

# React Native 프로젝트 시작부터 공유 컴포넌트 만들기까지

박태성  
**@geoseong** by Slack

# 목차

- `$ react-native init`
  - 프로젝트 생성
  - Flexbox Styling
  - Flow check
- 공유 컴포넌트 만들기
- Running Android/iOS Simulator
- Android APK 만들기
- Let's create React-Native-Seoul ourselves

**\$ react-native init**

# react-native

## 프로젝트 생성

### CLI 입력

```
$ react-native init [프로젝트명]
```

```
ex) $ react-native init AnnyeongRN
```

### 참고문서

[Getting Started -> Building Projects with Native Code](#)

```
├── android
│   ├── app
│   │   ├── BUCK
│   │   ├── build.gradle
│   │   ├── proguard-rules.pro
│   │   └── src
│   │       ├── main
│   │       │   ├── AndroidManifest.xml
│   │       │   ├── java
│   │       │   │   └── com
│   │       │   │       └── annyeongrn
│   │       │   │           ├── MainActivity.java
│   │       │   │           └── MainApplication.java
│   │       └── res
│   │           ├── mipmap-hdpi
│   │           │   ├── ic_launcher.png
│   │           │   └── mipmap-mdpi
│   │           │       ├── ic_launcher.png
│   │           │       ├── mipmap-xhdpi
│   │           │       ├── ic_launcher.png
│   │           │       ├── mipmap-xxhdpi
│   │           │       └── ic_launcher.png
│   │           ├── values
│   │           │   ├── strings.xml
│   │           │   └── styles.xml
│   ├── build.gradle
│   ├── gradle
│   │   ├── wrapper
│   │   │   ├── gradle-wrapper.jar
│   │   │   └── gradle-wrapper.properties
│   ├── gradle.properties
│   ├── gradlew
│   ├── gradlew.bat
│   ├── keystores
│   │   ├── BUCK
│   │   └── debug.keystore.properties
│   └── settings.gradle
├── ios
│   ├── AnnyeongRN
│   │   ├── AppDelegate.h
│   │   ├── AppDelegate.m
│   │   ├── Base.lproj
│   │   │   └── LaunchScreen.xib
│   │   ├── Images.xcassets
│   │   │   ├── AppIcon.appiconset
│   │   │   │   └── Contents.json
│   │   │   └── Contents.json
│   │   ├── Info.plist
│   │   └── main.m
│   ├── AnnyeongRN-tvOS
│   │   └── Info.plist
│   ├── AnnyeongRN-tvOSTests
│   │   └── Info.plist
│   ├── AnnyeongRN.xcodeproj
│   │   ├── project.pbxproj
│   │   ├── xcshareddata
│   │   │   └── xcschemes
│   │   │       ├── AnnyeongRN-tvOS.xcscheme
│   │   │       └── AnnyeongRN.xcscheme
│   └── AnnyeongRNTests
│       ├── AnnyeongRNTests.m
│       └── Info.plist
├── App.js
├── app.json
├── index.js
├── node_modules
├── package.json
└── yarn.lock
```

# react-native

## 프로젝트 생성

### 대략적인 실행 원리

- index.js에서 시작
- `AppRegistry.registerComponent`의 두번째 인자 부분의 클래스를 참조
- index.js에서 참조하는 해당 클래스에서 본격적인 코딩이 이루어짐

```
├─ App.js
├─ app.json
├─ index.js
├─ node_modules
├─ package.json
└─ yarn.lock
```

#### <index.js>

```
import { AppRegistry } from 'react-native';
import App from './App';

AppRegistry.registerComponent('AnnyeongRN', () => App);
```

#### <App.js>

```
import React, { Component } from 'react';
import { Text, View } from 'react-native';

export default class App extends Component {
  render() {
    return (<View style={{flex: 1}}>
      <Text style={{textAlign: 'center'}}>
        AnnyeongRN!
      </Text>
    </View>)
  }
}
```

# Styling : flexbox

## flexbox

일반 웹의 css에서는 `display: flex` 로 사용

## flex

- DOM 컨테이너들의 너비를 채워줌 . 숫자로 표현.
- 1 이면 `flex` 속성 할당된 컨테이너 각 비율에 맞게 할당
- 2 이면 컨테이너 각 비율 x2

