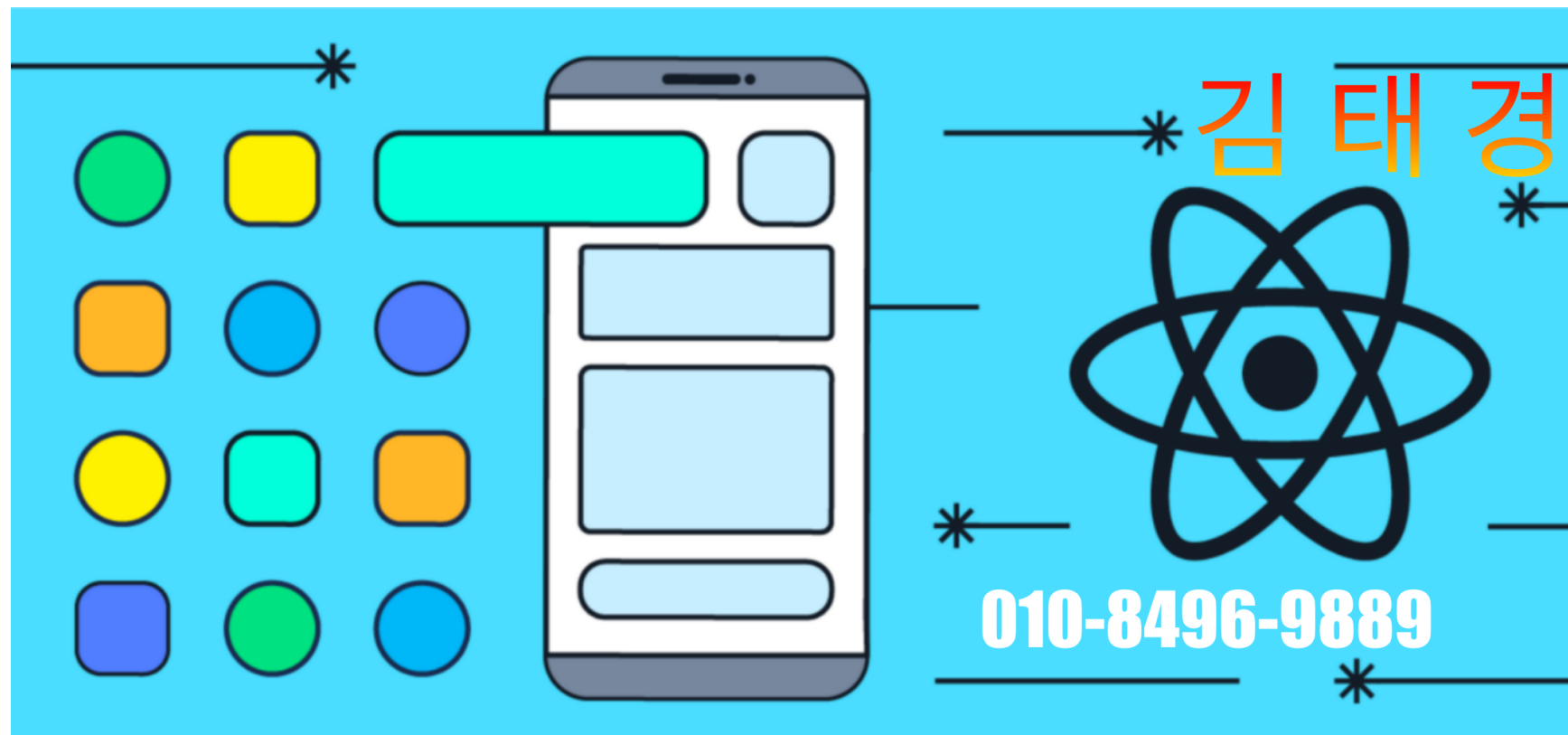


하이퍼레저와 연동하는

리액트 네이티브 앱 개발 강의



Web이나 App 개발 경험이 있나요?

from **Zero** to **App**

4주간 알차고 재밌는 시간

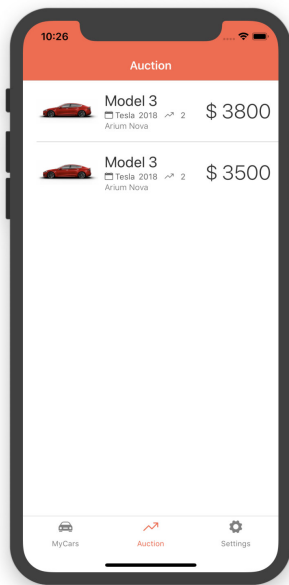
앱 개발 관련 궁금증을 해소하는 시간

이 강의의 목표

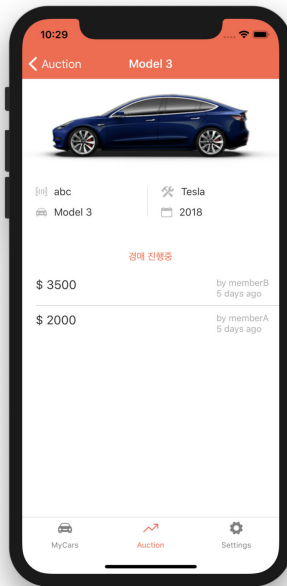
하이퍼레저를 이용하는 모바일 앱을 만든다!

