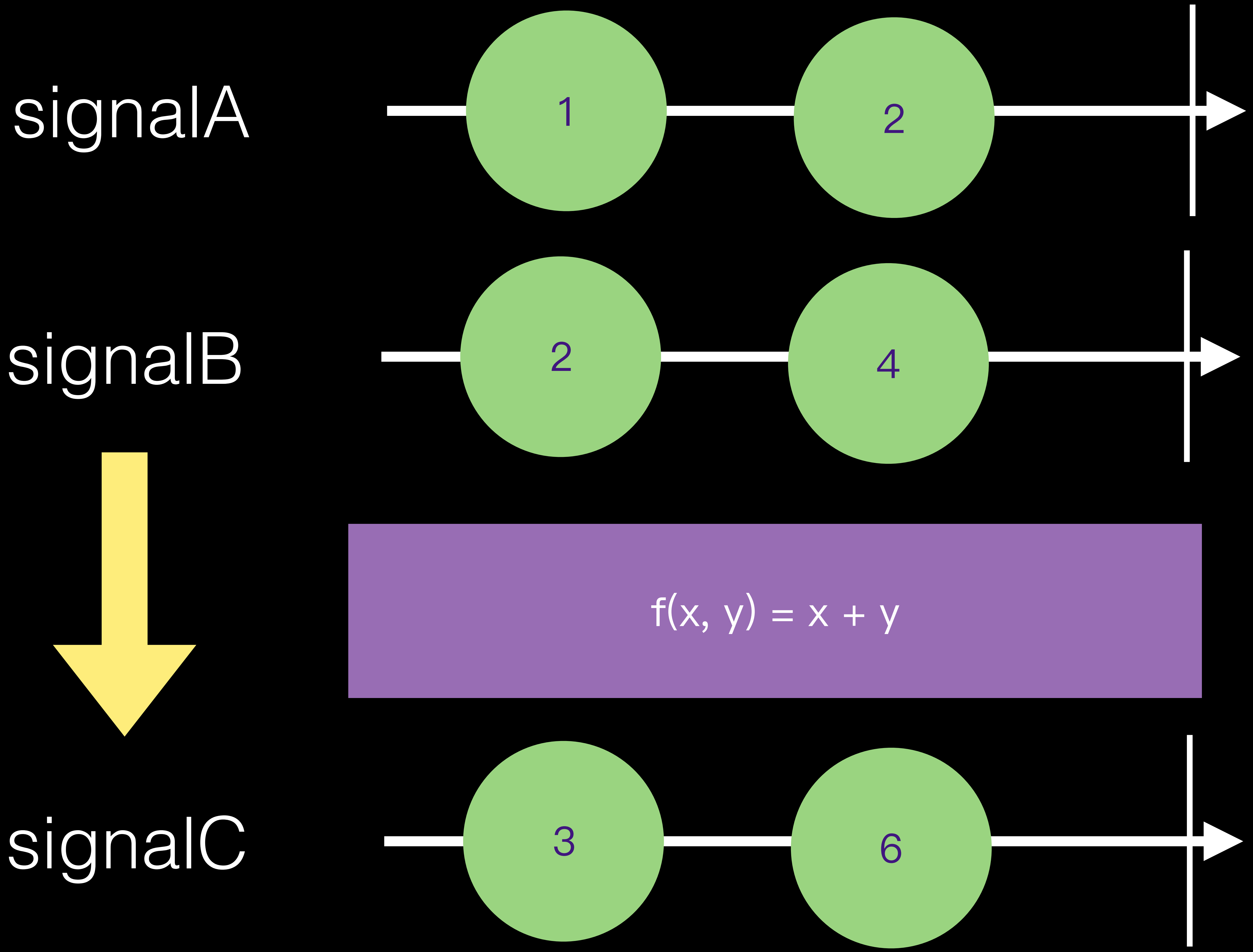
RACSignal的使用基础

信号的变换和组合



RACSignal的使用基础

信号的变换和组合



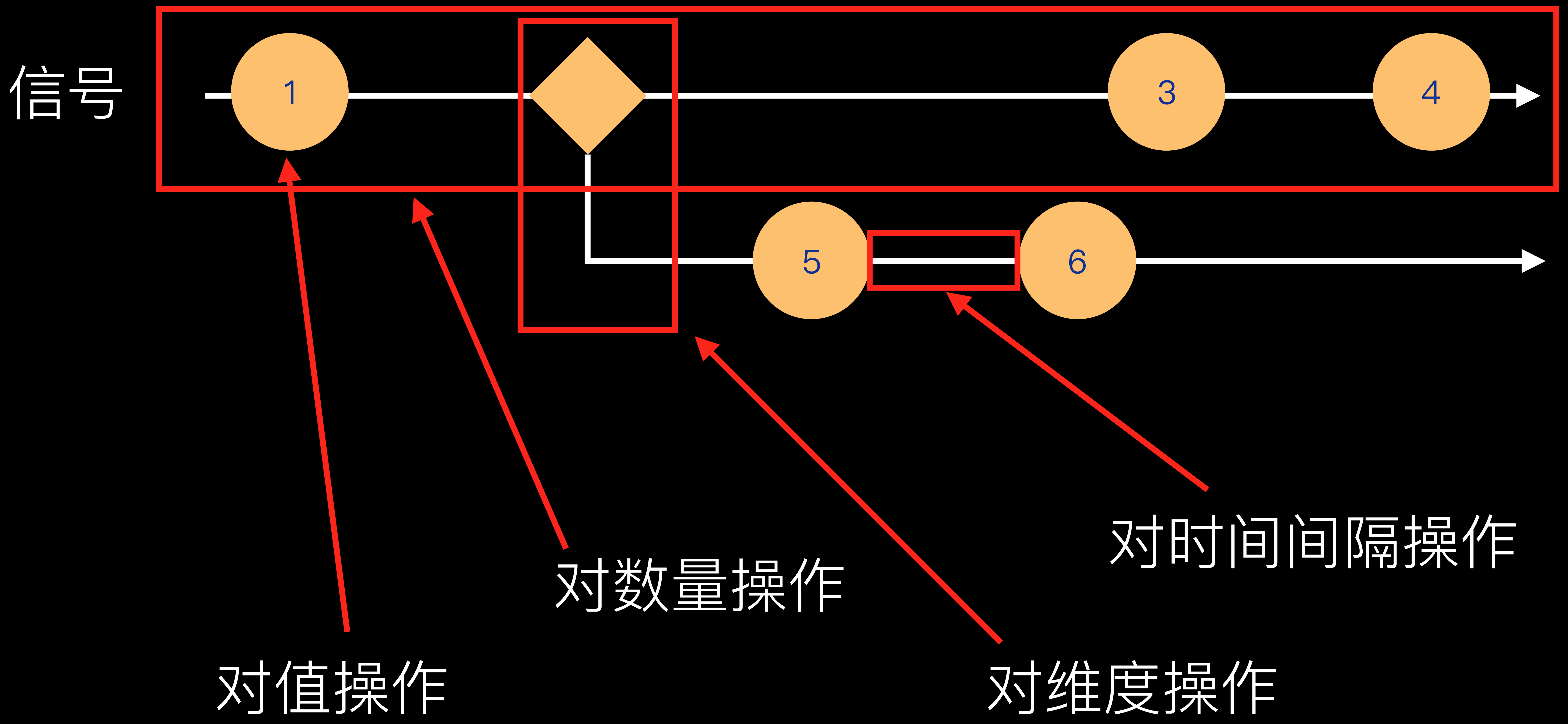
RACSignal各类操作

RACSignal各类操作

- 单个信号的变换
- 多个信号的组合
- 高阶操作