## justifyContent

- 세로정렬, 양쪽균등정렬과 좌우정렬 가능

## alignItems

- 가로정렬, 양쪽균등정렬과 좌우정렬 가능

## 공식문서

<http://facebook.github.io/react-native/docs/0.55/flexbox>

## flex개념을 잡아주는 게임 사이트

[flexboxfroggy.com](http://flexboxfroggy.com)

## FLEXBOX FROGGY

◀ 단계 1 of 24 ▶

Flexbox Froggy에 오신 것을 환영합니다! Flexbox Froggy는 Froggy와 친구들 돕는 CSS 코드 게임입니다. 오른쪽의 `justify-content` 속성을 용하여 개구리가 수련잎으로 이동할 수 있도록 도와주세요. 이 속성은 다음의 값들을 인자로 받아 요소들을 가로선 상에서 정렬합니다:

- `flex-start`: 요소들을 컨테이너의 왼쪽으로 정렬합니다.
- `flex-end`: 요소들을 컨테이너의 오른쪽으로 정렬합니다.
- `center`: 요소들을 컨테이너의 가운데로 정렬합니다.
- `space-between`: 요소들 사이에 동일한 간격을 둡니다.
- `space-around`: 요소들 주위에 동일한 간격을 둡니다.

예를 들어, `justify-content: flex-end;`는 개구리를 오른쪽으로 이동시킵니다.

```
1 #pond {  
2   display: flex;  
3     
4 }  
5  
6  
7  
8  
9  
10
```

다음

# Flow check

## 참고문서

<https://medium.com/react-native-training/getting-started-with-react-native-and-flow-d40f55746809>  
<https://flow.org/en/docs/types/>

- 프로젝트 테스트에 앞서 코드의 오타나 타입 체크를 잡아줌
- eslint의 .eslintrc와 비슷한 기능을 한다고 함
- 그렇지만 실제 flow 를 돌려보려면 패키지 별도 설치작업이 필요.

## 패키지 설치 & 세팅

1. **Package.json**에 scripts에 "flow": "flow" 를 추가한다
2. **flow-bin**을 설치한다. 버전은 맞춰야하는것인지 모르겠지만 .flowconfig 내용 최하단에 있는 version에 맞춰 설치해 본다

**\$ npm i flow-bin@0.67.0 --save-dev** <-- 근데 이게 안된다.

**\$ npm i flow-bin@0.67.1 --save-dev** <-- 이걸로 된다.

3. **\$ npm run flow**

```
type Props = {
  foo: number,
};

type State = {
  bar: number,
};

class MyComponent extends Component<Props, State> {
  state = {
    bar: 42,
  };
  {
    "name": "AnnyeongRN",
    "version": "0.0.1",
    "private": true,
    "scripts": {
      "start": "node node_mod",
      "test": "jest",
      "flow": "flow"
    },
  },
}
```

Error App.js:31:10

Cannot extend `Component` [1] with `MyComponent` because `string` [2] is incompatible with `number` [3] in property `state.bar`.

```
26 type State = {
[3] 27   bar: number,
28 };
[1] 29 class MyComponent extends Component<Props, State> {
30   state = {
[2] 31     bar: '42',
32   };
33   static defaultProps = {
34     foo: '42',
35   };
36
37   render() {
38     // console.log('MyComponent]props', this.props);
39     // console.log('MyComponent]state', this.state);
40     return (
41       <Text style={styles.welcome}>
42         {this.props.foo} + {this.state.bar}
43       </Text>
44     );
45   }
46 }
47
48 export default class App extends Component<Props> {
49   render() {
```

# 공유 컴포넌트 만들기



# 공유 컴포넌트 만들기

## - 단위 환산기 모바일앱 만들기

- 버튼을 공유 컴포넌트로 만들어서  
재사용할 예정
  - cm단위를 기준으로 다른 단위로 변환시켜  
주는 모바일앱 제작
  - 단위 참고했던 사이트 :
- 실습코드 의 대략적인 형태는 우측과 같음
  - 완성된 코드 :

