

**Observable ?**



# Observer

observer [-'zɜːrv-]

명사

## 1. 보는 사람, 목격자

According to observers, the plane exploded shortly after take-off.

목격자들에 의하면 그 비행기는 이륙 직후에 폭발했다.

## 2. (회의·수업 등의) 참관인

A team of British officials were sent as observers to the conference.

영국 공무원들 한 팀이 그 총회에 참관인으로 파견되었다.

## 3. 관찰자; 관측자

a royal **observer**

왕실 입회인

# Observer

Observable에 만들어진 이벤트에 반응(react)하는 것의 주체

전달한 데이터를 받고 이에 대한 행동을 취함

||

Observable에서 획득한 이벤트나 값을 수신하는 객체



# Observable

Observable|

어브저버벌



자동완성

관측 가능



# Observable

이벤트를 만들어내는 (emit)의 주체

상태변화가 있을 때마다 메소드를 호출하여 객체가  
직접 Observer에게 여러 이벤트나 값을 보내는 역할

||

Observer에게 알림을 보내는 객체

**Observable**



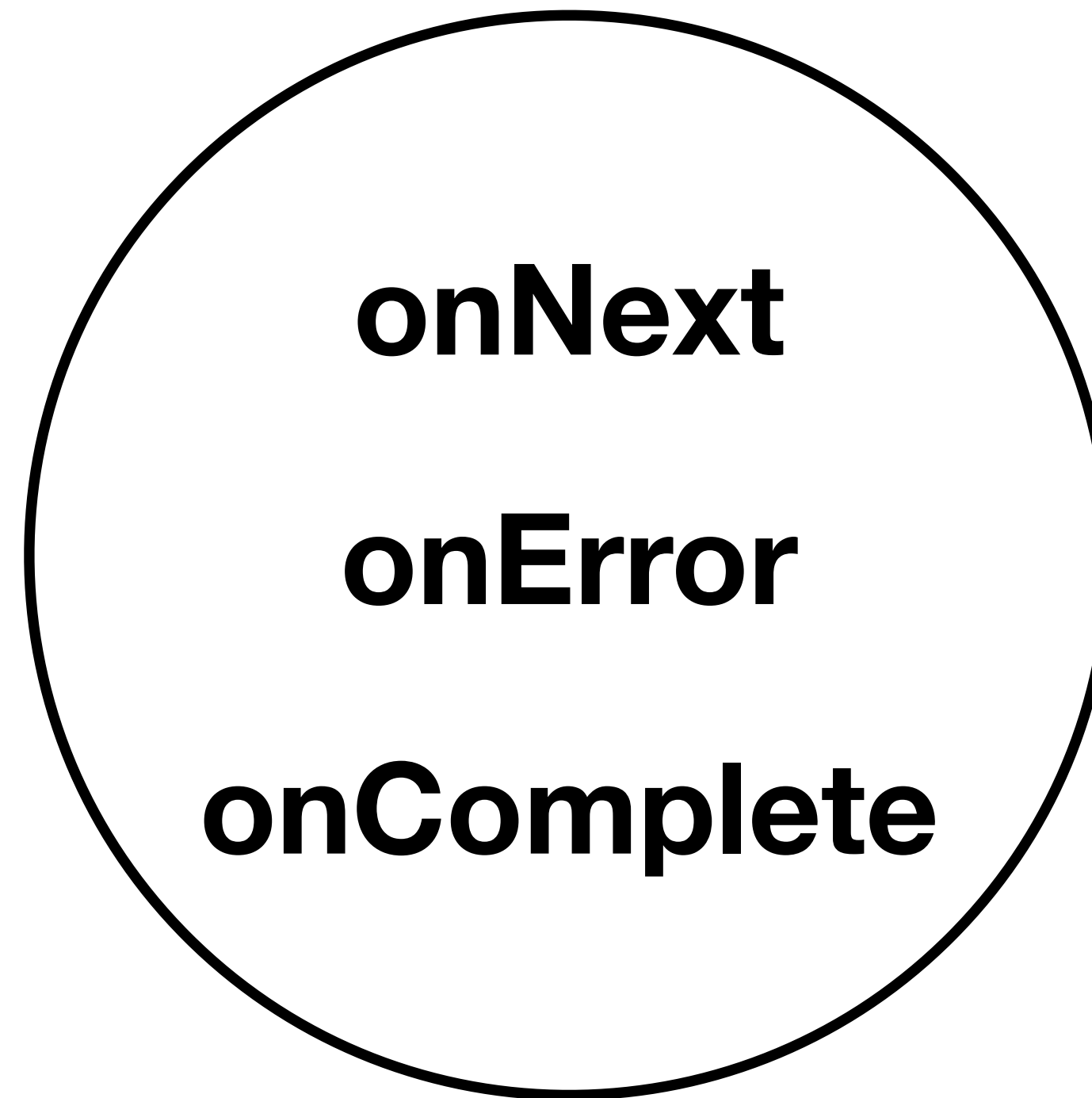
**Observer**

# Observable



# Observer

**Subscribe**



# Observable ↔ Observer

## onNext

Observable은 새로운 항목들을 배출할 때마다 이 메소드를 호출

이 메소드는 Observable이 배출하는 항목을 파라미터로 전달



# Observable ↔ Observer

## onError

Observable은 기대하는 데이터가 생성되지 않았거나 다른 이유로 오류가 발생할 경우 오류를 알리기 위해 이 메소드를 호출한다.

이 메소드는 Observable이 배출하는 항목을 파라미터로 전달


# Observable ↔ Observer

## onCompleted

오류가 발생하지 않았다면 Observable은 마지막 onNext를 호출한 후  
이 메소드를 호출

# Observable ↔ Observer

```
def myOnNext    = { 필요한 연산을 처리한다 };  
