

2020.12.12

react-native

github에 올리기위에 이제 디렉토리를

```
/Users/mac/documents/github
```

로 옮겨서 작업하기로 했다.

```
nvm use 10.15.1  
  
react-native init --version 0.61.5 react_nativie_01
```

react-native의 화면 출력은

App.js을 통해서 만들어짐

index.js를 보면 앱의 메인 입구. 앱을 시작하는 곳

첫번째 인자는 앱 이름, app.json에서 임포트

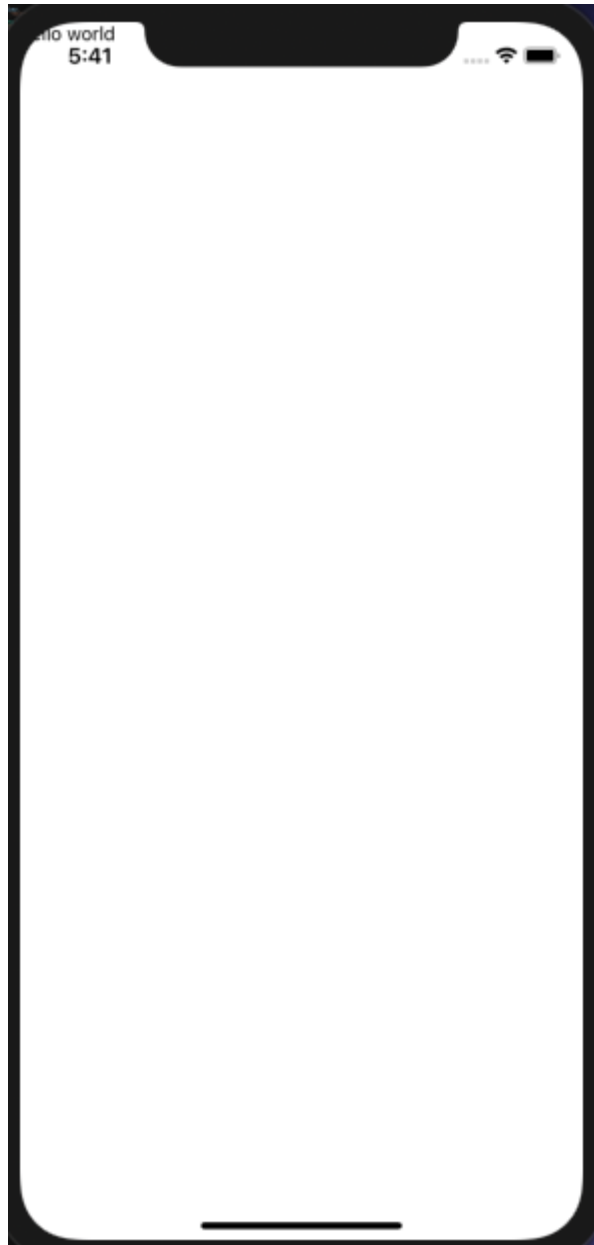
두번째 인자는 앱을 시작할때 렌더링 되는것, App을 가리킴

App.js 수정

```
/**  
 * Sample React Native App  
 * https://github.com/facebook/react-native  
 *  
 * @format  
 * @flow  
 */  
  
import React, { Component } from 'react';  
import {
```

```
SafeAreaView,  
StyleSheet,  
ScrollView,  
View,  
Text,  
StatusBar,  
} from 'react-native';  
  
class App extends Component {  
  render() {  
    return (  
      <View>  
        <Text>hello world</Text>  
      </View>  
    )  
  }  
}  
  
export default App;
```

로 수정후 저장,



좌측 상단에 hello world가 나타남.