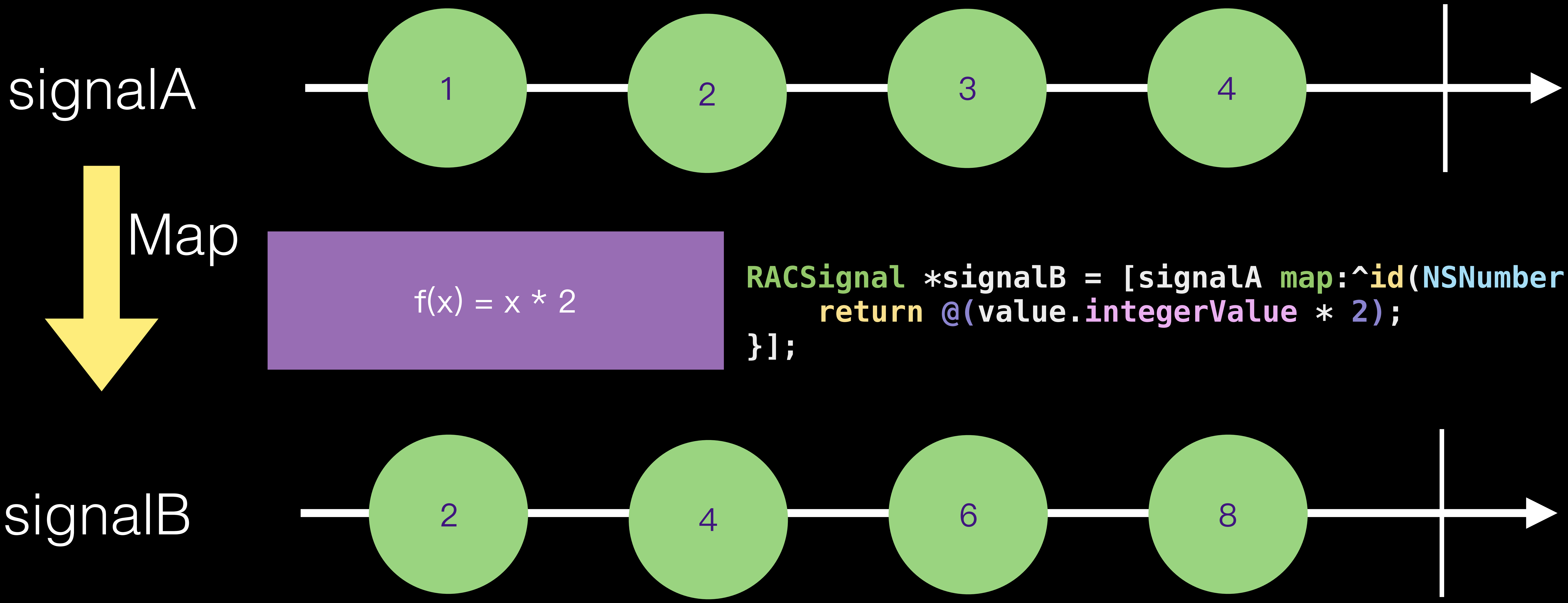
RACSignal各类操作

单个信号的变换



RACSignal各类操作

值操作——Map



```
RACSignal *signalB = [signalA map:^id(NSNumber *value) {
    return @(value.integerValue * 2);
}];
```