- 자동차 경매
- 모바일 앱 : **react-native** (expo)
- 블록체인 : **hyperledger** (composer / playground / rest-server)

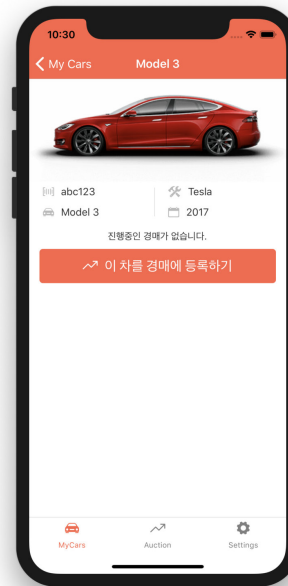
우리가 만들 앱의 모습



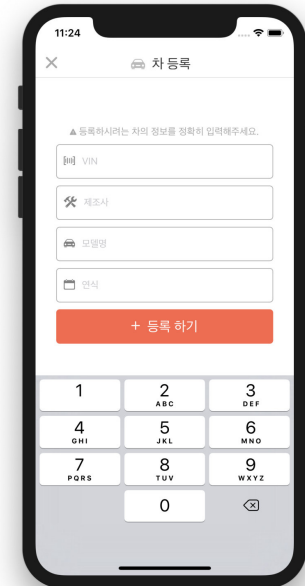
iPhone X - 12.1



iPhone X - 12.1



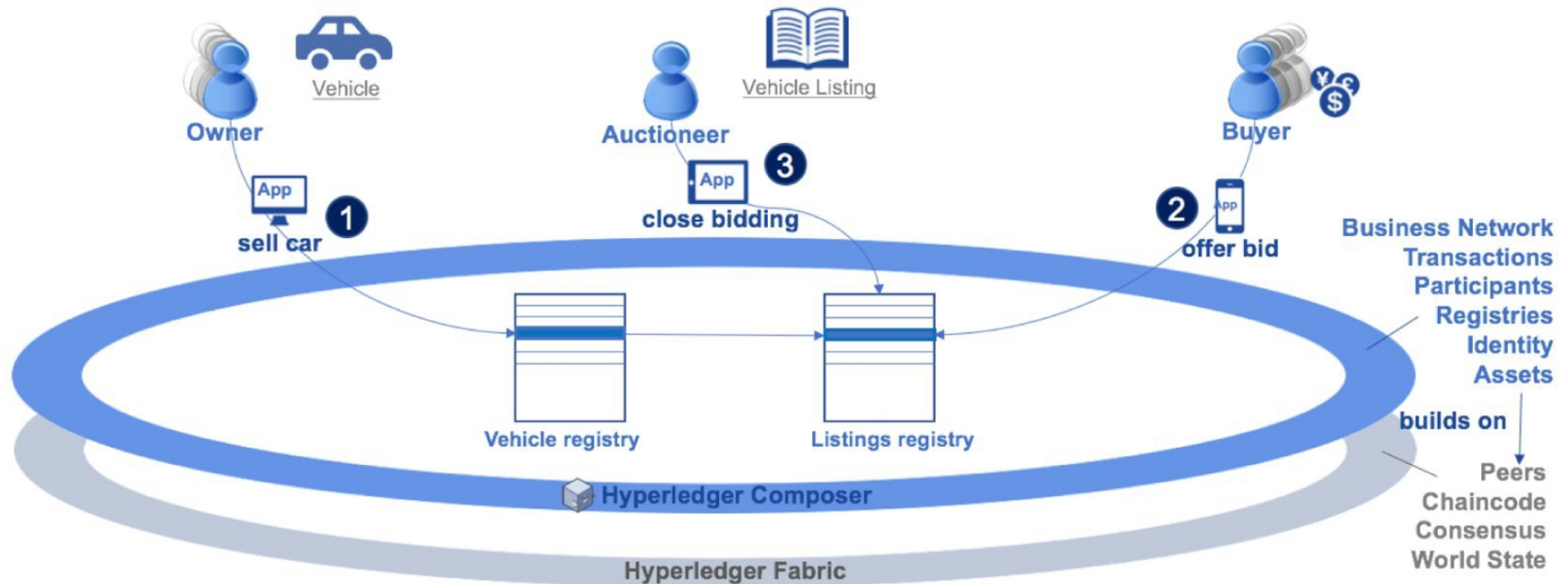
iPhone X - 12.1



iPhone X - 12.1

연동할 블록체인 네트워크

자동차 경매 네트워크



리액트란 무엇인가?