View, Style, Text

App.js에서는

react라는 모듈에서 components라는 클래스를 임포트하고

Component를 상속받는 App 이라는 클래스를 만든 것

이 App안에는 화면을 렌더링하는 함수가 있고 리턴되는 것들이 화면을 구성하는 것임.

```

/**
 * Sample React Native App
 * https://github.com/facebook/react-native
 *
 * @format
 * @flow
 */

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

class App extends Component {
  render() {
    return (
      <View>
        <Text>hello world</Text>
      </View>
    )
  }
}

export default App;

```

View 와 Text는 react-native라는 모듈에서 임포트해서 가져온 것.

화면을 출력하기 위해서는 View라는 태그에 감싸져 있어야함

View는 다른 컴포넌트들을 감싸는 역할을 함

view안에 view가 있어도 됨.

```

class App extends Component {
  render() {
    return (
      <View>
        <View>
          <Text>hello world</Text>
        </View>
        <View>
          <Text>hello world</Text>
        </View>
        <View>
          <Text>hello world</Text>
        </View>
      </View>
    )
  }
}

```

Style을 주는 방법에는 두 가지가 있음.

1) in-line 스타일링

태그 안에 추가 코딩을 하는 방법

```
<View style={{}}>
```

첫번째 중괄호는 jsx를 두번째 중괄호는 객체를 담는 것

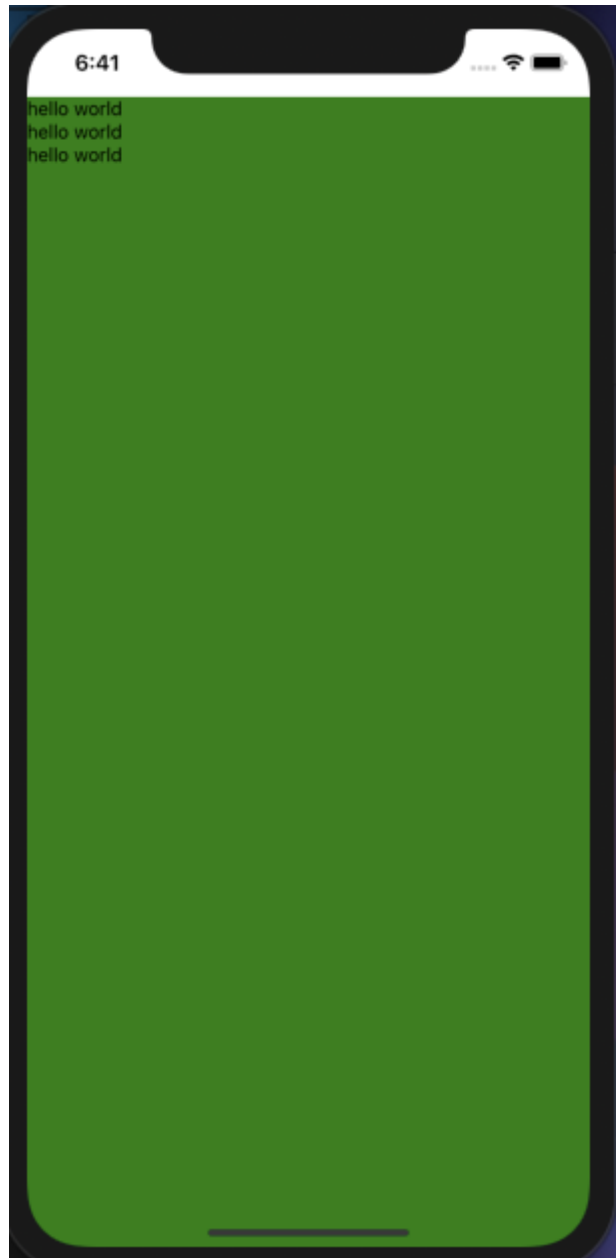
```
/**
 * Sample React Native App
 * https://github.com/facebook/react-native
 *
 * @format
 * @flow
 */

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

class App extends Component {
  render() {
    return (
      <View style={{
        backgroundColor: 'green',
        marginTop: 50,
        height: '100%'
      }}>
        <View >
          <Text>hello world</Text>
        </View>
        <View>
          <Text>hello world</Text>
        </View>
        <View>
          <Text>hello world</Text>
        </View>
      </View>
    )
  }
}
```

```
}  
}  
  
export default App;
```

margin은 픽셀 단위가 아니라 스마트폰 각자의 고유 단위 (아마 dp?)



