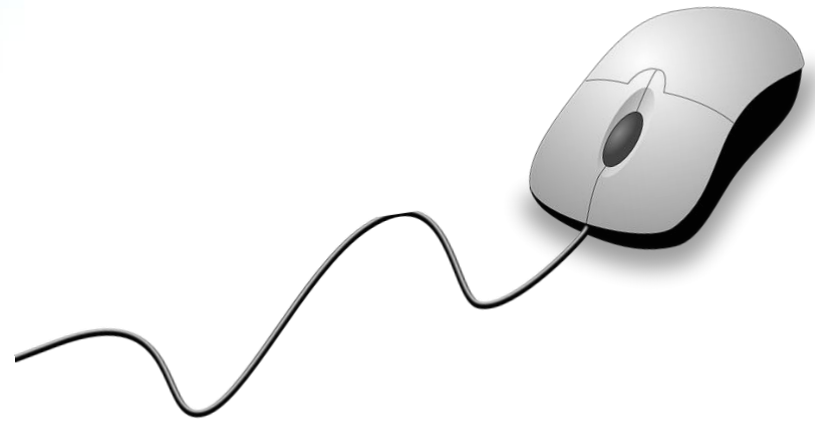


공개SW 솔루션 설치 & 활용 가이드

응용SW -> 콘텐츠배포



React Native

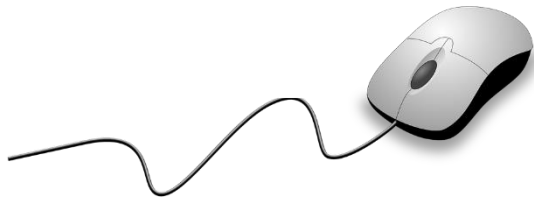
제대로 배워보자

How to Use Open Source Software

Open Source Software Installation & Application Guide



오픈소스 소프트웨어 통합지원센터
Open Source Software Support Center



CONTENTS

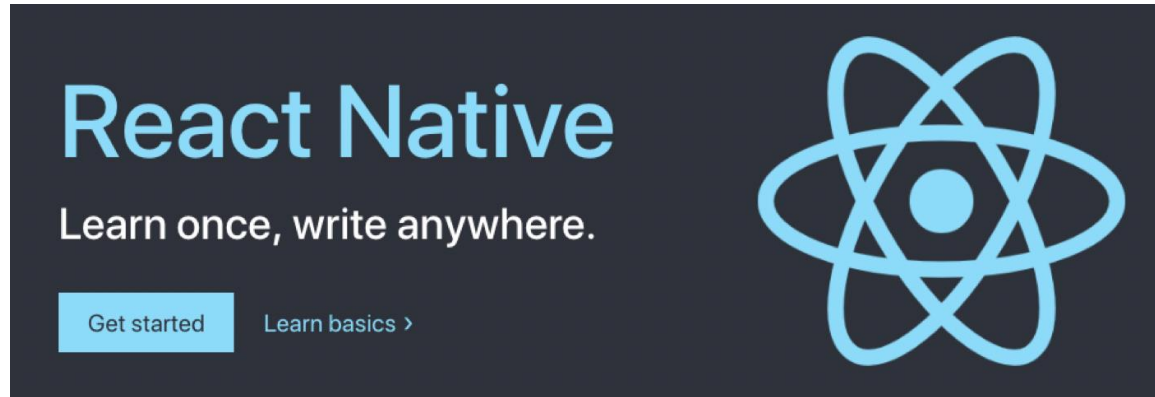
1. 개요
2. 기능 요약
3. 실행 환경
4. 설치 및 실행
5. 기본 개념
6. 동작 원리
7. 용어 정리

1. 개요



소개	<ul style="list-style-type: none">• JavaScript를 사용하여 iOS, Android 그리고 Mac과 Windows 등의 소프트웨어를 만들 수 있게 해주는 Cross Platform Framework• 2015년 3월 26일 출시		
주요기능	<ul style="list-style-type: none">• React의 개념을 기반으로 하나의 코드로 여러 플랫폼을 지원한다.• JavaScript와 JSX 기반으로 개발이 가능하다.• iOS, Android 개발에 주로 사용되며 기존의 하이브리드 방식(Cordova, Ionic, Xamarin등)에 비해 반응 속도가 빠르며 네이티브에서 사용하는 기능을 대부분 지원한다.		
대분류	<ul style="list-style-type: none">• 응용SW	소분류	<ul style="list-style-type: none">• 콘텐츠배포
라이선스형태	<ul style="list-style-type: none">• MIT License	사전설치 솔루션	<ul style="list-style-type: none">• 없음
		버전	<ul style="list-style-type: none">• 0.63.1 (2020년 11월 기준)
특징	<ul style="list-style-type: none">• JavaScript를 사용하여 멀티 플랫폼을 지원할 수 있기 때문에 생산성이 높다.• JavaScript 코드가 백그라운드에서 실시간으로 브릿지(Bridge)를 통해 네이티브 플랫폼과 통신하며 실행된다.• 방대한 기존의 Node.js 기반의 JavaScript 패키지들을 활용할 수 있다.		
개발회사/커뮤니티	<ul style="list-style-type: none">• Facebook 및 외부 기여자들		
공식 홈페이지	<ul style="list-style-type: none">• https://reactnative.dev		





- ✓ 페이스북에서 오픈소스로 공개한 리액트 네이티브는 리액트의 개념을 이용하여 모바일 애플리케이션을 비롯하여 맥, 윈도우 등 다른 프론트엔드 소프트웨어 개발을 할 수 있는 프레임워크이다.
- ✓ 현재 페이스북 및 인스타그램 등 여러 모바일 애플리케이션 제작에 사용되었으며 리액트가 가지고 있는 장점과 기술을 그대로 사용하여 JavaScript로 여러 네이티브 플랫폼을 동시에 개발할 수 있다.



1. 개요



facebook / react-native

Watch ▾

3.7k

★ Unstar

91.4k

Fork

20.2k

JavaScript로 만들어진 프로젝트에서 브릿지를 통해 다양한 네이티브 플랫폼을 지원할 수 있다는것은 리액트 네이티브의 특징이다. 기존 웹 기반의 하이브리드 애플리케이션과 비교하여 성능상의 이점과, 원소스 멀티 플랫폼이라는 두마리의 토끼를 잡을 수 있기 때문에 큰 인기를 누리고 있다.