A JavaScript library for building user interfaces

- <https://reactjs.org>
- 선언적
- 컴포넌트
- Learn Once, Write Anywhere

리액트 네이티브란 무엇인가?

iOS와 안드로이드에서 동작하는

네이티브 앱 을 만드는

자바스크립트 프레임워크

RN 장점 1 : 진짜 네이티브 UI

흉내낸 네이티브 UI가 아닌 **진짜** (웹뷰/HTML 아님)

- 대상 플랫폼의 표준 렌더링 API
- 리엑트는 메인 UI 스레드와 분리되어 실행

RN 장점 2 : 뛰어난 개발 경험

투자 시간 대비 결과물 완성도

- 변경사항 빠른 확인
- JS개발자에게 익숙한 개발 방법 및 디버깅 방법
- code-push를 이용한 빠른 소스 업데이트

RN 장점 3 : Learn Once, Use Everywhere

한쪽에서 알게 된 지식과 경험을 다른 곳에 적용할 수 있다

- JavaScript, 리액트 : 웹, 앱, 서버까지..
- 페이스북 애드매니저는 안드로이드 버전의 코드 87%를 iOS와 공유

RN의 단점

장점 때문에 생긴 단점

- 브리지 기술이다보니 한가지 레이어가 더 들어간 것
- 새로운 OS의 기능이 바로 지원되지 않는다?
- 네이티브를 몰라도 되는 것이 아니다

누가 리액트 네이티브를 사용하나요?



진행방식 안내

목표제시 -> 관련 지식 -> 실습

- 자동차 경매 앱에 필요한 컴포넌트를 하나씩 만들어 간다.
- 컴포넌트를 만들어 가면서 필요한 관련 지식을 배운다.
- 참여자 모두 직접 코딩하면서 하나의 완성된 앱을 만든다.

실습 진행 방법

ㅍㅍ을 이뤘서 서로 궁금한거 묻고 답하면서 실습 진행

- 해봐야 궁금한 점이 더 생긴다.
- 개발자 성장과 밀접한 질문과 응답 연습
- 스스로 해결 시도 그 다음 도움 요청해야 이해도가 더 높아진다.

목표 1. 리액트 네이티브 개발 환경을 만든다

Q1-1. Expo 설치/설정?

Q1-2. 리액트 네이티브를 위한 VS Code 설치/설정?

Q1-3. 이 환경에서는 어떻게 개발하게 되나요?

잠깐) 개발 환경을 만들면 뭘 할 수 있는건가요?

- Expo를 이용한 개발 구경하기 (시연)

Q1-1. Expo란 무엇이고 어떻게 설정하나요?



Expo란?

Expo is a set of tools, libraries and services which let you build native iOS and Android apps by writing JavaScript.

JavaScript와 React를 사용하여 앱을 제작하는 무료 및 오픈소스 플랫폼

- <https://expo.io/learn>

Expo 이외의 개발 방법

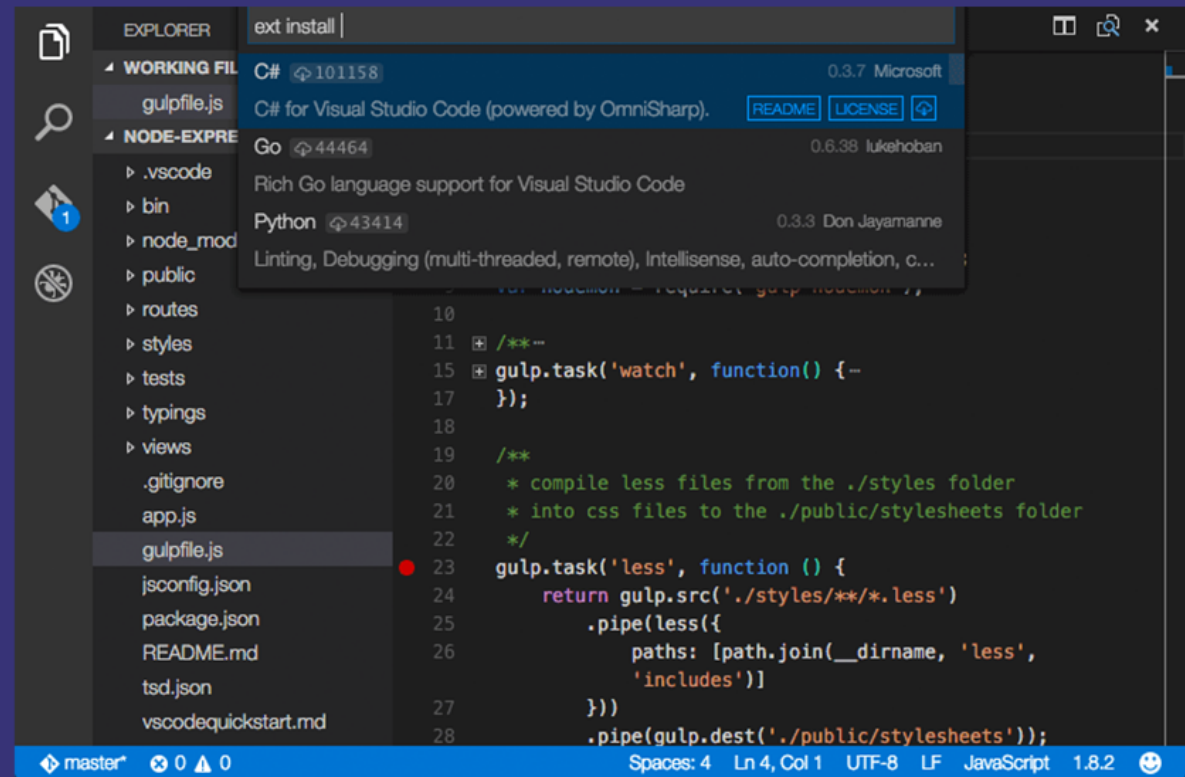
- `react-native cli` 를 이용하여 제작
- 네이티브 코드를 수정해야 한다면 `react-native cli` 로 프로젝트 생성하거나 expo로 생성한 프로젝트를 eject 해야 함