marginTop이라는 프로퍼티로 밑으로 좀 내려옴

```

/**
 * Sample React Native App
 * https://github.com/facebook/react-native
 *
 * @format
 * @flow
 */

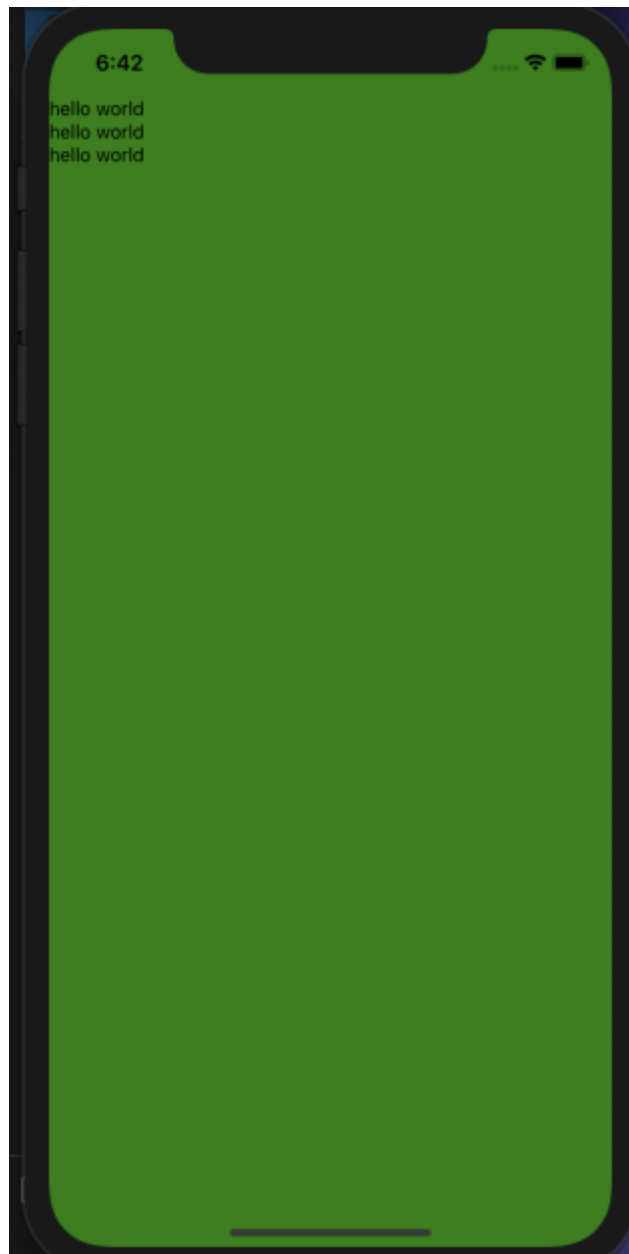
import React, { Component } from 'react';
import {
  View,
  Text
} from 'react-native';

class App extends Component {
  render() {
    return (
      <View style={{
        backgroundColor: 'green',
        //marginTop: 50,
        paddingTop: 50,
        height: '100%'
      }}>
        <View >
          <Text>hello world</Text>
        </View>
        <View>
          <Text>hello world</Text>
        </View>
        <View>
          <Text>hello world</Text>
        </View>
      </View>
    )
  }
}

export default App;

```

marginTop → paddingTop



marginTop은 뷰가 다른 컴포넌트와 간격을 띄워줘야할 때

paddingTop은 다른 컴포넌트 뷰안에 있는 다른 컴포넌트와의 간격을 띄워줘야 할 때

react는 float 와 같은 부동소수점은 이해 하지 못함 (정수만 이해)