2. 기능요약



• React Native 주요 기능

React Native로 개발 가능한 플랫폼	<ul style="list-style-type: none">• iOS• Android• Web (2020년 11월 현재 부분적으로 지원한다.)• Mac• Windows
성능	<ul style="list-style-type: none">• 퍼포먼스: Hybrid < React Native <= Flutter < Native(iOS, Android)
개발 편의성	<ul style="list-style-type: none">• JavaScript 최신 문법을 사용하여 개발 가능• 한 개발팀에서 여러 플랫폼을 담당 가능• 기존에 존재하는 Node.js 기반의 다양한 패키지들을 사용 가능
대표 응용 프로그램	<ul style="list-style-type: none">• 페이스북 by facebook• 인스타그램 by facebook• Cake by snow



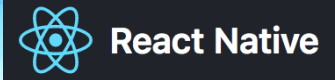


- 윈도우, Linux, Mac 모든 개발 환경 지원
- Node.js 기반의 프로젝트 관리
- 목표 플랫폼 별 빌드 가능 환경
 - ✓ iOS: Mac(XCode)
 - ✓ Android: Windows, Linux, Mac (Android Studio)
 - ✓ Web: Windows, Linux, Mac
- 초기 프로젝트 개발 환경
 - ✓ Expo CLI
Expo CLI를 통해 환경을 구축 후 개발 가능하며 개발 및 배포에 필요한 다양한 기능들이 이미 포함되어 있다. 자체 클라우드 서버를 통해 원격으로 iOS, Android에 개발중인 프로젝트를 구동할 수 있다. 다만 내부적으로 구현되어 있지 않은 네이티브 기능을 추가하려면 `expo eject` 명령어를 통해 Expo를 걷어내야만 한다.
 - ✓ React Native CLI
기본적인 React Native 개발 환경으로 시작한다.



4. 설치 및 실행

세부 목차



4.1 설치 및 실행(공통)

4.2 설치 및 실행(Expo CLI)

4.3 설치 및 실행(React Native CLI)



4. 설치 및 실행



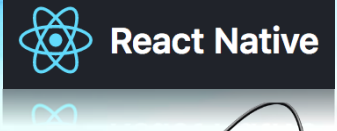
4.1 설치 및 실행(공통)(1/4)

기본 환경 구성

- 필요한 기본 환경
 - ✓ 개발환경: Node 12 이상의 LTS 버전 추천
 - ✓ 패키지 매니저: [Yarn](#) 혹은 npm(Node.js 기본)
 - ✓ Editor: VSCode 혹은 다른 에디터
 - ✓ Android 빌드: Android Studio 최신 버전
 - ✓ iOS 빌드: XCode 최신 버전
- 초기 프로젝트 설치 환경
 - ✓ Expo 환경
 - ✓ Android Studio, XCode가 없어도 실 기기에 개발버전을 배포할 수 있습니다.
 - ✓ React Native 기본 환경
 - 안드로이드 빌드 및 배포를 위해 Android Studio가 필요합니다.
 - iOS 빌드 및 배포를 위해 XCode가 필요합니다.



4. 설치 및 실행



4.1 설치 및 실행(공통)(1/4)

에뮬레이터 설치

- Android: <https://developer.android.com/studio/run/emulator> 참고
- iOS: XCode 설치시 기본 시뮬레이터 자동 설치



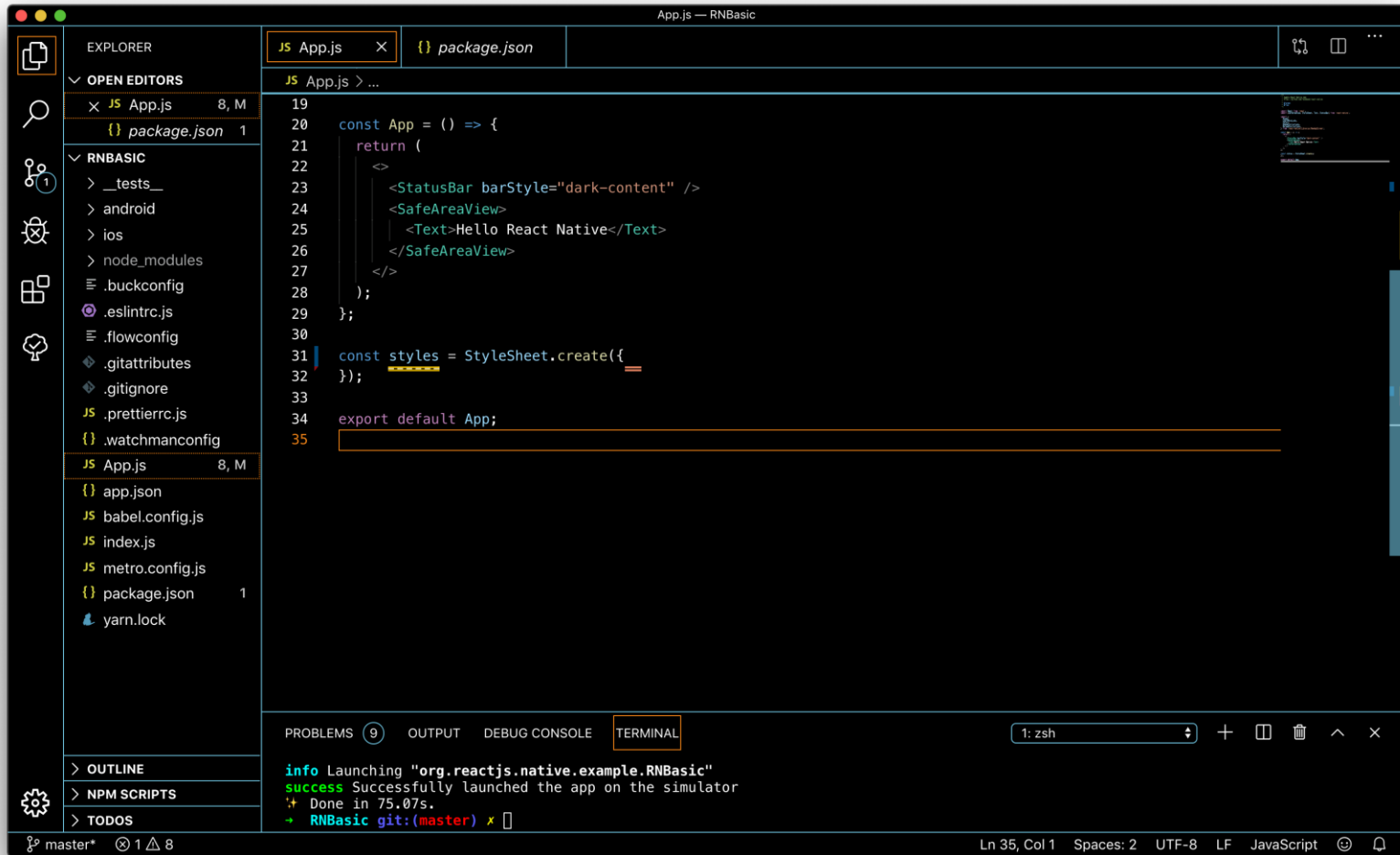
4. 설치 및 실행



4.1 설치 및 실행(공통)(1/4)

VSCode

VSCode 및 Extention 설치 (다른 Editor도 가능)