Expo 설치 단계

- git 설치 : <https://git-scm.com>
- Node.js 설치
- Expo CLI 설치
- 스마트폰에 Expo 앱 설치
- Expo를 이용하여 프로젝트 생성 및 내 폰에서 실행
- 참고 : <https://expo.io/learn>

Expo 계정 설정

- Expo 가입
- Expo CLI 에서 로그인
- Expo 앱에서 로그인



Q1-2. 리액트 네이티브를 위한 VS Code 설치/설정?

VS Code 설치 및 설정

Visual Studio Code

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux.

- Microsoft의 오픈소스
- JavaScript
- 다양한 확장 기능(Extension)
 - MS가 직접 만드는 react native를 위한 확장 기능

설치 방법

- <https://code.visualstudio.com/download>

리액트 네이티브 개발을 도와주는 VS Code 확장 도구 설치

- react native tools
- eslint
- prettier - code formatter
- 참고 : <https://class.codejong.kr/t/topic/90>

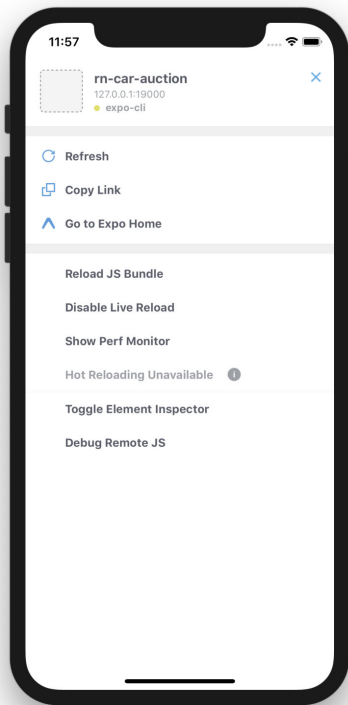
Q1-3. Expo와 VS Code를 이용한 개발 과정

0. 프로젝트 생성 및 실행

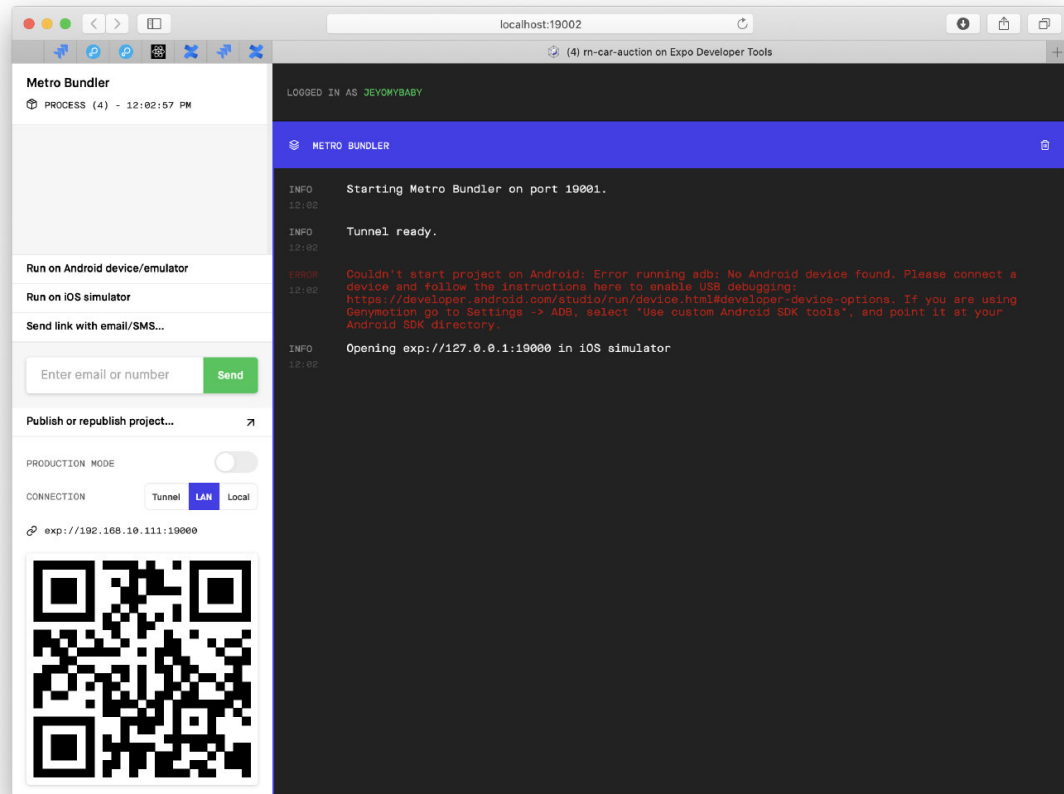
```
expo init my-new-project  
cd my-new-project  
expo start
```