2)아래에 따로 빼서 스타일링 하는 방법

flex는 화면을 채우는 컴포넌트들간의 차지하는 영역의 비율 같은 것을 나타내는 지표


```

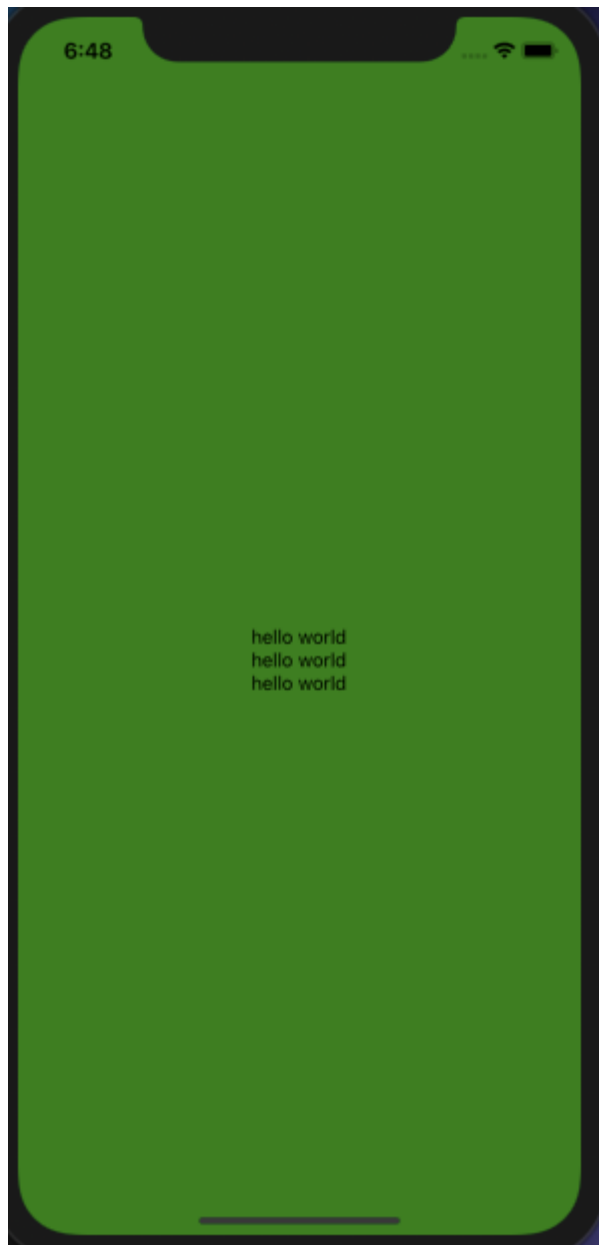
/**
 * Sample React Native App
 * https://github.com/facebook/react-native
 *
 * @format
 * @flow
 */

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

class App extends Component {
  render() {
    return (
      <View style={styles.mainView}>
        <View >
          <Text>hello world</Text>
        </View>
        <View>
          <Text>hello world</Text>
        </View>
        <View>
          <Text>hello world</Text>
        </View>
      </View>
    )
  }
}

const styles = StyleSheet.create({
  mainView: {
    flex: 1,
    backgroundColor: 'green',
    //marginTop: 50,
    paddingTop: 50,
    alignItems: 'center',
    justifyContent: 'center'
  }
})
export default App;

