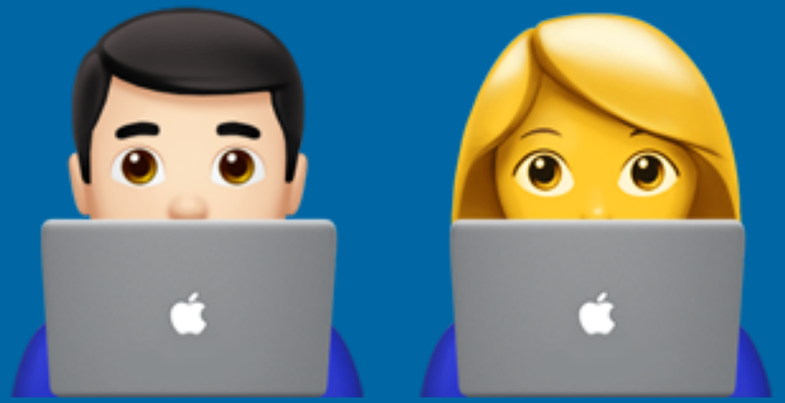
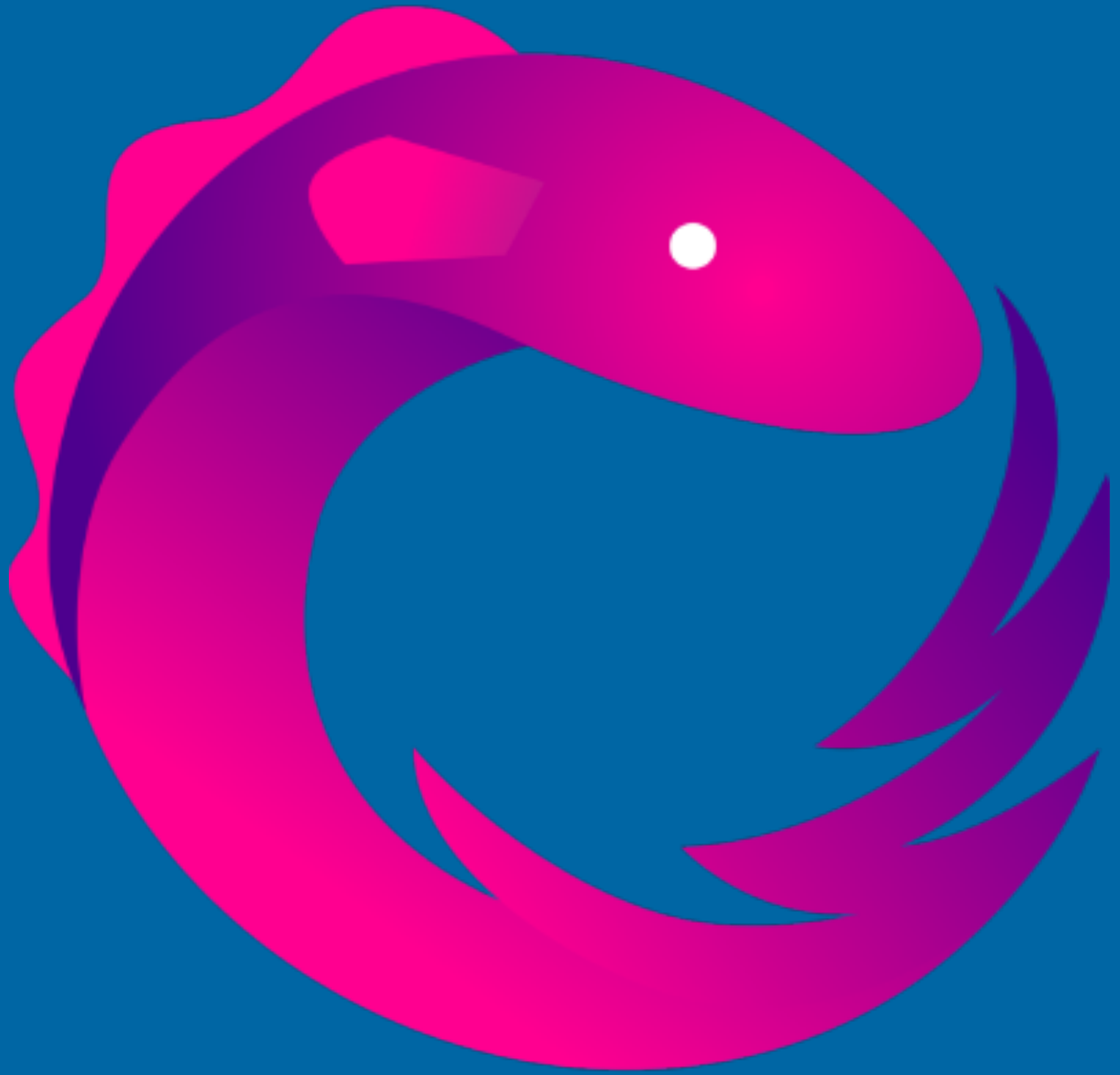