1. 시뮬레이터나 디바이스에서 expo 앱 실행하기
2. VSCode를 해당 폴더에서 열기 : `code .`
3. 코드 변경하고 변경사항 확인하기

잠깐) Expo 개발자 메뉴에 대해 자세히 알아보기

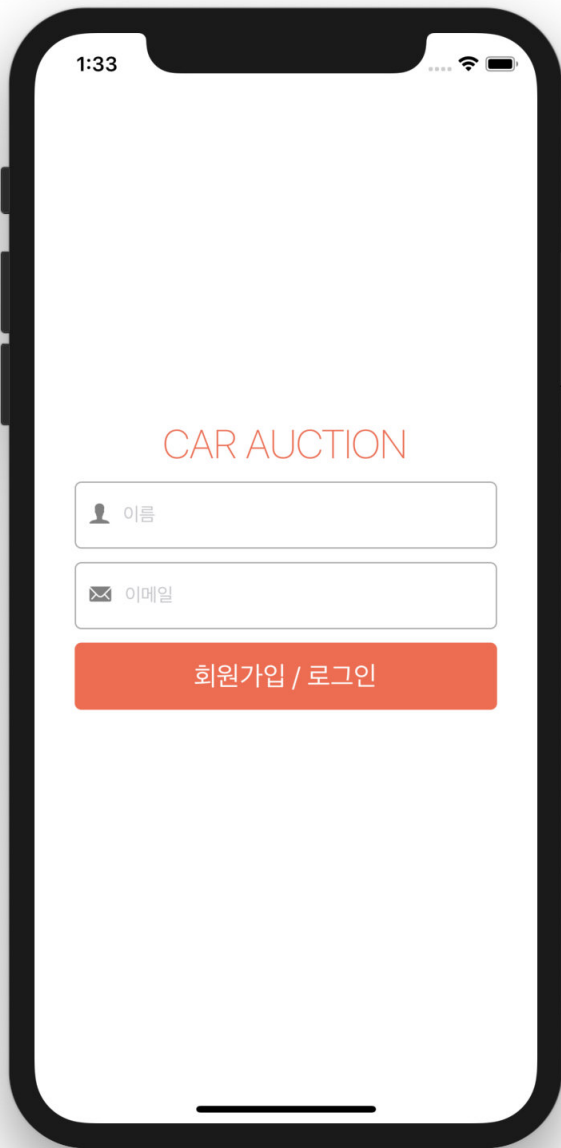


iPhone X - 12.1



잠깐) 에러의 원인을 확인하는 방법

잠깐) 기본 프로젝트 파일 둘러보기



iPhone X - 12.1

목표 3.
로그인 화면 UI 만든다.

목표 2 Part 1 : 스타일 지정 기초

Q2-1. 화면을 표현하는 단위? (컴포넌트란?)

Q2-2. 스타일 지정 방법은?

Q2-3. Layout : 위치 및 크기 지정 방법은?

Q2-4. 스타일 관리방법?

목표 2 Part 2 : 필수 컴포넌트 및 컴포넌트 제작 기초

Q2-5. 글자를 표현하는 방법은?

Q2-6. 이미지를 표현하는 방법은?

Q2-7. 아이콘을 표시하는 방법은?

Q2-8. 사용자에게 키보드 입력은 어떻게?

Q2-9. 재사용을 위한 컴포넌트제작 방법은?

Q2-10. 버튼 처럼 터치를 다루는 방법?

Q2-1. 화면을 표현하는 단위? (컴포넌트란?)

리액트 컴포넌트

잠깐Q) 함수란 무엇인가?

잠깐A) 함수는...