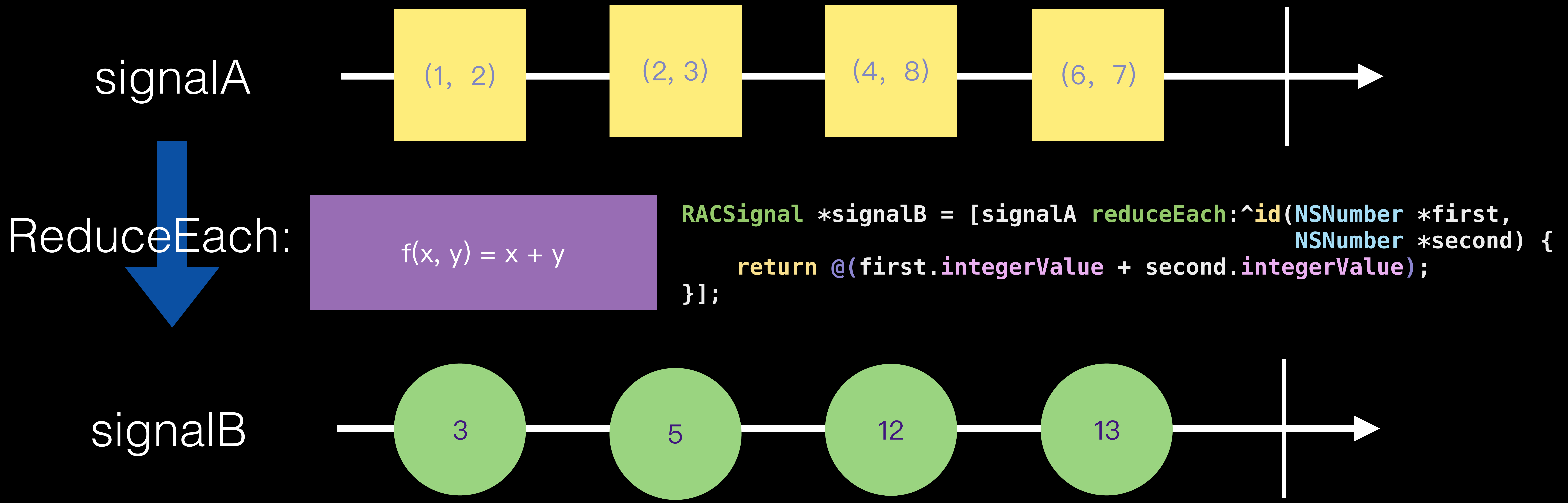
RACSignal各类操作

值操作——MapReplace

```
RACSignal *signalB = [signalA map:^id(id value) {
    return @8;
}]; // signalB is --8--8--8--8--|

RACSignal *signalC = [signalA mapReplace:@8];
// signalC is --8--8--8--8--| too.
```

RACSignal各类操作

值操作——ReduceEach:



```
RACSignal *signalB = [signalA reduceEach:^id(NSNumber *first,
                                             NSNumber *second) {
    return @(first.integerValue + second.integerValue);
}];
```