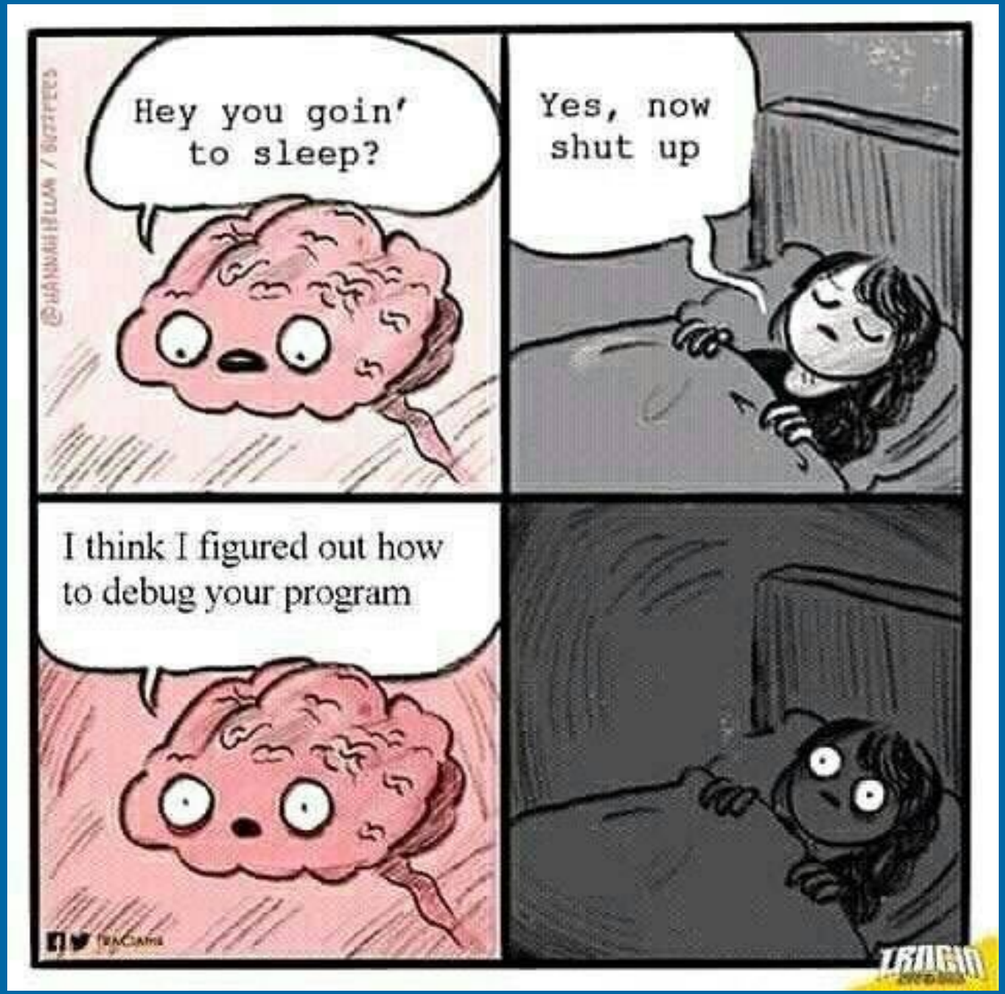
REACTIVE