```



alignItems: 수평

justifyContent: 수직

style 2

```
/**
 * Sample React Native App
 * https://github.com/facebook/react-native
 */
```

```

* @format
* @flow
*/

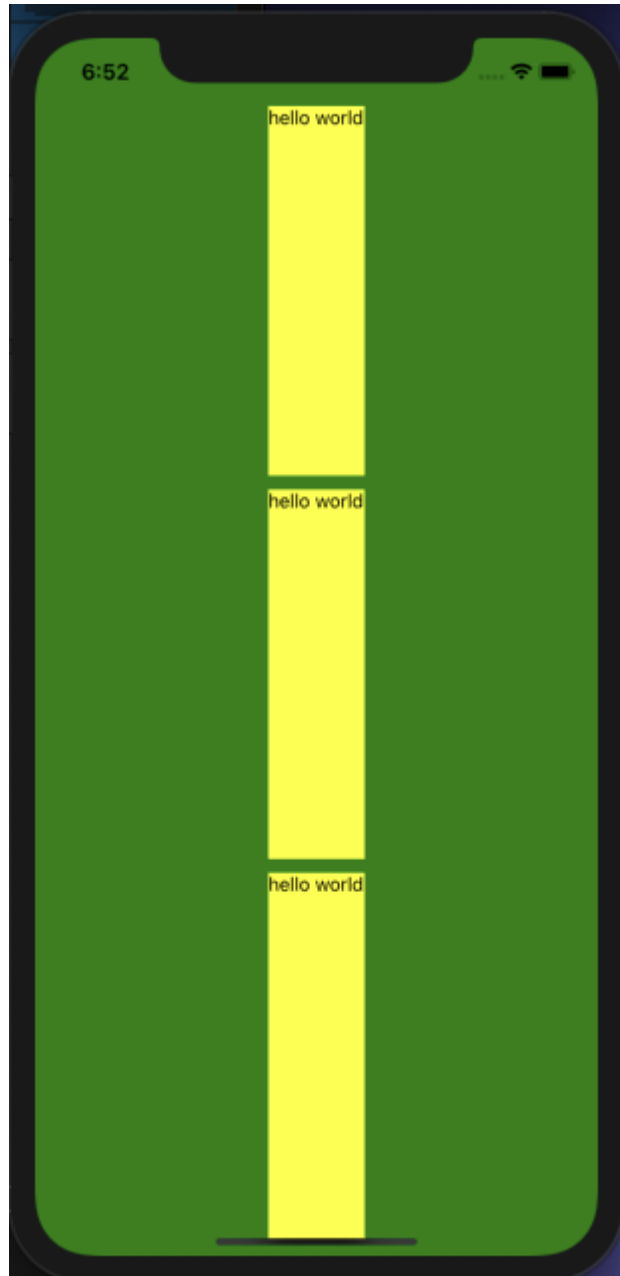
import React, { Component } from 'react';
import {
  View,
  Text,
  StyleSheet
} from 'react-native';

class App extends Component {
  render() {
    return (
      <View style={styles.mainView}>
        <View style={styles.subView}>
          <Text>hello world</Text>
        </View>
        <View style={styles.subView}>
          <Text>hello world</Text>
        </View>
        <View style={styles.subView}>
          <Text>hello world</Text>
        </View>
      </View>
    )
  }
}

const styles = StyleSheet.create({
  mainView: {
    flex: 1,
    backgroundColor: 'green',
    //marginTop: 50,
    paddingTop: 50,
    alignItems: 'center',
    justifyContent: 'center'
  },
  subView: {
    flex: 1,
    backgroundColor: 'yellow',
    marginBottom: 10
  }
})

export default App;