[http://gofind.co.kr/Yahoo/unit\\_con.htm](http://gofind.co.kr/Yahoo/unit_con.htm)

[https://github.com/react-native-seoul/Meetup/blob/master/2.RN\\_1.com\\_start\\_to\\_make\\_shared\\_component/sample\\_AnmyeongRN/App.js](https://github.com/react-native-seoul/Meetup/blob/master/2.RN_1.com_start_to_make_shared_component/sample_AnmyeongRN/App.js)

```
import React, { Component } from 'react';
import { 컴포넌트 목록 } from 'react-native';
type shareProps = { 변수: 타입, };
class [공유컴포넌트명] extends Component<shareProps> {
  render() {
    return (
      <컴포넌트 style={styles.속성}>{this.props.변수}</컴포넌트>
    );
  }
}
type State = { 변수: 타입, };
export default class App extends Component<null, State> {
  constructor(props: any) {
    super(props);
    this.state = { key: value };
  }
  커스텀핑션 = (param: string) => {
    ...
  }
  render() { return ( <컴포넌트 style={styles.속성}></컴포넌트> ) }
}
const styles = StyleSheet.create({
  wrapper: { flex: 1, },
});
```

# Running Android/iOS Simulator

# RN 시뮬레이터 돌려보기

## 시뮬레이터 준비 - 참고문서

<https://facebook.github.io/react-native/docs/getting-started>

### 공통

1. `$ brew install node`
2. `$ brew install watchman`
3. `$ npm install -g react-native-cli`

### Android

[Android Studio 다운로드](#)

### iOS

[Xcode 다운로드](#)

## 시뮬레이터 돌려보기

### iOS

`$ react-native run-ios`

**\*\* No bundle URL present** 메시지가 뜬다면...

다른 터미널에서 `$ react-native start` 를 먼저 실행 후  
시도해 본다.



# RN 시뮬레이터 돌려보기

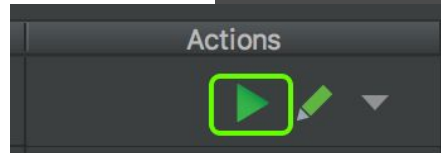
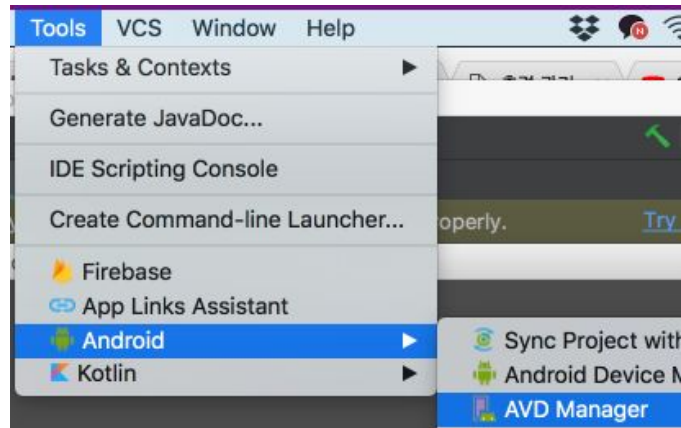
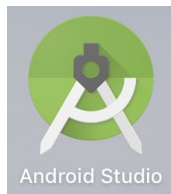
## 시뮬레이터 돌려보기

### Android

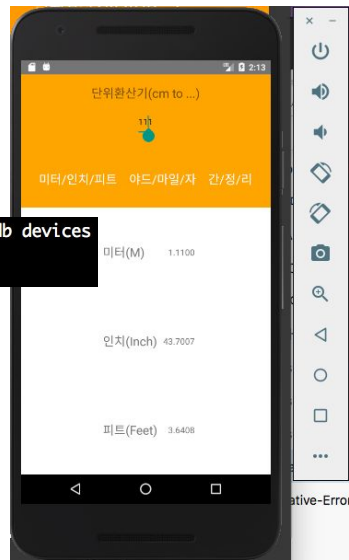
1. Android Studio에서 Emulator를 미리 실행을 해놔야 한다.  
기기연결 확인방법 : `$ adb devices`
2. `$ react-native run-android`