DISCLAIMER



AGENDA:

» What is Reactive programming?

» Why Reactive programming?

» Rx in iOS (Swift)

» Rx in Android (Java)

» Examples

» Homework

» Q&A

WHAT IS REACTIVE PROGRAMMING?

WHAT IS REACTIVE PROGRAMMING?

» Reactive Programming (aka Rx) is a programming paradigm like Imperative or Functional.

WHAT IS REACTIVE PROGRAMMING?

» Reactive Programming (aka Rx) is a programming paradigm like Imperative or Functional.

» In fact its an extension of functional programming paradigm also called the Functional Reactive Programming(FRP).

WHAT IS REACTIVE PROGRAMMING?

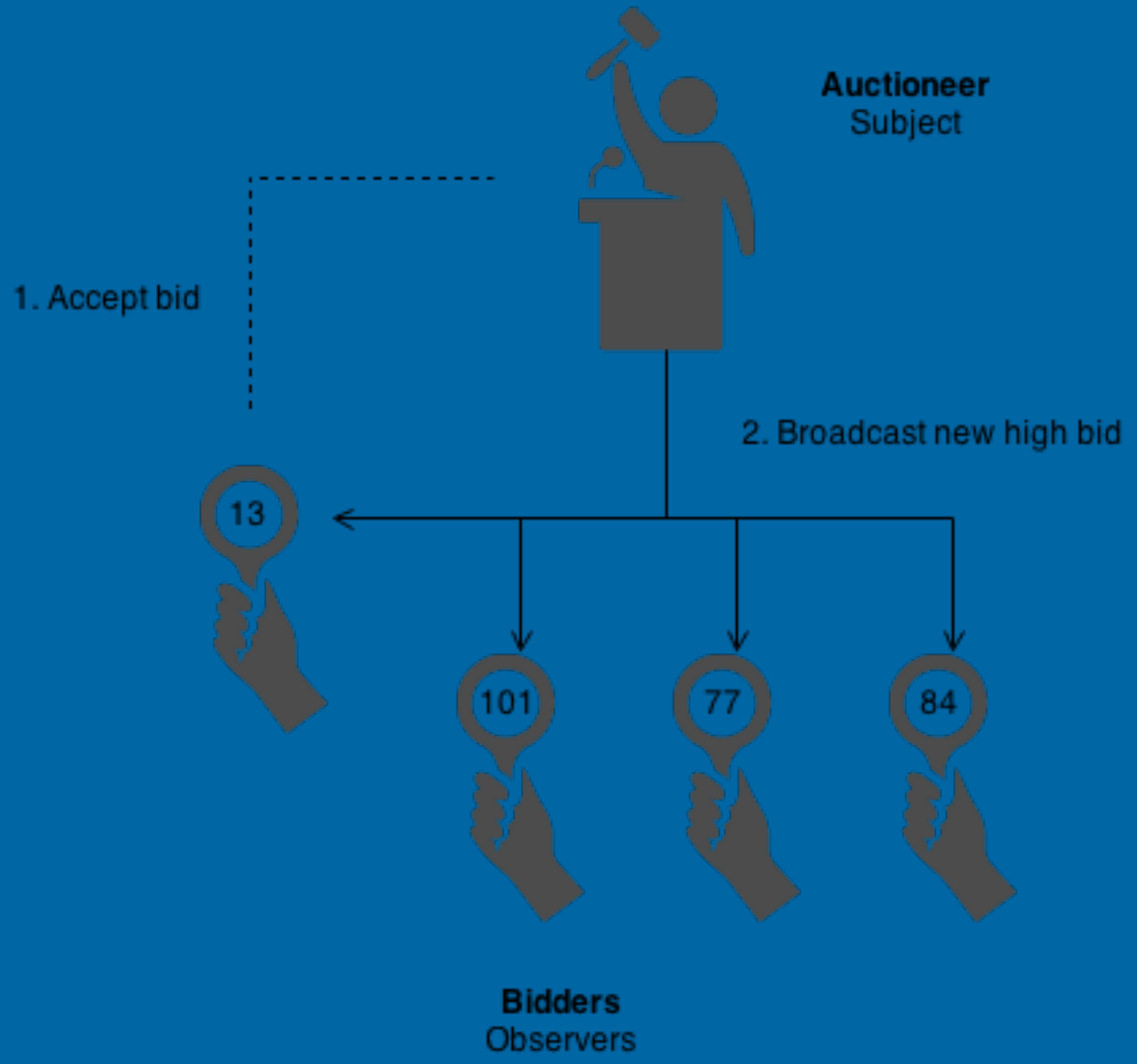
» Reactive Programming (aka Rx) is a programming paradigm like Imperative or Functional.