- 입력 => (do something) => 출력
- 입력은 인자, 출력은 리턴 값

```
function add(a, b){  
  return a+b;  
}
```

- 나중에 알아 볼 내용
 - Pure Function
 - High Order Function

컴포넌트의 기본 모양

```
class HelloMessage extends React.Component {  
  render() {  
    return (  
      <div>  
        Hello {this.props.name}  
      </div>  
    );  
  }  
}  
  
ReactDOM.render(  
  <HelloMessage name="Taylor" />,  
  mountNode  
);
```

- 컴포넌트에서 무엇이 입력이고 무엇이 출력일까?

개념적으로 컴포넌트는 자바스크립트 함수와 같다.

Conceptually, components are like JavaScript functions. They accept arbitrary inputs (called "props") and return React elements describing what should appear on the screen.

출처: [react doc](#)

HTML 엘리먼트와 리액트 네이티브 컴포넌트의 유사성

리액트	리액트 네이티브
div	View
span	Text
li, ul	FlatList, 자식 아이템
img	Image

- 대체할 수 없지만 비슷한 용도로 쓰임

로그인 화면의 코드 일부

```
<View style={styles.container}>
  <Text style={styles.titleText}>
    CAR AUCTION
  </Text>
  <IconTextInput
    style={{ marginTop: 10 }}
    iconName={'ios-person'}
    placeholder={'이름'}
  />
  <IconTextInput
    style={{ marginTop: 10 }}
    iconName={'ios-mail'}
    placeholder={'이메일'}
  />
  <RoundButton
    style={{ marginTop: 10 }}
    title={'회원가입 / 로그인'}
    onPress={() => {
      //do signin/signup
    }}
  />
</View>
```

잠깐Q) JSX란?

마크업, 로직, 스타일까지 한 파일에 작성

JSX는 기술에 따라 코드를 분리하기보다

하는 일에 따른 분리 하는 것을 더 중요하게 생각했다.

잠깐Q) JSX를 쓸 때 vs 안쓸 때

```
class HelloMessage extends React.Component {
  render() {
    return React.createElement(
      "div",
      null,
      "Hello ",
      this.props.name
    );
  }
}
```

```
ReactDOM.render(React.createElement(HelloMessage, { name:
```

```
class HelloMessage extends Component {
  render() {
    // Instead of calling createElement, we return ma
    return <div>Hello {this.props.name}</div>;
  }
}
```

// 더 이상 createElement를 여기서 호출할 필요가 없다

```
ReactDOM.render(<HelloMessage name="Bonnie" />, mountNode
```

Q2-2. 스타일 지정 방법은?

- inline
- JS Object
- StyleSheet.create

- 인라인 스타일

```
<Text>  
  The quick <Text style={{fontStyle: "italic"}}>brown</Text>  
</Text>
```

- 오브젝트로 스타일 정의

```
const bold = {  
  fontWeight: "bold"  
};
```

- StyleSheet.create 이용

```
const styles = StyleSheet.create({  
  button: {  
    borderRadius: "8px",  
    backgroundColor: "#99CCFF"  
  }  
});
```

어떤 방법으로 스타일을 지정해야 하나?

- 오브젝트로 스타일 정의하기 보다 StyleSheet.create 이용
 - 생성한 스타일이 네이티브 영역에 존재하고 해당 스타일의 id만 브릿지를 통해서 전달함 **예정**
 - It also allows to send the style only once through the bridge. All subsequent uses are going to refer an id (**not implemented yet**).
- 인라인은 아주 요긴

잠깐Q. 레이아웃이란?

잠깐A.

- 위치 (position)
- 크기 (size)

Q2-3. Layout : 위치 및 크기 지정 방법은?

flex를 알아보자!

flex 레이아웃 #1 위치(position)

- 자식들의 위치
 - `flexDirection`
 - `justifyContent`
 - `alignItems`
 - `alignContent`
 - `flexWrap`
- 나 자신의 위치
 - `alignSelf` : 부모가 정해진 정렬(`alignItems`) 말고 내 방식대로
 - `alignContent`

flexDirection

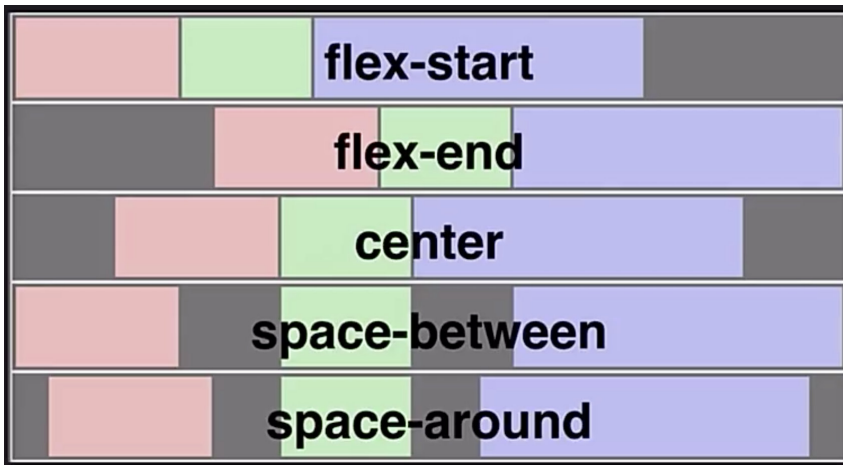
자식들을 어느 방향을 배치할 것인가?