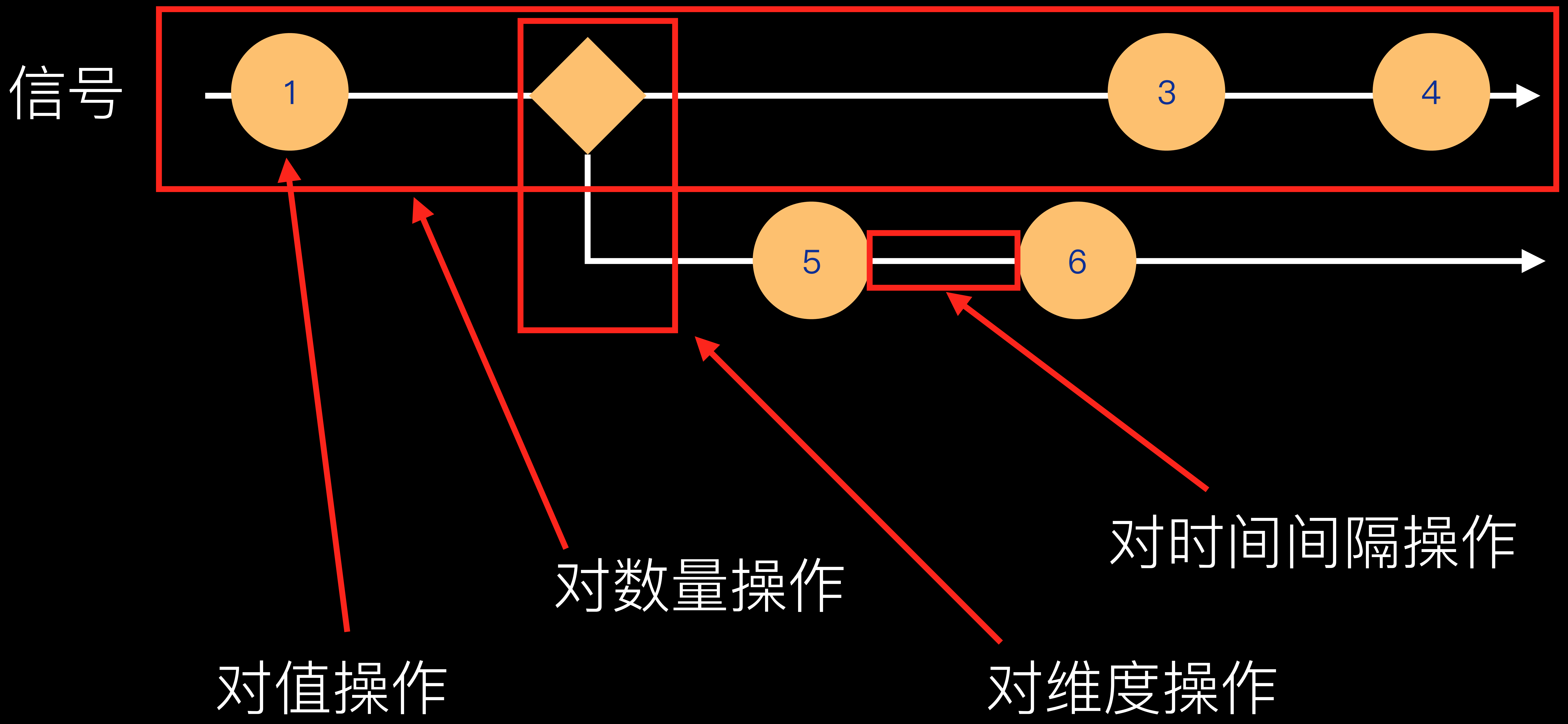
RACSignal各类操作

值操作——其他

```
-(RACSignal *)not;
-(RACSignal *)and;
-(RACSignal *)or;
-(RACSignal *)reduceApply;
-(RACSignal *)materialize;
-(RACSignal *)dematerialize;
```

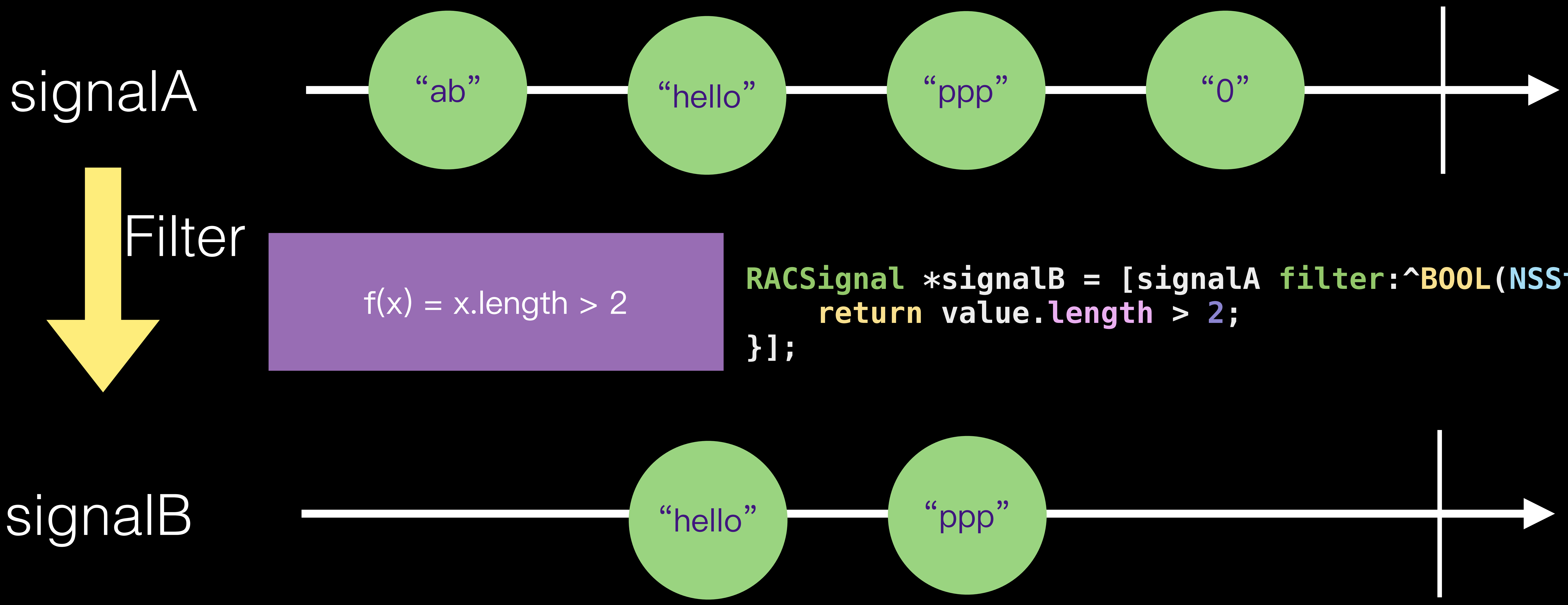
RACSignal各类操作

单个信号的变换



RACSignal各类操作

数量操作——Filter



```
RACSignal *signalB = [signalA filter:^BOOL(NSString *value) {
    return value.length > 2;
}];
```