def myError     = { 실패한 호출에 대응한다 };  
def myComplete  = { 최종 응답 후 정리 작업을 한다 };  
def myObservable = someMethod(itsParameters);  
  
myObservable.subscribe(myOnNext, myError, myComplete);
```

**Observable**  **Observer**

**Q. Subscribe 를 해지하는 법?**

**UnSubscribe**

# Observer 생성

비동기 모델에서의 메소드 호출 흐름



# Observer 생성

비동기 메소드 호출로 결과를 리턴받고 필요한 동작을 처리하는 메소드를 정의  
= 해당 메소드는 Observer의 일부!

Observable로 비동기 호출을 정의

구독을 통해 옵저버를 Observable 객체에 연결시킨다  
(+ 동시에 Observable의 동작을 초기화)

나머지 코드들을 모두 정의하고 메소드 호출로 결과가 리턴될 때마다  
Observer의 메소드는 리턴 값 또는 Observable이 배출하는  
항목들을 사용해서 연산을 시작한다

# Observer 생성

```
// 옵저버의 onNext 핸들러를 정의한다, 하지만 실행하지는 않는다
// (이 예제에서는, 단순히 옵저버에 onNext 핸들러만 구현한다)
def myOnNext = { it -> /* 필요한 연산을 처리한다 */ };

// Observable을 정의하지만, 역시 실행하지는 않는다
def myObservable = someObservable(itsParameters);

// 옵저버가 Observable을 구독한다. 그리고 Observable을 실행한다
myObservable.subscribe(myOnNext);

// 필요한 코드를 구현한다
```

**Q.Observable은 연속된 항목들을 언제 배출할까?**



**Hot Observable**

**Cold Observable**

**Hot Observable**

생성되자마자 항목들을 배출

옵저버가 구독할 때까지  
항목을 배출하지 않음

**Cold Observable**

# Observable 연산자를 활용한 구성

Observable 생성

Observable 항목변환

Observable 필터

Observable 결합

오류처리, 유틸리티, 조건 및 불린 ....