» In fact its an extension of functional programming paradigm also called the Functional Reactive Programming(FRP).

» The essence of Rx is the Observer pattern.

WAT IS OBSERVER PATTERN? 🤔

The Observer pattern is a software design pattern where in data sources or streams called Observables emit data and one or more Observers who are interested in getting the data subscribe to the observable.



RX = OBSERVABLE + OBSERVER + OPERATORS (+ SCHEDULERS)

OBSERVABLES

» Observables are nothing but the data streams.

» They basically emit the data periodically or only once in their life cycle based on their configurations.

» You can think observers as suppliers. They process and supply the data to other components.

» 😳

» simply: publish signals

OBSERVERS AKA SUBSCRIBERS

» Observers consume the data stream emitted by the observable

» Observers as usual have several callbacks:

» onNext is invoked when new data is emitted by the observable

» onError is invoked if something goes wrong

» 😳

» simply: receive signals

OPERATORS

» Operators allow you to manipulate the data that was emitted

» Operators can also be used to create observables

» the data remains the same but its just converted

» 😳

» simply: manipulate data

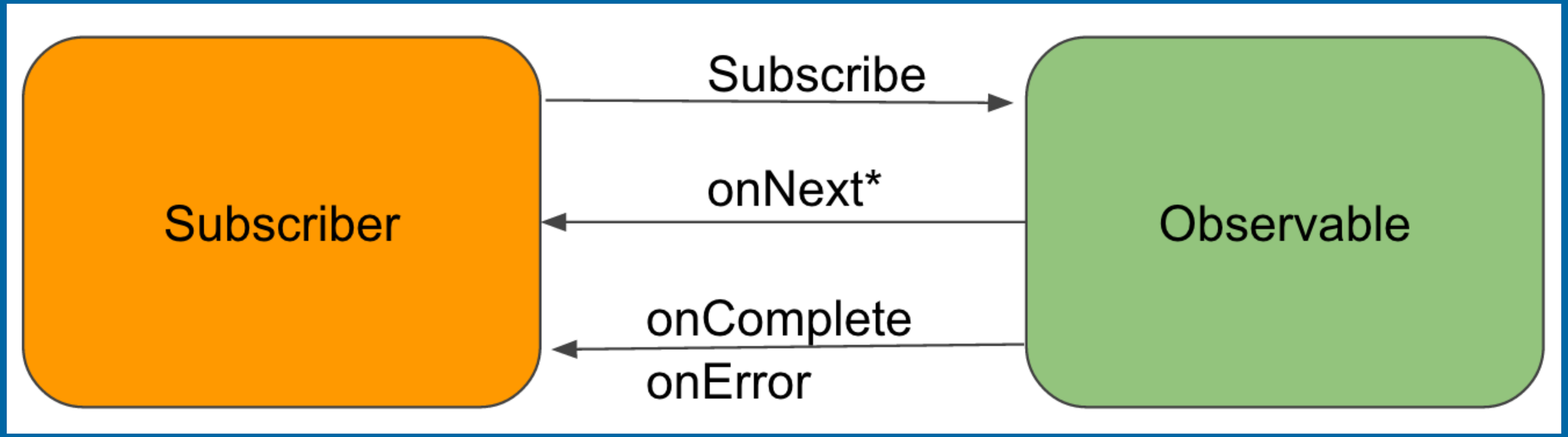
SCHEDULERS

» Schedulers are the component in Rx that tells observable and observers, on which thread they should run.