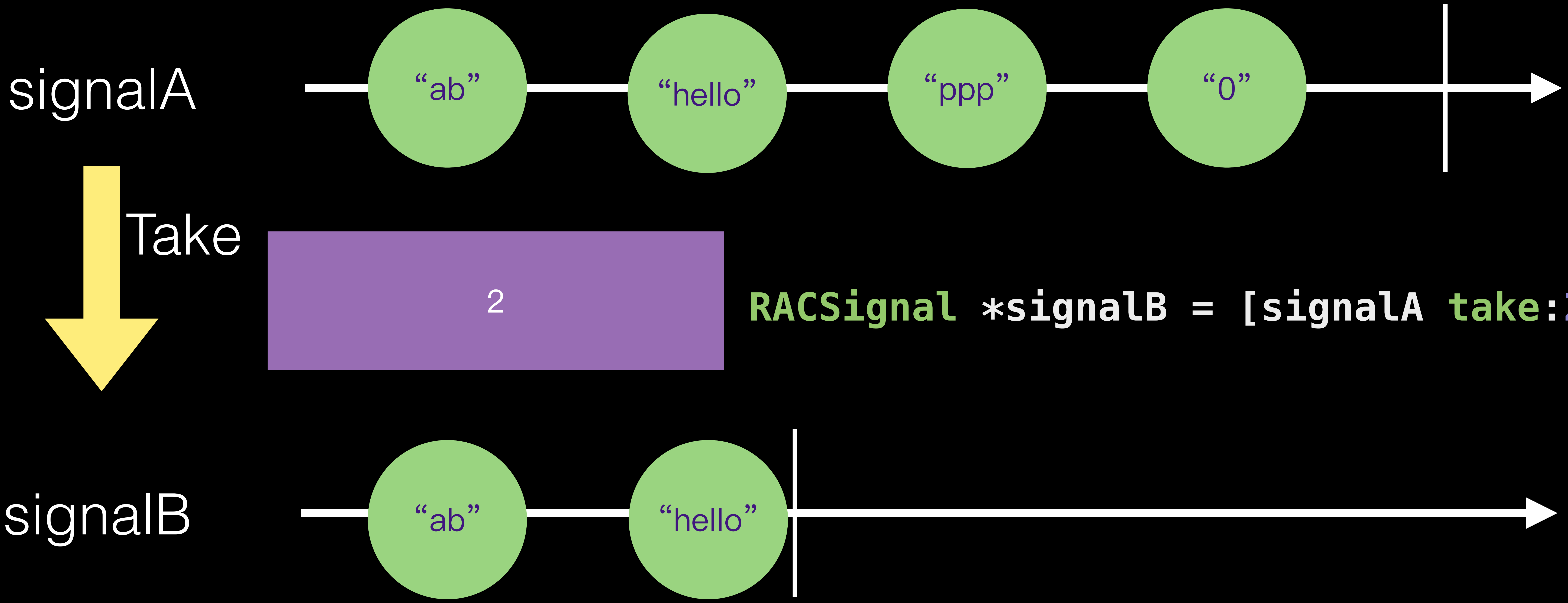
RACSignal各类操作

数量操作——Ignore

```
RACSignal *signalB = [signalA filter:^BOOL(id value) {
    return ![@1 isEqual:value];
}];

RACSignal *signalC = [signalA ignore:@1];


- (RACSignal *)ignoreValues;
- (RACSignal *)distinctUntilChanged;
```

RACSignal各类操作

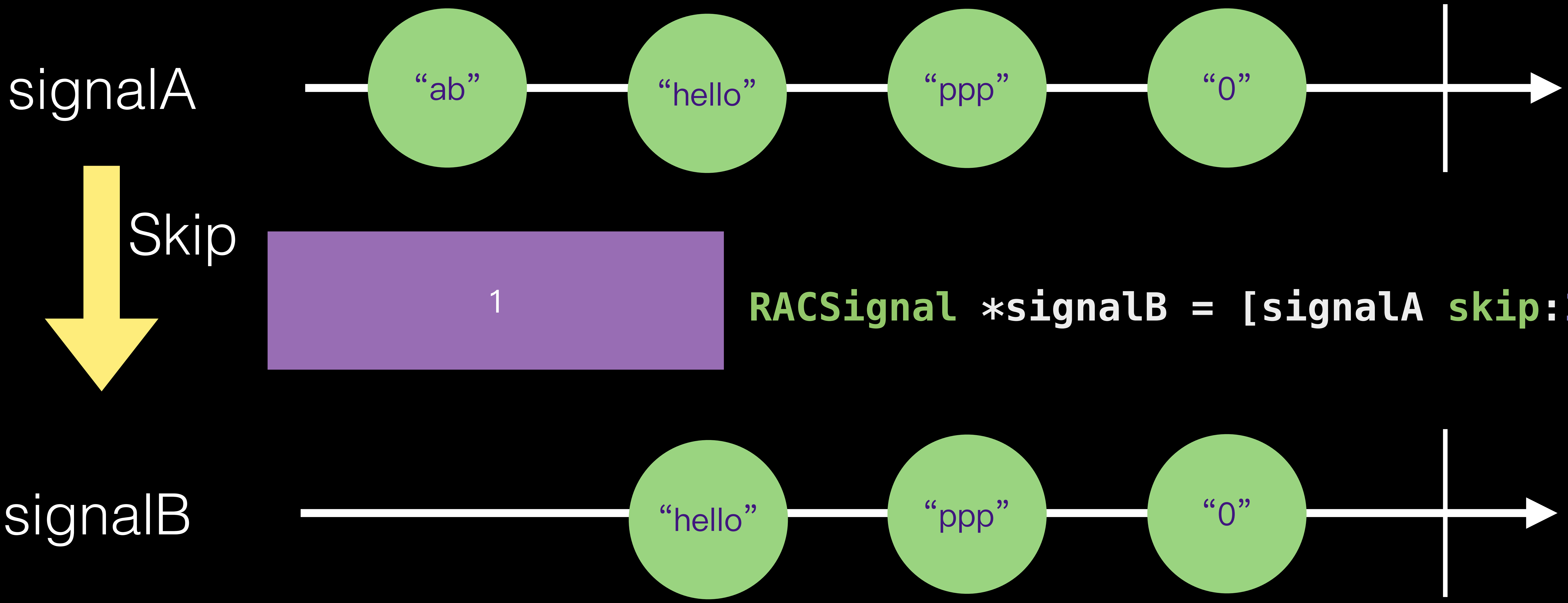
数量操作——Take



```
RACSignal *signalB = [signalA take:2];
```