### 처음으로 Android 시뮬레이터 돌리기

1. Android Studio 실행
2. 새 Empty 프로젝트 만들기
3. 상단메뉴 Tools -> Android -> AVD Manager -> Virtual Device 생성 -> [원하는 기기] -> x86 images -> Marshmellow System Image 다운로드
4. 다운로드 후 추가된 기기의 재생버튼 클릭
5. `$ react-native run-android`



```
ideasam-geoseongui-MacBook-Pro:AnnyeongRN geoseong$ adb devices
List of devices attached
emulator-5554    device
```



# Android APK 만들기

# Android APK 만들기

<https://facebook.github.io/react-native/docs/signed-apk-android.html>

## Generating a signing key

[프로젝트폴더]/android/app 경로에서 ..

```
$ keytool -genkey -v -keystore my-release-key.keystore -alias
```

```
my-key-alias -keyalg RSA -keysize 2048 -validity 10000
```

## android/gradle.properties

```
MYAPP_RELEASE_STORE_FILE=my-release-key.keystore
```

```
MYAPP_RELEASE_KEY_ALIAS=my-key-alias
```

```
MYAPP_RELEASE_STORE_PASSWORD=*****
```

```
MYAPP_RELEASE_KEY_PASSWORD=*****
```

## android/app/build.gradle

우측 코드 참고

## Generating the release APK

```
$ cd android && ./gradlew assembleRelease
```

## Testing the release build of your app

```
$ react-native run-android --variant=release
```

## <android/app/build.gradle>

```
...
android {
    ...
    defaultConfig { ... }
    signingConfigs {
        release {
            if (project.hasProperty('MYAPP_RELEASE_STORE_FILE')) {
                storeFile file(MYAPP_RELEASE_STORE_FILE)
                storePassword MYAPP_RELEASE_STORE_PASSWORD
                keyAlias MYAPP_RELEASE_KEY_ALIAS
                keyPassword MYAPP_RELEASE_KEY_PASSWORD
            }
        }
    }
    buildTypes {
        release {
            ...
            signingConfig signingConfigs.release
        }
    }
}
...
```

**Let's create React-Native-Seoul ourselves**

# React Native Seoul 홈페이지

---

커뮤니티

<https://reactnativeseoul.org/>

react-native-seoul

회원가입로그인

전체 카테고리 > 최근글 인기글 카테고리

| 주제  | 카테고리 | 사용자 | 답변 | 조회수 | 활동    |
|---|------|-----|----|-----|-------|
| 푸 리액트네이티브서울에 오신 것을 환영합니다 🙌<br>리액트네이티브서울 밋업에서 운영하는 커뮤니티사이트에 오신 것을 환영합니다. 리액트네이티브 개발 이외에도 앱과 웹 개발 전반에 대해 자유롭게 의견을 나눌 수 있는 공간입니다. 리액트네이티브를 처음 접하시는 분들, 경험이 많으신 분들 모두 환영합니다. 자유롭게 리액트네이티브 및 기타 관련 기술에 대해 논의하고 질문의 응답할 수 있습니다. 리액트네이티브를 사용하면서 세로로 알아낸 것들이 있다면 다른... <a href="#">더 읽기</a> |      |     | 0  | 24  | 7시간   |
| 안녕하세요 궁금한게 있습니다   |      |     | 1  | 6   | 13분 전 |

더 이상 읽을 최신 주제가 없습니다

기술로그

<https://react-native.co.kr/>



React Native Seoul

Resource

React Native Seoul



Last updated 10 minutes ago

리액트네이티브서울에서 운영하는 사이트입니다.

React Native Seoul에서 제공하는 사이트입니다.

Meetup

Forums

Slack

Github

매주 수요일 신논현역 인근에서 밋업을 진행하고 있습니다.

<https://www.meetup.com/ko-KR/react-native-seoul>



# 발표자 모집!!!!!!

---

- 함께 스터디하고 경험을 공유할 분을 모집합니다
- JS 사용자라면 **RN** 초보자라도 의지만 있다면 할 수 있다!
- **React-Native-Seoul** 경험자 버프를 받을 수 있는 기회

Question