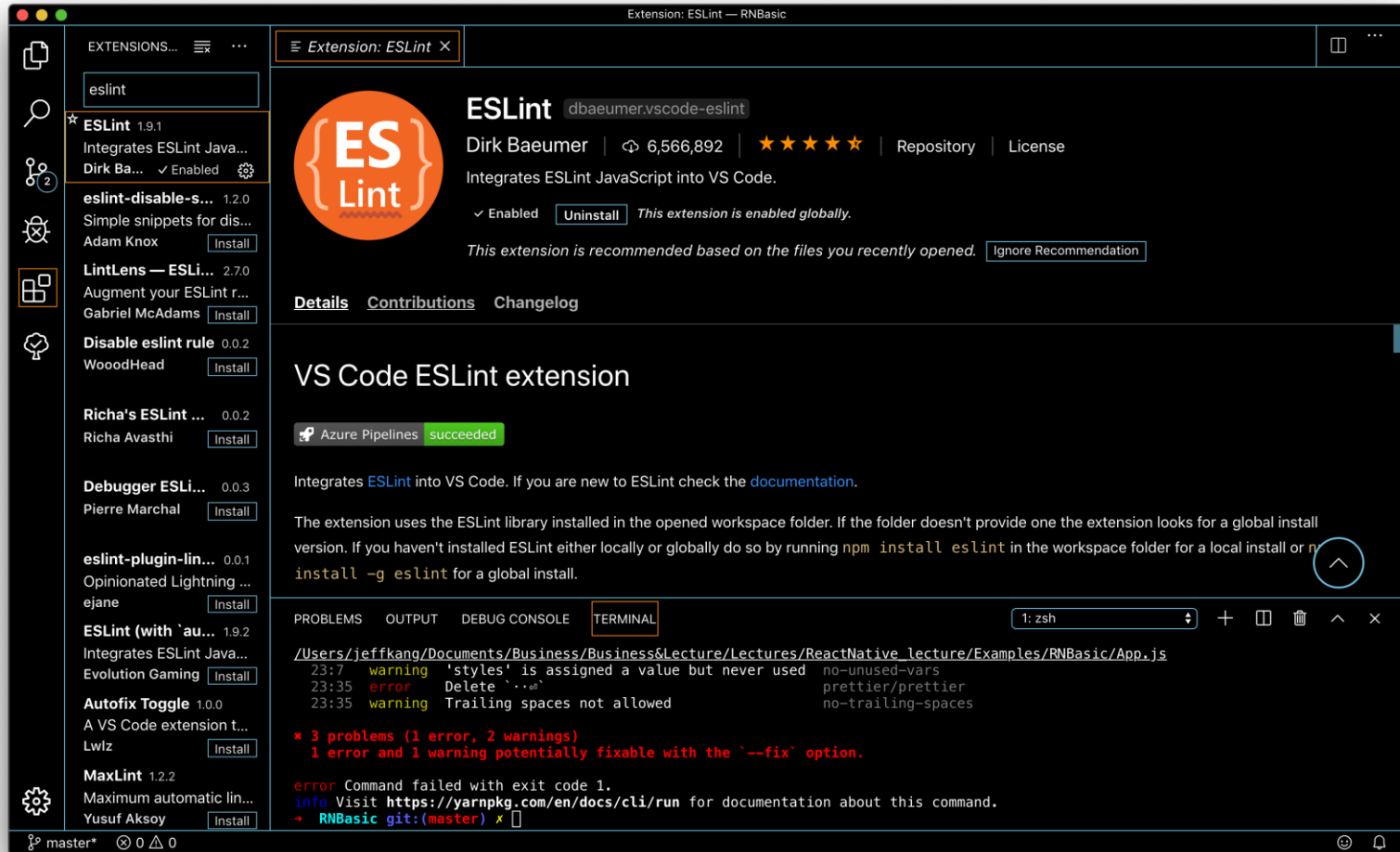
4. 설치 및 실행



4.1 설치 및 실행(공통)(1/4)

VSCode

VSCode 추천 Extension: ESLint, Prettier, React Natie Tools



4. 설치 및 실행

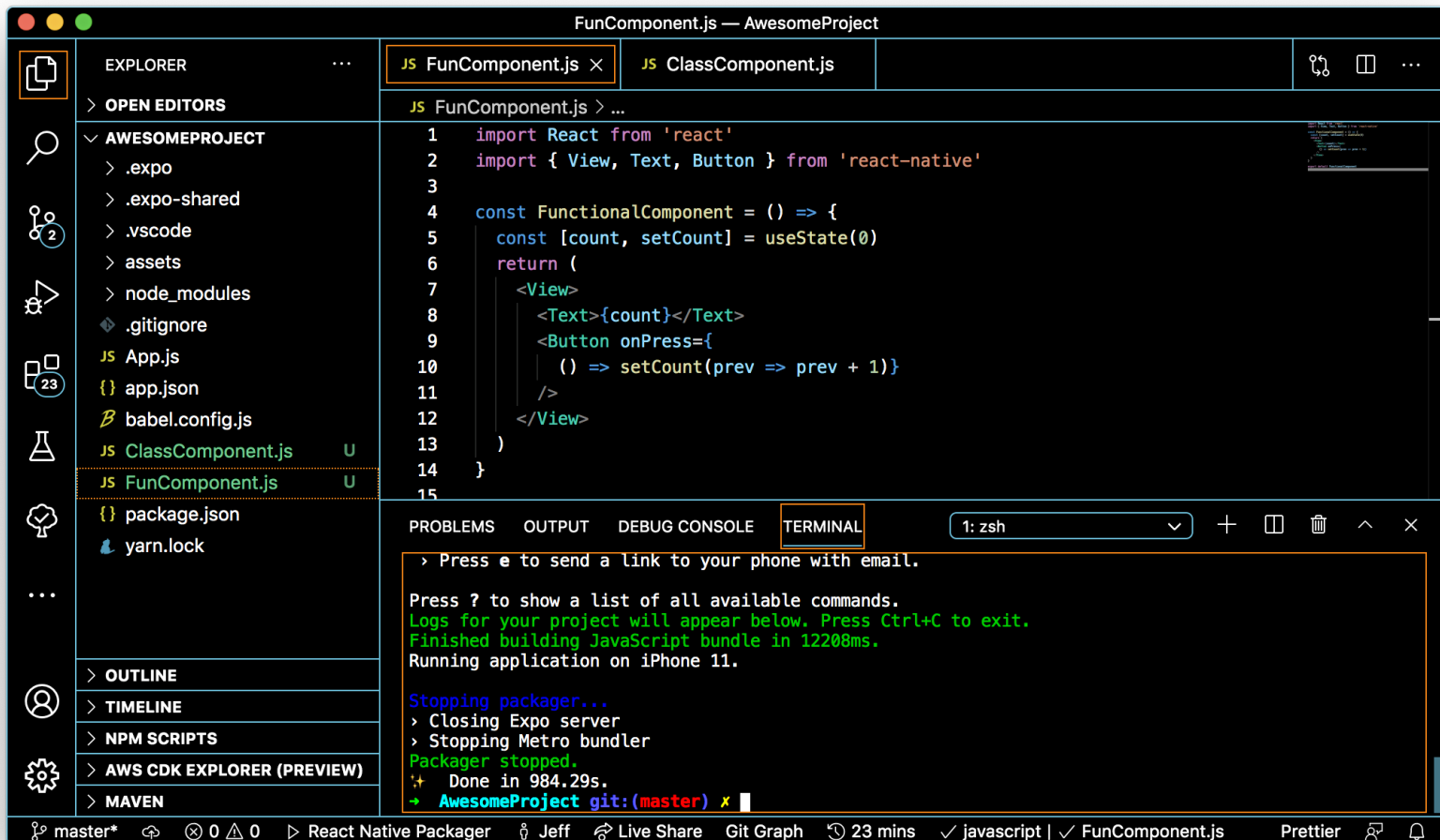


4.1 설치 및 실행(공통)(1/4)

VSCoDe

VSCoDe 내장 Terminal 사용

주로 내장되어 있는 Terminal을 사용하여 간단한 명령들을 수행한다.