RACSignal各类操作

数量操作——Skip



```
RACSignal *signalB = [signalA skip:1];
```

RACSignal各类操作

数量操作——Take&Skip其他

```
- (RACSignal *)takeLast:(NSUInteger)count;
- (RACSignal *)takeUntilBlock:(BOOL (^)(id x))predicate;
- (RACSignal *)takeWhileBlock:(BOOL (^)(id x))predicate;
- (RACSignal *)skipUntilBlock:(BOOL (^)(id x))predicate;
- (RACSignal *)skipWhileBlock:(BOOL (^)(id x))predicate;
```

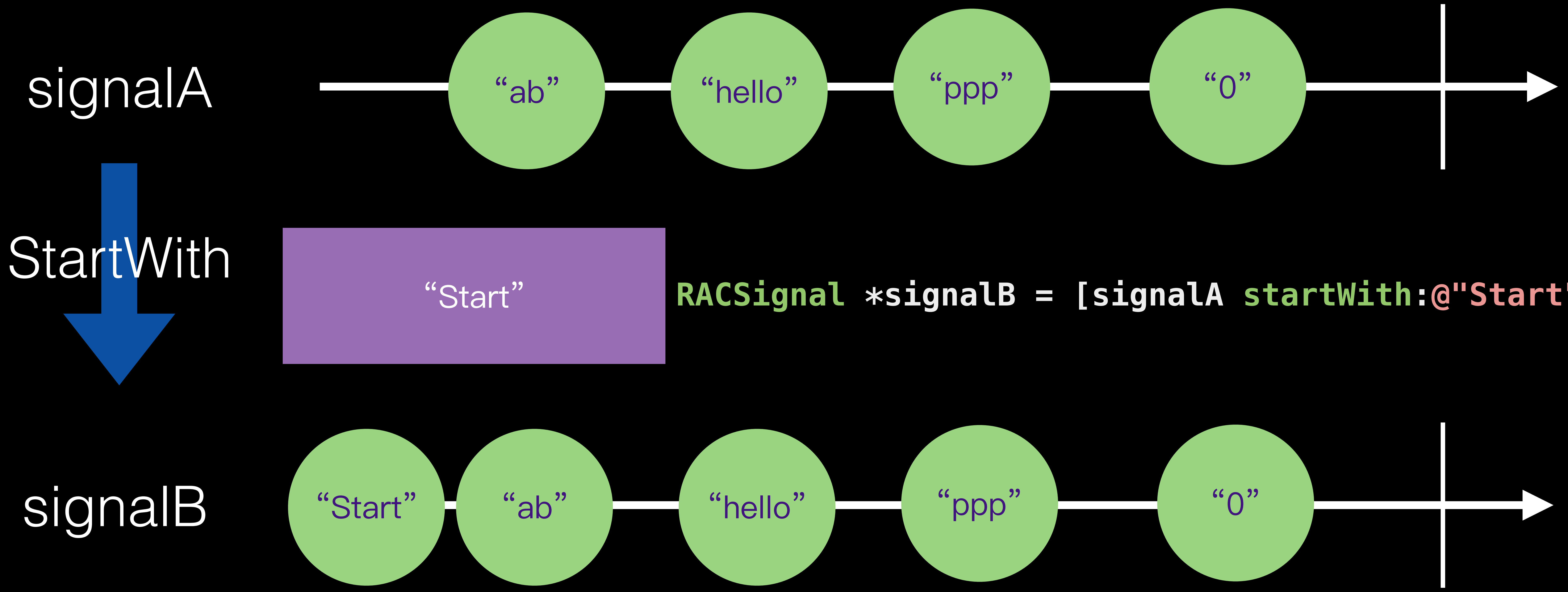
RACSignal各类操作

一些混合操作

```
- (RACSignal *)any;
- (RACSignal *)any:(BOOL (^)(id object))predicateBlock;
- (RACSignal *)all:(BOOL (^)(id object))predicateBlock;
```

RACSignal各类操作

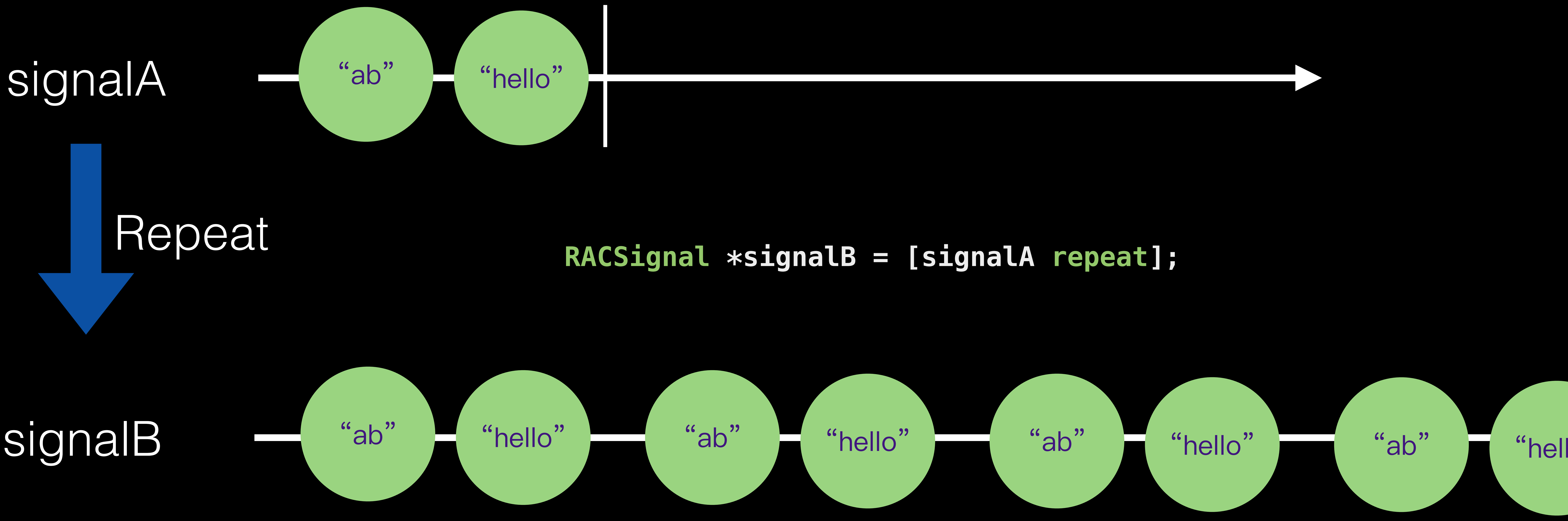
数量操作——StartWith



```
RACSignal *signalB = [signalA startWith:@"Start"];
```

