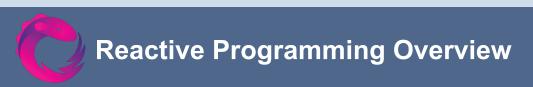
RxSwift Basics – Day 1



Day 1 – Observable, Operator (Filter, Transform, Combine)

- Day 2 Subject (flatMap, flatMapFirst, flatMapLatest)
- Day 3 Two VCs communications with Subject, RxCocoa (Button)
- Day 4 Sequential, Merged Observable Calls
- Day 5 RxCocoa, UI Binding (Button, TextField, Label, TableView)



Reactive Programming Overview (Jafar Hussain from Netflix)

<u>https://www.youtube.com/watch?v=dwP1TNXE6fc</u>



-What are Reactive Extensions (Video)



-Use Marble Diagrams(Video)



Observable

Emits events over time

Observer

Subscribe to listen events emitted by the observable

Observable Sample - 1

```
func observableTestOne() {
    Observable.of(1, 2, 3)
        .subscribe(onNext: { some in
            print(some)
        })
        .disposed(by: self.disposeBag)
}
func observableTestTwo() {
    Observable.from([4,5,6])
        .subscribe(onNext: { some in
            print(some)
        })
        .disposed(by: self.disposeBag)
}
```

Observable Sample - 2

```
func createTestObservable() -> Observable<Int>{
    return Observable<Int>.create { observer in

    return Disposables.create()
  }
}
```

Observable Sample - 3

```
func observableTestThree() {
    Observable<Int>.create { observer in
        observer.onNext(7)
        observer.onNext(8)
        observer.onNext(9)
        observer.onCompleted()
        return Disposables.create()
        }.subscribe(onNext: { some in
            print(some)
        })
        .disposed(by: self.disposeBag)
```

map vs flatMap - 1

map vs flatMap - 2

map vs flatMap - 3

```
func mapVsflatMap() {
    let visitors = ["Tom", "Jerry", "Tweety", "Spike"]
    Observable.from(visitors)
        .map{ "Hello \($0)" }
        .subscribe(onNext: { some in
            print(some)
        })
        .disposed(by: self.disposeBag)
    Observable.from(visitors)
        .flatMap{ Observable.of("Hello \($0)")}
        .subscribe(onNext: { some in
            print(some)
        })
        .disposed(by: self.disposeBag)
```



Create Observables

Pikachu, Ash, Misty, Brock, Charmander, Squirtle

- 1. of
- 2. from
- 3. create



filter

1,2,3,4,5 => less than 3

map

 $1,2,3,4,5 \Rightarrow 10,20,30,40,50$

Lab - 3

```
merge, concat
```

Create two observables and use merge and concat operator

Lab - 4

map vs flatMap

Input 1,2,3 Output 1. [2,3] [4,6] [6,9]

Output 2