4. 설치 및 실행



4.1 설치 및 실행(Expo CLI)(1/4)

1. Expo CLI를 설치, 이미 설치되어 있는 경우 생략

```
yarn add
```

2. expo init을 통한 프로젝트 생성 및 환경 시작

```
expo init AwesomeProject
```

3. 초기 템플릿 설정

본 예제에서는 blank를 선택한다.

```
? Choose a template: > - Use arrow-keys. Return to submit.
----- Managed workflow -----
> blank a minimal app as clean as an empty canvas
blank (TypeScript) same as blank but with TypeScript configuration
tabs (TypeScript) several example screens and tabs using react-navigation and TypeScript
----- Bare workflow -----
minimal bare and minimal, just the essentials to get you started
minimal (TypeScript) same as minimal but with TypeScript configuration
```



4. 설치 및 실행

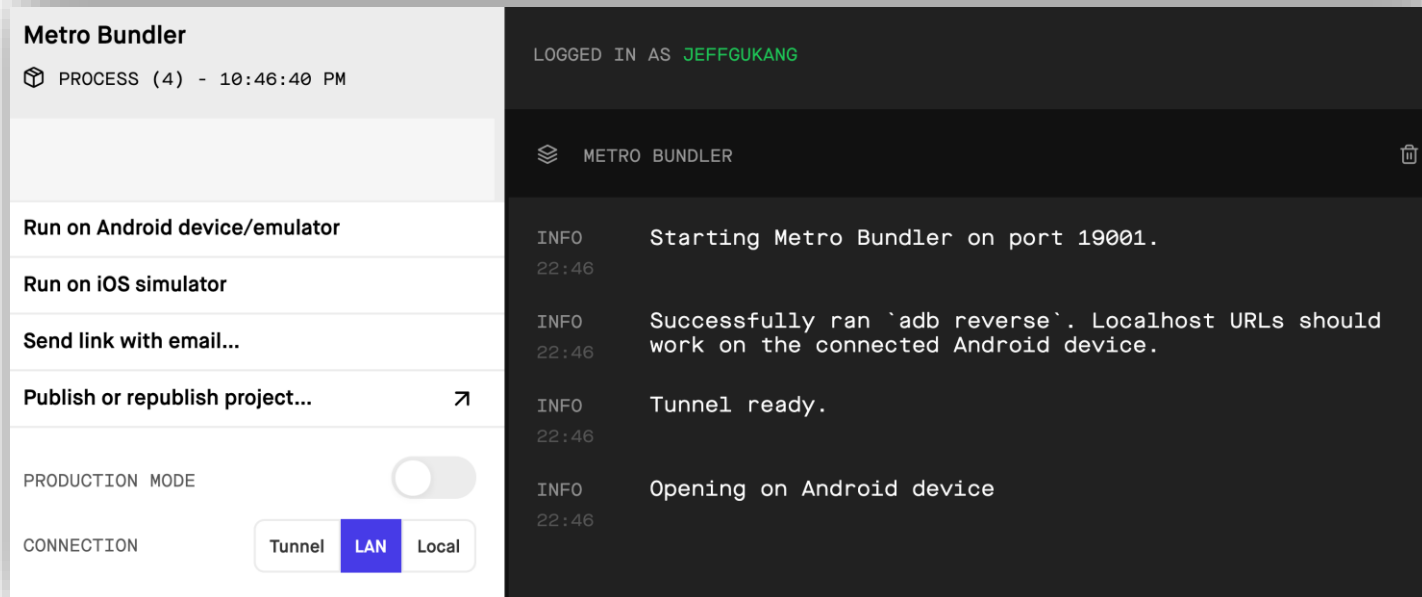


4.1 설치 및 실행(Expo CLI)(2/4)

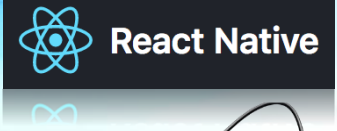
4. 프로젝트 실행

Expo 계정이 없을 경우 계정을 생성해야 진행할 수 있다.

```
cd AwesomeProject  
yarn start # you can also use: expo start
```



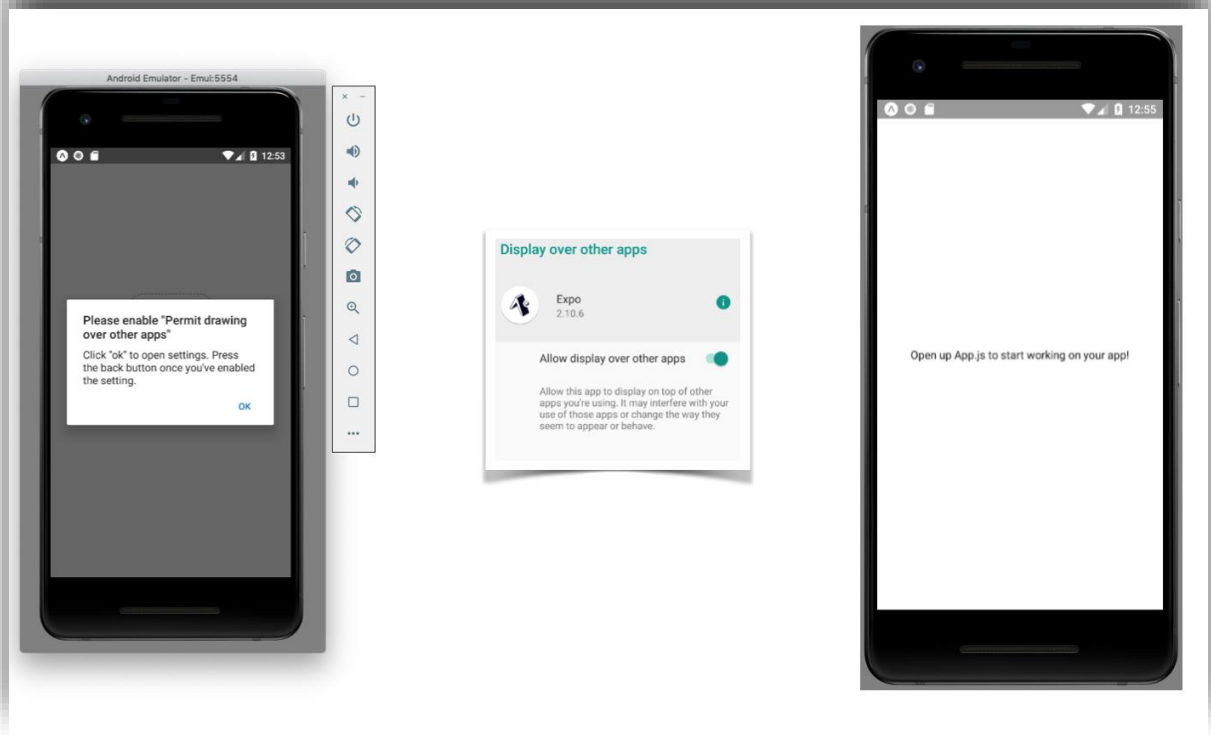
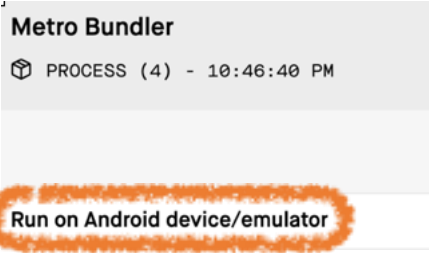
4. 설치 및 실행