RACSignal各类操作

数量操作——Repeat



```
RACSignal *signalB = [signalA repeat];
```

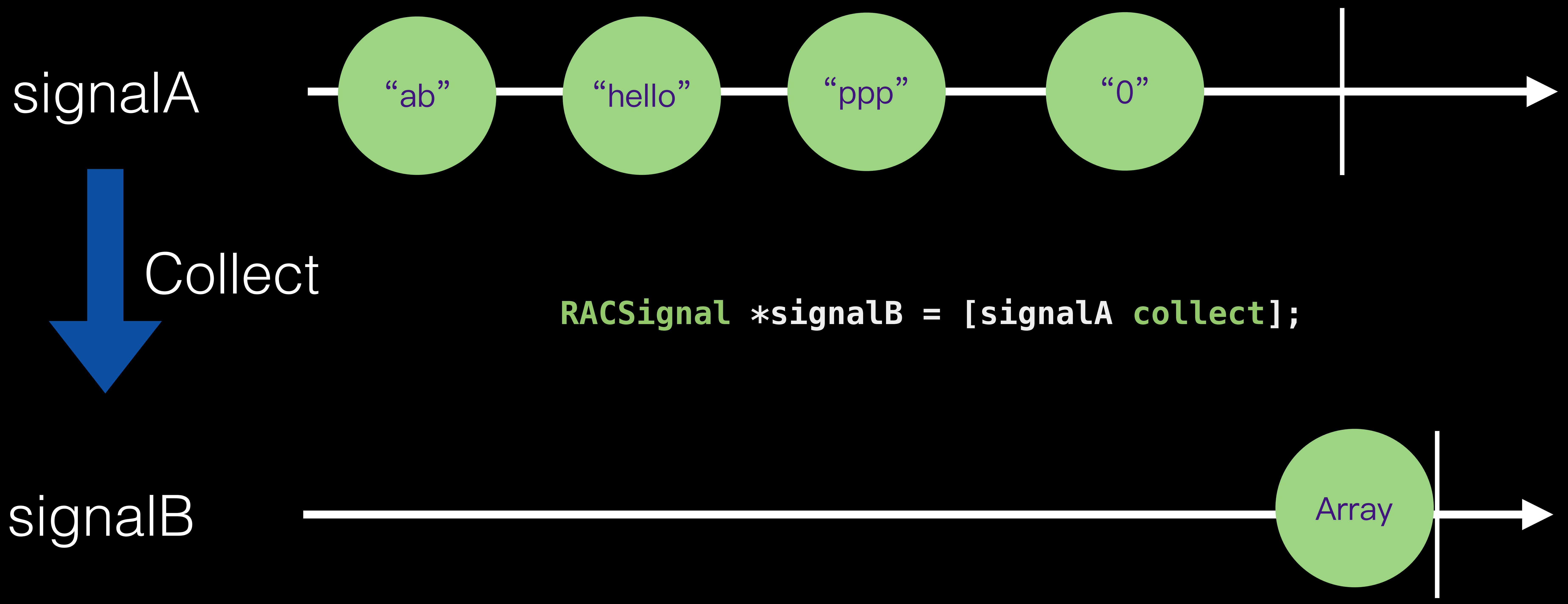
RACSignal各类操作

副作用操作

```
RACSignal *signalB = [signalA map:^id(id value) {
    // do some thing;
    return value;
}];

RACSignal *signalC = [signalA doNext:^(id x) {
    // do some thing;
}];


- (RACSignal *)doError:(void (^)(NSError *error))block;
- (RACSignal *)doCompleted:(void (^)(void))block;
```