- **row** : 왼쪽에서 오른쪽
 - row-reverse
- **column** : 위에서 아래
 - column-reverse

justifyContent

자식들을 배치 방향에서 어느쪽으로 정렬 할 것인가?

- flex-start (기본값)
- flex-end
- space-between
- space-around
- space-evenly



alignItems

flexDirection과 직교하는 방향에서 어느쪽으로 정렬 할 것인가?

- flex-start
- flex-end
- center
- stretch
- base-line ([참고](#))

flexWrap

공간이 비좁을 때 자식을 어떻게 하지?

- nowrap : 모든 요소들을 한 줄에 정렬합니다.
- wrap : 요소들을 여러 줄에 걸쳐 정렬합니다.
- wrap-reverse : 요소들을 여러 줄에 걸쳐 반대로 정렬합니다.

개구리 게임 실습

- <https://flexboxfroggy.com/#ko>
- RN과 상관 없는 단계
 - 14 단계 답 : `order: 1;`
 - 15 단계 답 : `order: -1;`

CSS	RN Style
order 속성	X
<code>flex-direction: row;</code>	<code>flexDirection: 'row',</code>
<code>justify-content: flex-end;</code>	<code>justifyContent: 'flex-end',</code>

flex 레이아웃 #1 크기(size)

- flex
- flexGrow
- flexShrink
- flexBasis
- 참고 문서
 - [how flexbox works - freecodecamp](#)

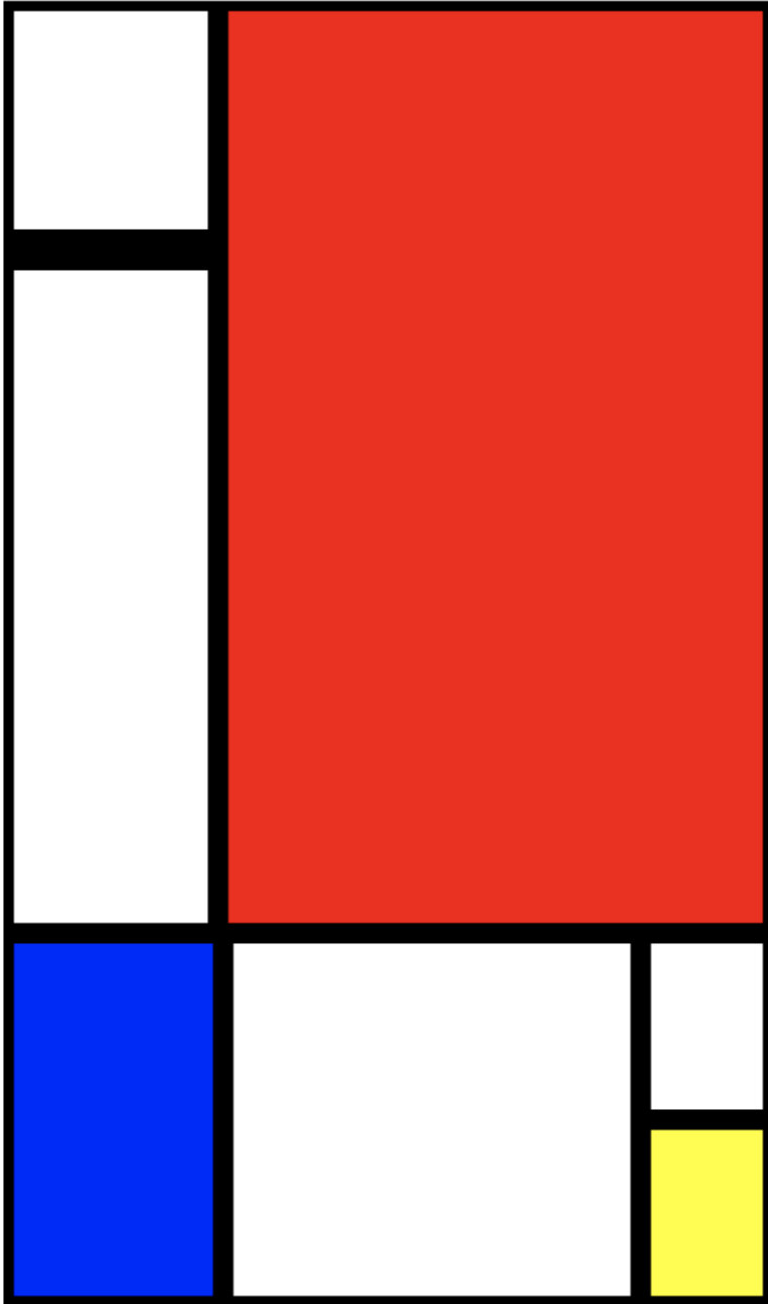
24단계 힌트

```
flex-direction: ;  
flex-wrap: ;  
justify-content: ;  
align-content: ;
```

