

2020.12.09

state

스테이트는 컴포넌트에서 렌더링되는 데이터를 담고 유지 관리하는 자바스크립트 객체
중요!

스테이트의 값에 따라 화면에 보여지는 아웃풋이 달라짐

App.js

```
/**
 * 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 style={styles.background}>
        <Text> Hello World</Text>
      </View>
    )
  }
}

const styles = StyleSheet.create({
  background: {
    flex: 1,
    backgroundColor: 'ffffff',
    alignItems: 'center',
    justifyContent: 'center'
  }
})
export default App;
```

npm start → new terminal → nvm use 10.15.1 → react-native run-ios

state는

```
/**
 * 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 {

  state = {
    sampleText: 'Hello world'
  }

  render() {
    return (
      <View style={styles.background}>
        <Text> {this.state.sampleText}</Text>
      </View>
    )
  }
}

const styles = StyleSheet.create({
  background: {
    flex: 1,
    backgroundColor: '#ffff',
    alignItems: 'center',
    justifyContent: 'center'
  }
})
export default App;
```

sample text를 선언하고 {this.state.sampleText}라는 변수를 출력
위 코드와 결과는 같음

변수에 할당해서 필요할 때 갖다 쓰는 것이 더 수정에도 용이하고 효율적임
-this는 자바스크립트 문법으로 상위 스코프를 가리킴-

state 문법 2

함수 넣기

```
/**
 * 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 {

  state = {
    sampleText: 'Hello world',
    sampleBoolean: true
  }

  inputText = () => (
    this.state.sampleBoolean ?
      <Text>sampleBoolean is ture</Text> //true 일 때
    :
      <Text>sampleBoolean is false</Text> //false 일 때
  )

  render() {
    return (
      <View style={styles.background}>
        {this.inputText()} // 함수
      </View>
    );
  }
}
```

```

        </View>
      )
    }
  }

  const styles = StyleSheet.create({
    background: {
      flex: 1,
      backgroundColor: 'ffff',
      alignItems: 'center',
      justifyContent: 'center'
    }
  })
  export default App;

```

setstate

state는 직접 변경을 하면 안되는 특성을 갖고 있음

setstate라는 메소드를 사용해야 함 그래야만 스테이트 값이 갱신될 때 갱신

스테이트는 렌더링 되는 데이터를 관리하기 때문에 일부러 직접 변경될 수 없는 특성을 가짐

이벤트 만들기

```

changeState = () => (
  this.setState({
    sampleText: 'Text changed!!'
  })
)

render() {
  return (
    <View style={styles.background}>
      <Text onPress={this.changeState}> //onPress 누르면
        {this.state.sampleText}
      </Text>
    </View>
  )
}

```

텍스트를 누르면 텍스트가 바뀜

```

changeState = () => {
  if (!this.state.sampleBoolean) {
    this.setState({
      sampleText: 'Text changed!!',
      sampleBoolean: true
    })
  } else {
    this.setState({
      sampleText: 'Hello World!!!',
      sampleBoolean: false
    })
  }
}
}

```

누를때마다 번갈아가면서 텍스트가 바뀜

비동기성

setstate는 성능향상을 위해 단일 업데이트를 지원함 데이터를 변경할 때에는 현재 버전을 카피한 다음에 업데이트를 진행함

```

onAdd = () => {
  this.setState({
    sampleNum: sampleNum + 1
  })
}
render() {
  return (
    <View style={styles.background}>
      <Text onPress={this.onAdd}>
        {this.state.sampleNum}
      </Text>
    </View>
  )
}
}

```

예러가 뜸

setstate를 통한 데이터 값의 변경을 위해서는 현재버전을 카피하고 업데이트를 진행함

prevState를 setState의 첫번째 인자로 넣어야함

```
onAdd = () => {
  this.setState(prevState => {
    return {
      sampleNum: prevState.sampleNum + 1
    }
  })
}
render() {
  return (
    <View style={styles.background}>
      <Text onPress={this.onAdd}>
        {this.state.sampleNum}
      </Text>
    </View>
  )
}
```

이렇게 전 버전의 state를 이용해서 변경을 해 줘야함

Props

props는 read only 수정 변경이 불가능한 읽기전용 프로퍼티

부모 자식 관계가 형성되어야 의미가 있음

부모가 가지고 있는 정보가 똑같이 100명의 자식에게 손쉽게 전달할 수 있는 수단

propsChild.js 생성

부모가 가지고 있는 데이터를 자식에게 간편하게 전달하는 방법

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

const PropsChild = (props) => {
  return (
    <View>
      <Text onPress={props.changeState}>
        {props.sampleText}
      </Text>
    </View>
  )
}
```

```
        </View>
    )
}
export default PropsChild;
```

propsChild