4.1 설치 및 실행(Expo CLI)(4/4)

5. 안드로이드 에뮬레이터로 프로젝트 실행

Run on Android device/emulator 클릭



4. 설치 및 실행



4.3 설치 및 실행(React Native CLI)(1/4)

1. 기본 환경 구축

<https://reactnative.dev/docs/environment-setup>

2. 프로젝트 생성

```
npx react-native init AwesomeProject
```

- ✓ Downloading template
- ✓ Copying template
- ✓ Processing template
- ✓ Installing CocoaPods dependencies (this may take a few minutes)

Run instructions for iOS:

- `cd "/Users/jeffkang/Downloads/AwesomeProject2" && npx react-native run-ios`
- or -
- Open `AwesomeProject2/ios/AwesomeProject2.xcworkspace` in Xcode or run `"xed -b ios"`
- Hit the Run button

Run instructions for Android:

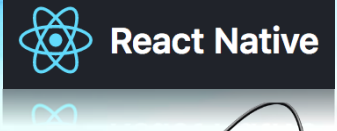
- Have an Android emulator running (quickest way to get started), or a device connected.
- `cd "/Users/jeffkang/Downloads/AwesomeProject2" && npx react-native run-android`

Run instructions for Windows and macOS:

- See <https://aka.ms/ReactNative> for the latest up-to-date instructions.



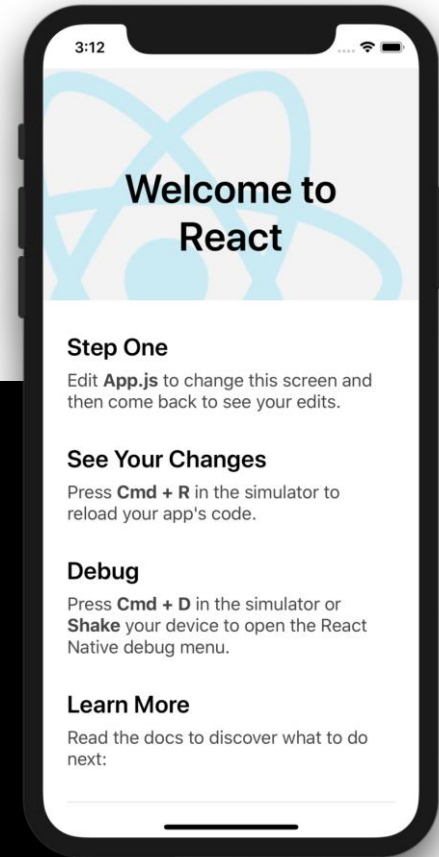
4. 설치 및 실행



4.3 설치 및 실행(React Native CLI)(2/4)

3. 프로젝트 실행

```
cd AwesomeProject
yarn ios
yarn android
```



iPhone X — 12.4

- ✓ Downloading template
- ✓ Copying template
- ✓ Processing template
- ✓ Installing dependencies
- ✓ Installing CocoaPods dependencies (this may take a few minutes)

Run instructions for iOS:

- `cd ReactNativeBasic && react-native run-ios`
- or –
- Open `ReactNativeBasic/ios/ReactNativeBasic.xcworkspace` in Xcode or run `"xed -b ios"`
- Hit the Run button

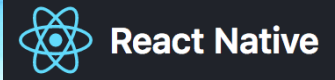
Run instructions for Android:

- Have an Android emulator running (quickest way to get started), or a device connected.
- `cd ReactNativeBasic && react-native run-android`



5. 기본 개념

세부 목차



5.1 컴포넌트

5.2 Props (Properties)

5.3 State

5.4 프로젝트 환경



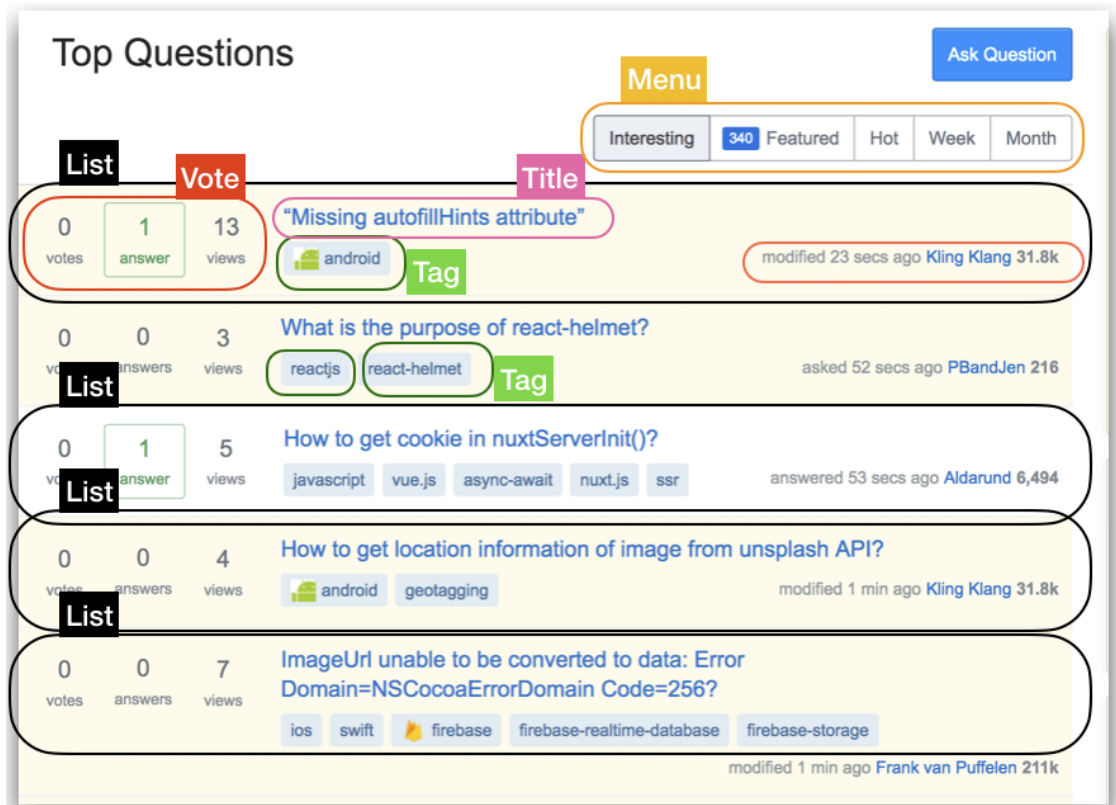
5. 기본 개념



5.1 컴포넌트

컴포넌트란?