RACSignal各类操作

数量操作——Collect



```
RACSignal *signalB = [signalA collect];
```

RACSignal各类操作

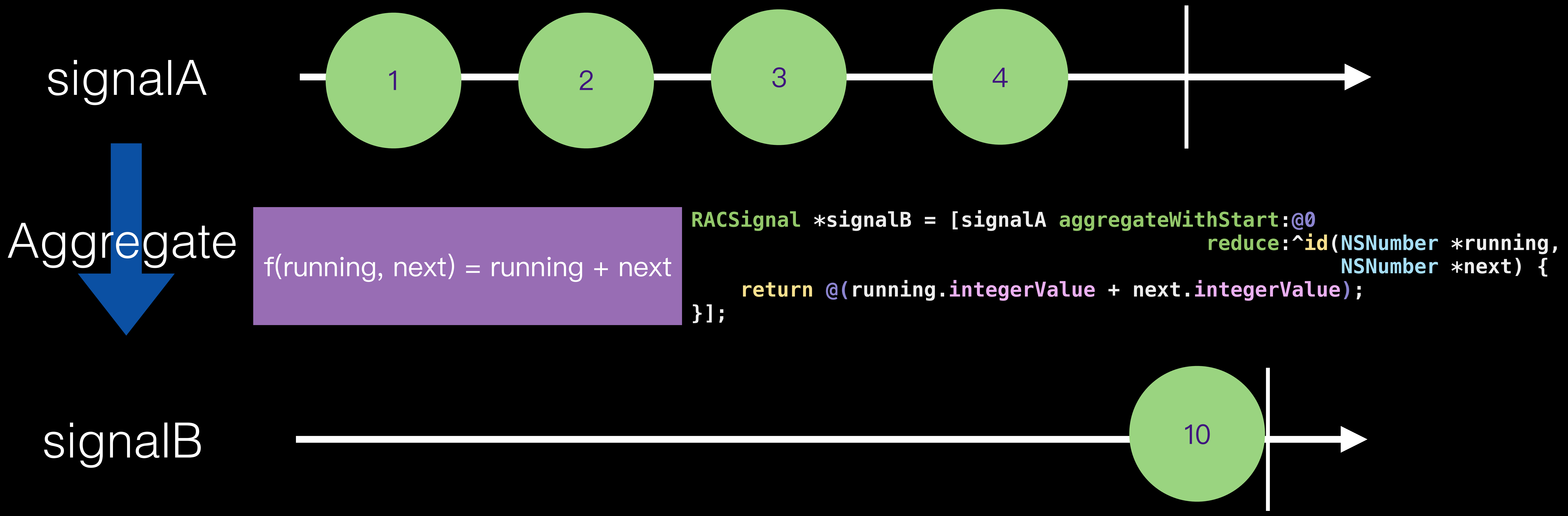
数量操作——Aggregate

折叠函数

```
int fold(int *array, int count, FoldFunction func, int start)
{
    int current = array[0];
    int running = func(start, current);
    if (count == 1) {
        return running;
    }
    return fold(array + 1, count - 1, func, running);
}

int arr[] = {1, 2, 3, 4, 5};
int result = fold(arr, 5, ^int(int running, int next) {
    return running + next;
}, 0);
```

RACSignal各类操作

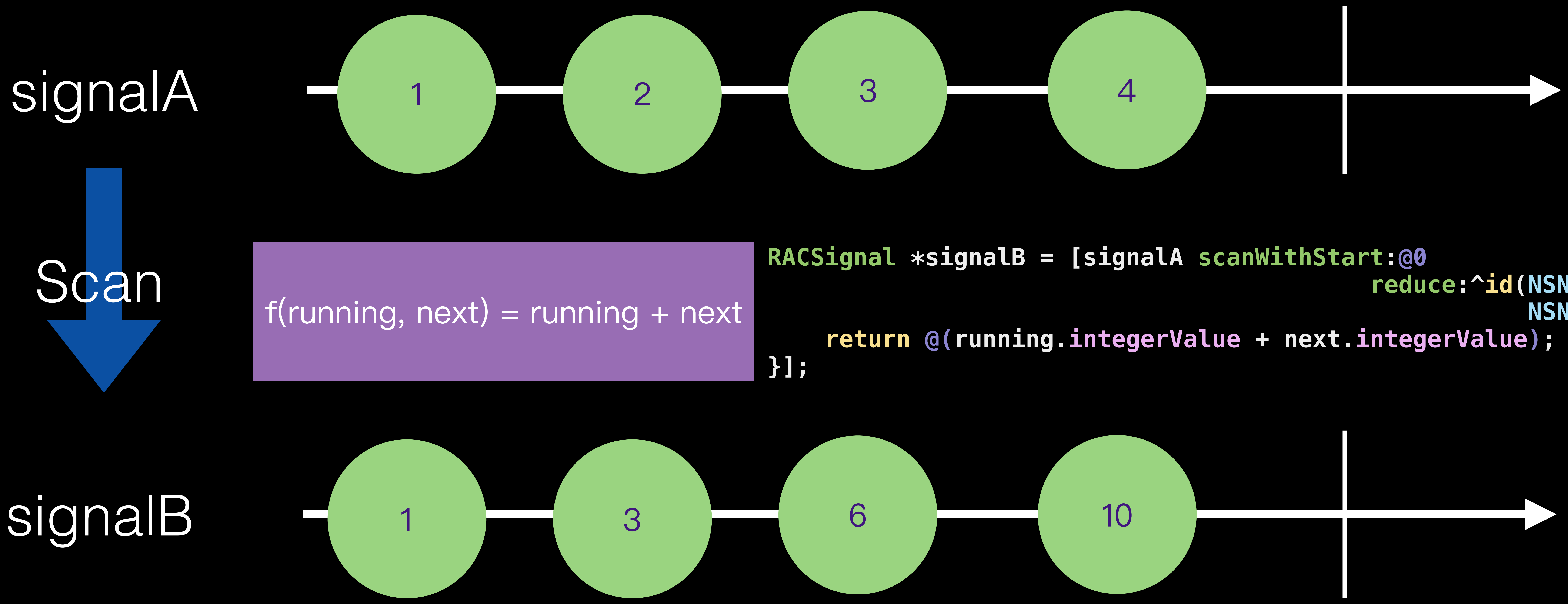
数量操作——Aggregate



```
RACSignal *signalB = [signalA aggregateWithStart:@0
                                          reduce:^id(NSNumber *running,
                                                     NSNumber *next) {
    return @(running.integerValue + next.integerValue);
}];
```

RACSignal各类操作

数量操作——Scan



```
RACSignal *signalB = [signalA scanWithStart:@0
                                          reduce:^id(NSNumber *running,
                                                     NSNumber *next) {
    return @(running.integerValue + next.integerValue);
}];
```