```



subView 생성 후 각 뷰에 적용

스타일 각각 다르게 설정하기

```
/**
 * Sample React Native App
 * https://github.com/facebook/react-native
 *
 * @format
 * @flow
 */
```

```

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

class App extends Component {
  render() {
    return (
      <View style={styles.mainView}>
        <View style={styles.subView}>
          <Text>hello world</Text>
        </View>
        <View style={styles.subView}>
          <Text>hello world</Text>
        </View>
        <View style={styles.anotherView}>
          <Text>hello world</Text>
        </View>

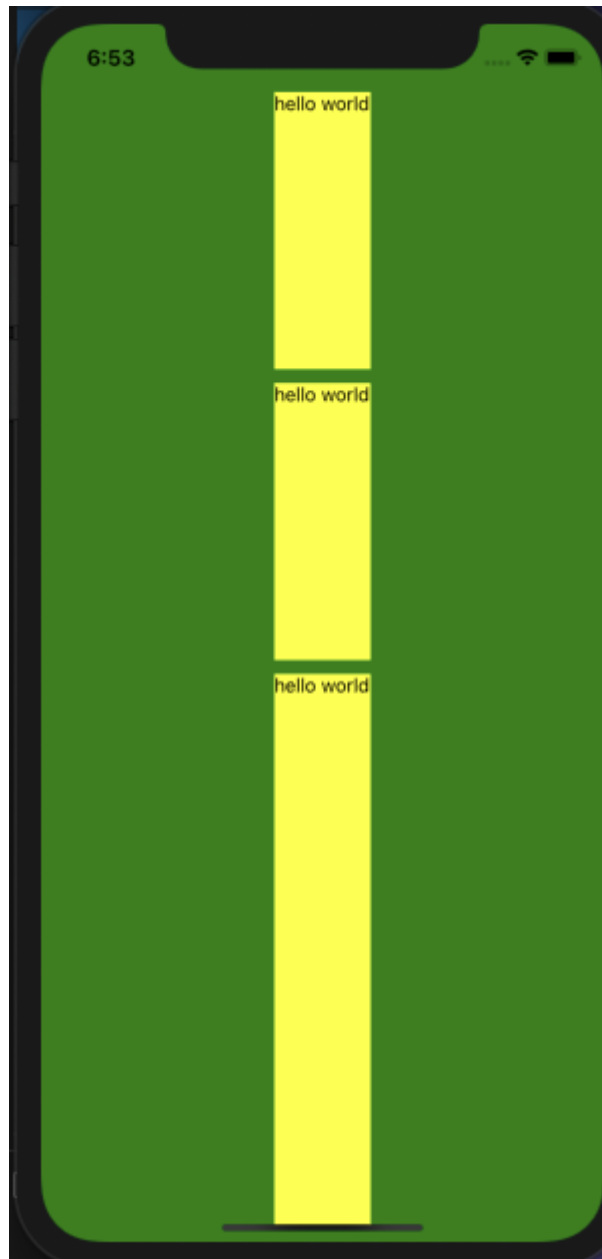
      </View>
    )
  }
}

const styles = StyleSheet.create({
  mainView: {
    flex: 1,
    backgroundColor: 'green',
    //marginTop: 50,
    paddingTop: 50,
    alignItems: 'center',
    justifyContent: 'center'
  },
  subView: {
    flex: 1,
    backgroundColor: 'yellow',
    marginBottom: 10
  },
  anotherView: {
    flex: 2,
    backgroundColor: 'yellow',
    marginBottom: 10
  }
})

export default App;

```

마지막 뷰만 flex 2 설정



전체를 1/1/2로 나눠서 flex 값을 가짐

```
/**
 * Sample React Native App
 * https://github.com/facebook/react-native
 *
 * @format
 * @flow
```

```

*/

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

class App extends Component {
  render() {
    return (
      <View style={styles.mainView}>
        <View style={styles.subView}>
          <Text style={styles.mainText}>hello world</Text>
        </View>
        <View style={styles.subView}>
          <Text>hello world</Text>
        </View>
        <View style={styles.anotherView}>
          <Text style={styles.mainText}> hello world</Text>
        </View>

      </View>
    )
  }
}

const styles = StyleSheet.create({
  mainView: {
    flex: 1,
    backgroundColor: 'green',
    //marginTop: 50,
    paddingTop: 50,
    alignItems: 'center',
    justifyContent: 'center'
  },
  subView: {
    flex: 1,
    backgroundColor: 'yellow',
    marginBottom: 10
  },
  anotherView: {
    flex: 2,
    backgroundColor: 'yellow',
    marginBottom: 10,
    width: '100%',
    alignItems: 'center',
    justifyContent: 'center'
  },
  mainText: {
    fontSize: 20,
    fontWeight: 'bold',
    color: 'red',
    padding: 20
  }
})

```

```
})  
export default App;
```

TextStyle 주기