» can be platform specific (no schedulers in iOS, but tons in Android)

» 😳

» simply: handle multithreading



EXAMPLE

» Observable: newspaper providers like New York Times or DN or The Daily Mirror.

» Observer: end users who consume the news provided by NY Times.

» Operator: you telling/narrating the news to your friend after reading it from the newspaper. The language/word changes, the facts remain the same.

» Scheduler: you get a newspaper online/in kiosk, you read it while working/commuting/eating

WHY REACTIVE?

WHY REACTIVE?

because of lots of Asynchronious Programming ...

WHY REACTIVE?

because of lots of Asynchronious Programming ...
which is?..

WHY REACTIVE?

because of lots of Asynchronious Programming ...
which is?..

... a means of parallel programming in which a unit of work runs separately from the main application thread and notifies the calling thread of its completion, failure or progress.

ASYNC PROGRAMMING EXAMPLES

» Network layer

» Listeners/Delegates on UI controls

» AsyncTask in Android

» ...

ASYNC PROGRAMMING CHALLENGES

» Complicated error processing

» Callback hell

» High risk of receiving untrackable errors

WHEN TO USE RX

» Async tasks = Networking, db, dataset search/ filter

» View event = text input change + when needing delayed events

» Boiler plate code = concat data sets for example

» Bindings ??? Discussion 🤔

» variables ??? Discussion 🤔

“WHEN YOU HAVE A HAMMER EVERYTHING LOOKS LIKE A NAIL.”

Abraham Maslow

WHEN NOT TO USE RX

When NOT to use RxJava

AS MENTIONED BEFORE

```
Reactive Programming =
 Async Programming (Observable & Observer) +
 Functional Programming (Operators)
```

RX: TRY IT OUT!

Play with RX on RxMarbles

RX IN IOS

RX IOS SOLUTIONS:

- ReactiveCocoa: Signal, SignalProducer, Property, Action
- RxSwift: Observable, Observer - all classics, follows ReactiveX
- ReactiveCocoa vs RxSwift

RX IN ANDROID

RXJAVA FTW 🤘

» add dependencies

```
implementation 'io.reactivex.rxjava2:rxandroid:2.0.2'
implementation 'io.reactivex.rxjava2:rxjava:2.1.12'
implementation 'com.jakewharton.rxbinding2:rxbinding:2.1.1'
```

ADD LAMBDAS SUPPORT

More info: Android Studio 3 & Java 8 support

```
android {
    ...
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}
```

LAMBDAS: BEFORE

```
RxTextView.textChanges(searchEditText)
        .filter(new Predicate<CharSequence>() {
            @Override
            public boolean test(CharSequence charSequence) {
                return charSequence.length() > 3;
            }
        })
        .map(new Function<CharSequence, String>() {
            @Override
            public String apply(CharSequence charSequence) {
                return charSequence.toString();
            }
        })
        .subscribe(new Consumer<String>() {
            @Override
            public void accept(String s) {
                search(s);
            }
        });
```

LAMBDAS: AFTER

```
RxTextView.textChanges(searchEditText)
          .filter(charSequence -> charSequence.length() > 3)
          .map(charSequence -> charSequence.toString())
          .subscribe(s -> search(s));
```

DEMO: IOS

» MVVM and data bindings example repo

» Workshop repo

» Old ReactiveCocoa Demo repo

DEMO: ANDROID

» Retrofit & RxJava

» MVP?

» RxBindings

» ...

» Workshop repo - check different branches

HOMEWORK

» Create new project with Rx support

» Play with UI and bindings

» build Network layer based on Rx

» run API call to github/cities etc and present results in the app

Q&A

USEFUL LINKS IOS:

- ReactiveCocoa vs RxSwift
- RxSwift and MVVM
- Eliminating the subscription for an observable in several ways

USEFUL LINKS ANDROID:

- GOTO 2016 Jake Wharton: Exploring RxJava 2 for Android
- When NOT to use RxJava