- 재사용 가능한 독립적인 UI 구성 단위
- React.js에서 상속
- 단방향 데이터 흐름을 권장



출처: Stackoverflow



5. 기본 개념



5.1 컴포넌트

React Native의 기본적인 컴포넌트

Basic Components

Most apps will end up using one of these basic components. You'll want to get yourself familiarized with all of these if you're new to React Native.

View

The most fundamental component for building a UI.

Text

A component for displaying text.

Image

A component for displaying images.

TextInput

A component for inputting text into the app via a keyboard.

ScrollView

Provides a scrolling container that can host multiple components and views.

StyleSheet

Provides an abstraction layer similar to CSS stylesheets.



5. 기본 개념



5.1 컴포넌트

컴포넌트를 만드는 방법

- Class Component
 - 과거엔 Class Component로 개발
 - Lifecycle
 - State
 - Props

```
// Class Component
class App extends React.Component {
  componentDidMount = () => {

  };

  render() {
    return <RootStack />;
  }
}
```



5. 기본 개념



5.1 컴포넌트

컴포넌트를 만드는 방법

- Functional Component
 - 현재 권장되고 있는 컴포넌트
 - Props
 - Hooks: State, LifeCycle 등을 대체 가능

```
// Functional Component  
App = () => {  
  return <RootStack />  
}
```



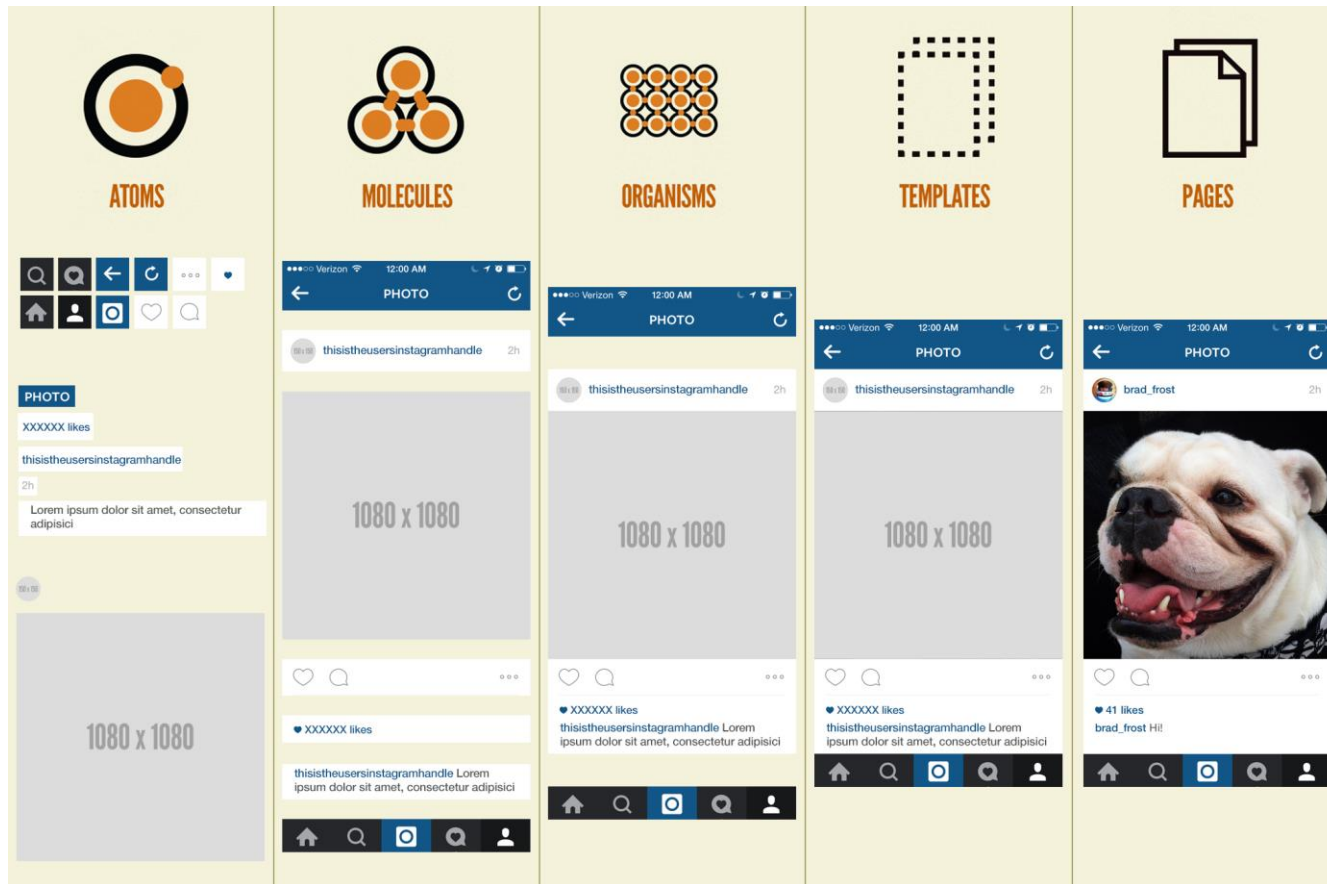
5. 기본 개념



5.1 컴포넌트

컴포넌트 예제

- 컴포넌트들의 조합으로 화면을 개발하게 된다.



출처: <https://atomicdesign.bradfrost.com/chapter-2/>



5. 기본 개념



5.2 Props (Properties)

Props란?

- 컴포넌트에서 사용 할 데이터 중 내부적으로 수정하지 않는 값
- parent 컴포넌트에서 child 컴포넌트로 데이터를 전할 때 주로 사용
- 상위 컴포넌트에서 하위 컴포넌트로 단방향 데이터 흐름
- 변경이 감지될 때 render가 수행되어 연관된 컴포넌트를 다시 그리게 된다.

```
const FunctionalComponent = (props) => {  
  return (  
    <View>  
      <Text>{props.title}</Text>  
    </View>  
  )  
}
```

```
class ClassComponent extends Component {  
  render() {  
    return (  
      <View>  
        <Text>{this.props.title}</Text>  
      </View>  
    )  
  }  
}
```



