

**React Native**

## 리액티 네이티브란?

UI 라이브러리 => 아이오닉, 코르도바와 다르게 네이티브임

- 마지막 컴파일 시, `Java/Object-c` 로 컴파일되어 네이티브에서 구동이 가능하게 됨

## 장점

1. Javascript 를 활용하여 **AOS/IOS** 앱 개발
2. 개발 커뮤니티가 큼
3. 다수의 회사가 실제로 React Native를 활용해 앱 개발 중

## 네이티브 앱 개발과 다른 점

- `.apk` 만들어서 배포하는 방식이 아닌 서버에서 클라이언트가 소스코드를 다운 받는 실시간 방식
- 빠른 오류 수정 가능

## 리액트 네이티브의 기본 컨셉 (공식 문서 참고)

- 더 이상 div, html이 없음 => 우리가 return 할 수 있는 컴포넌트가 정해져 있음
  - 그 이유는 OS 종류에 따라 컴포넌트가 알아서 컴파일 됨
  - 그러다 보니 모든 문법이 `strict` 함

# Expo

- Xcode와 Android studio 없이 Expo를 통해 두 플랫폼 모두 확인 가능
  - 단, 플랫폼 별 디바이스 혹은 시뮬레이터 설치 필요
- 실행 방법 변화로 최근 CLI를 활용하여 설치 및 실행 가능

## 리액트 네이티브 특징

## 1. Flexbox

- CSS의 개념을 차용하여 디바이스 별로 다른 디스플레이 및 해상도 크기에 유기적 대응 가능
- 안드로이드로 이해하자면 `Constraint Layout` 과 같은 개념



## 2. Component

- React Native에서의 `View` 의 기본 단위
- 리엑트는 상속보다 컴포넌트의 조합을 선호 (여러 빌딩 블록의 조합)
- JSX를 활용하여 컴포넌트 레이아웃을 정의 (HTML과 비슷한 형식의 마크업 언어)

## 3. Props와 State

### 3.1. Props (불변)

- 컴포넌트를 동적으로 만드는 메커니즘 제공
- 컴포넌트의 기본적인 작동에 변화를 주기 위해 자식 컴포넌트에 전달
- Props 데이터를 엄격하게 관리하기 위해 PropTypes로 형을 지정함
  - 개발자에게 경고

### 3.2. State (가변)

- Props와 다르게 컴포넌트 내부에 존재
- View의 시각적 효과와 관계가 큼
- 특정 이벤트를 통해 데이터 변화 시, `setState({})` 를 활용해 state 변경
  - React Native의 특성으로 인해 자동 UI update 일어남