24단계 정답

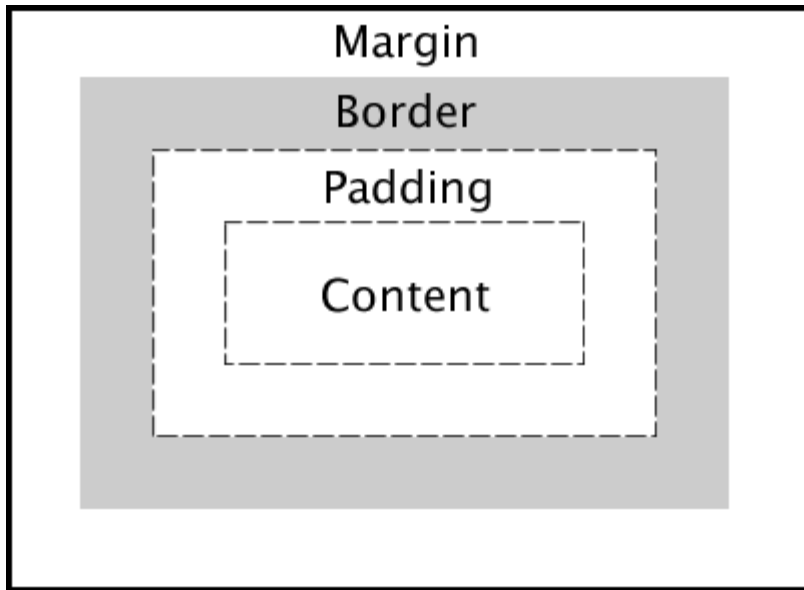
```
flex-direction: column-reverse;  
flex-wrap: wrap-reverse;  
justify-content: center;  
align-content: space-between;
```

레이아웃 종합 실습 : Mondrian 따라하기



margin, border, padding

어디까지가 size(height, width) 인가?



- 어디까지 height, width에 포함될까?
- 영역 밖의 여백?
- 영역 안의 여백?

Q2-4. 스타일 관리방법?

- 스타일 병합 / 적용

스타일 사용 팁

- 스타일 병합

```
<Text style={ [ styles.button, styles.accentText ] }>  
  Wow  
</Text>
```

- 조건에 따른 적용

```
<View style={ [  
  styles.button,  
  this.state.touching && styles.highlight  
] } />
```

Q 1/3

잠깐 : HTML에서 글자(Text)는 어떻게 넣었나요?

: 아무데나~!!

Q 2/3


잠깐 : HTML에서 글자(Text) 스타일이 적용된 Tag는?

: h1, h2, em, strong...

Q2-5. 글자를 표현하는 방법은?

: 오직 <Text> !!!!!!!

<Text> 1/3

-  h1, em, strong 등이 없다.
- <Text> 컴포넌트만이 플레인 텍스트 노드를 자식으로 가질 수 있다.
- 중첩 가능
- 일부 스타일이 상속된다. (Text 트리에서만 가능)

```
<Text style={{fontWeight: 'bold'}}>
  I am bold
  <Text style={{color: 'red'}}>
    and red
  </Text>
</Text>
```

```
/*
"I am bold and red"
0-9: bold
9-17: bold, red
*/
```

```
<Text style={{fontWeight: 'bold', color: 'red'}}>  
  성공!  
</Text>
```

```
<Text style={{fontWeight: 'bold', color: 'red'}}>  
  강조!  
</Text>
```

잠깐 : 반복적으로 사용하는 텍스트 서식은 어떻게 재사용?

잠깐 A : 스타일이 적용된 컴포넌트를 만들어서 재사용

```
class Em extend Component {  
  render () {  
    return (  
      <Text style={{fontWeight: 'bold', color:'red'}}>  
        {this.props.children}  
      </Text>  
    )  
  }  
}
```


<Text> 2/3

- 스타일이 적용된 컴포넌트를 만들어서 재사용

```
<Text>  
The quick <Em>brown</Em> fox jumped over the lazy <Strong>  
</Text>
```

- 오동작을 막기 위해서는 text는 text와 관련된 일만 하도록 권장 ([관련 jsdev.kr 글](#))
- iOS / Android 전용 속성이 많다: 주의
 - `adjustsFontSizeToFit` & `minimumFontScale` : iOS only
 - `textBreakStrategy` : Android only
- 개행 `{'\n'}` , 스페이스 `{' '}` 혹은 ` `

더 자세한 사항은 `<Text />` [공식 문서](#)

<Text> 3/3

- 자주 사용하는 스타일

```
{  
  fontSize: 12,  
  fontWeight: 300,  
  color: '#aaa',  
  textDecorationLine: 'underline'  
}
```

- 지원하는 font style 모음 [공식 문서](#)