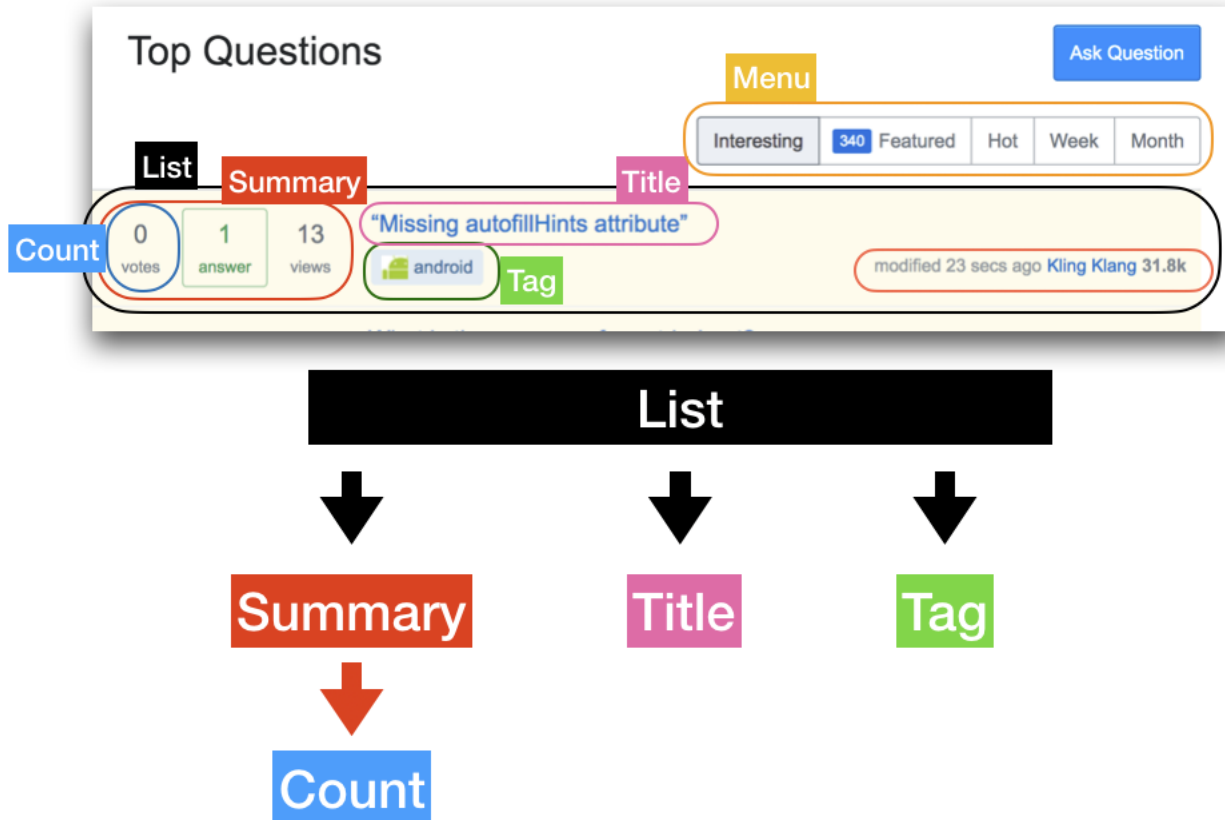
5. 기본 개념



5.2 Props (Properties)

Props의 데이터 흐름

- 부모 컴포넌트에서 자식 컴포넌트로 데이터 전달





State란?

- UI에 영향을 끼칠 수 있는 내부 데이터
- 컴포넌트에서 setState를 통해 내부적으로 재할당 가능
- 변경이 감지될 때 render가 수행되어 연관된 컴포넌트를 다시 그리게 된다
- 현재 컴포넌트의 화면과 관련된 값들을 주로 사용

```
const FunctionalComponent = () => {  
  const [count, setCount] = useState(0)  
  
  return (  
    <View>  
      <Text>{count}</Text>  
  
      <Button onPress={  
        () => setCount(prev => prev + 1)}  
      />  
    </View>  
  )  
}
```



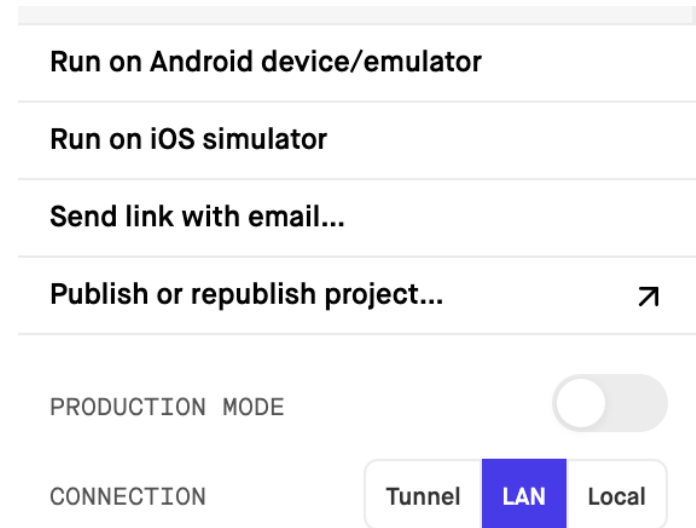
5. 기본 개념



5.4 프로젝트 환경

Expo 환경

- Expo 환경에서의 CONNECTION 옵션
 - ✓ Tunnel
Expo의 클라우드 서버 사용,
 - 느리지만 원격에서도 실행 가능
 - ✓ Lan
WiFi 등 사용
 - 빠름, 같은 망에 속해 있어야 함
 - ✓ Localhost
시뮬레이터 등에서 로컬 연결 사용
 - 제일 빠름, 무선 연결이 되지 않음



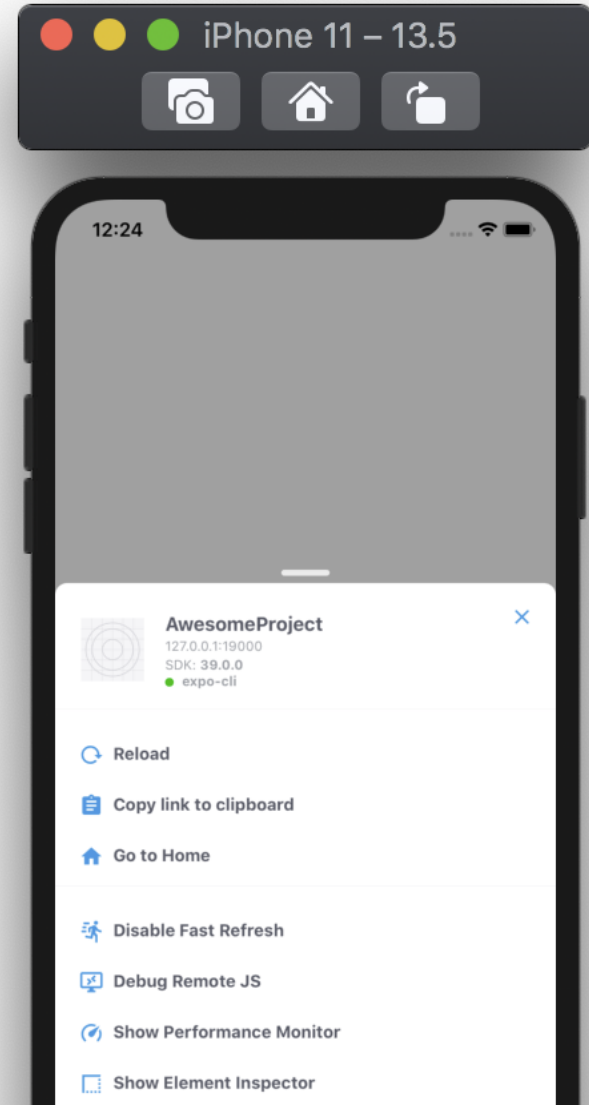
5. 기본 개념



5.4 프로젝트 환경

개발 모드

- 개발 모드에서 메뉴를 띄우는 법
 - 기기를 흔든다.
 - Android: `cmd + m` 혹은 `ctrl + m`
 - iOS: `cmd + ctrl + z`
- 메뉴
 - Reload: 코드 전체 새로 로딩
 - Fast Refresh: 변경된 코드를 바로 반영
 - Debug Remote JS: 디버깅 모드
 - Show Performance Monitor: 퍼포먼스 상태
 - Show Element Inspector: Element별 스펙 확인



5. 기본 개념



5.4 프로젝트 환경

package.json

- Node.js 환경의 프로젝트에서 기본적인 설정 파일
- 각종 프로젝트 정보와 패키지들의 의존성이 정의

```
{  
  "main": "node_modules/expo/AppEntry.js",  
  "scripts": {  
    "start": "expo start",  
    "android": "expo start --android",  
    "ios": "expo start --ios",  
    "eject": "expo eject"  
  }  
}
```

```
"dependencies": {  
  "expo": "^32.0.0",  
  "react": "16.5.0",  
  "react-native": "https://github.com/expo/react-native/archive/sdk-32.0.0.tar.gz"  
},  
"devDependencies": {  
  "babel-preset-expo": "^5.0.0"  
},  
"private": true  
}
```



5. 기본 개념



5.4 프로젝트 환경

app.json

- Expo 프로젝트에서의 앱의 설정 파일 (React Native CLI엔 포함되지 않음)
- 앱 아이콘, 버전, 첫 화면, 기본 상태바 설정 등의 앱 기본 설정등이 포함

```
{  
  "expo": {  
    "name": "RNBasic",  
    "slug": "react-native-basic",  
    "privacy": "public",  
    "sdkVersion": "32.0.0",  
    "platforms": [  
      "ios",  
      "android"  
    ],  
  },  
}
```

```
"version": "1.0.0",  
  "orientation": "portrait",  
  "icon": "./assets/icon.png",  
  "splash": {  
    "image": "./assets/splash2.jpg",  
    "resizeMode": "contain",  
    "backgroundColor": "#ffffff"  
  },  
  "updates": {  
    "fallbackToCacheTimeout": 0  
  },  
  ...  
}
```



6. 동작 원리



Metro Bundler

- Metro Bundler

- ✓ JavaScript Bundler로서 다양한 설정 옵션을 통해 Entry file을 입력받고, 최종적으로 프로젝트에 생성된 코드들과 패키지들을 합친 하나의 JavaScript 파일을 전달
- ✓ 배포 버전에서는 이 파일이 애플리케이션 내부에 알맞은 포맷으로 들어가게 되며 개발 버전에서는 Metro Bundler를 통해 실시간으로 변경된 파일이 빌드되어 실행

```
To reload the app press "r"  
To open developer menu press "d"
```

```
[Mon Nov 30 2020 01:08:12.438] BUNDLE ./index.js
```

```
[Mon Nov 30 2020 01:08:24.871] LOG Running "AwesomeProject2" with {"rootTag":1,"initialProps":{}}
```



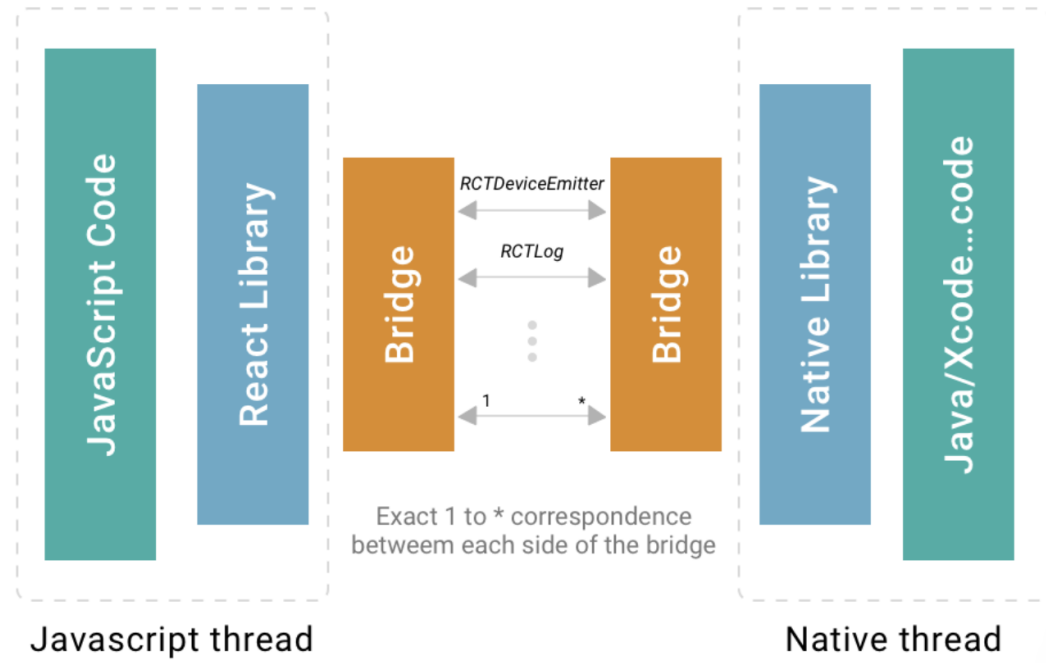
6. 동작 원리



React Native Bridge

- React Native Bridge

- ✓ JavaScript 코드와 Native 인터페이스 중간에서 데이터를 주고 받으며 컨트롤하는 역할
- ✓ 플랫폼에 맞는 Bridge만 있다면 동일 코드로 해당 플랫폼을 지원할 수 있다. 현재 iOS, Android, Windows 10, Mac, Web 등을 지원한다.



출처: <https://www.simform.com/react-native-vs-ionic/>



7. 용어정리



용어	설명
JSX	JavaScript를 확장한 문법으로서 React의 엘리먼트들을 마크업 언어처럼 배열하고 사용할 수 있게 해준다.
Hybrid 애플리케이션 개발 방식	애플리케이션 개발 환경에서 Hybrid 라는 용어는 주로 애플리케이션 내부의 WebView를 활용한 개발 환경을 의미하였다.
CLI	Command Line Interface 의 약자로서 GUI가 아닌 텍스트 터미널을 통해 사용자와 컴퓨터가 상호 작용하는 방식이다. Windows에서는 CMD, Mac과 Linux에서는 Terminal 을 주로 사용한다.
Cross Platform	소프트웨어, 프로그램, 운영 체제 등이 여러 종류의 플랫폼에서 동작이 가능하다는 의미이다. 멀티 플랫폼과 유사한 의미이며 React Native에서는 애플리케이션 결과물이 iOS, Android, Web, Windows, Mac 등 여러 플랫폼에서 동작 가능하다.



Open Source Software Installation & Application Guide



이 저작물은 크리에이티브 커먼즈 [저작자표시-비영리-동일조건 변경허락 2.0대한민국 라이선스]에 따라 이용하실 수 있습니다.