

Touch Event (2020.12.12)

main 디렉토리에

src 라는 소스 디렉토리를 생성

header.js 파일 생성

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

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

const Header = () => (
  <View style={styles.header}>
    <Text>This is header</Text>
  </View>
)

const styles = StyleSheet.create({
  header: {
    backgroundColor: 'pink',
    alignItems: 'center',
    padding: 5,
    width: '100%'
  }
});
export default Header;
```

외부파일에 작업한 걸 импорт 하는 방법을 익히는 과정

App.js 에서 header.js 파일 импорт해서 스타일 적용하기

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

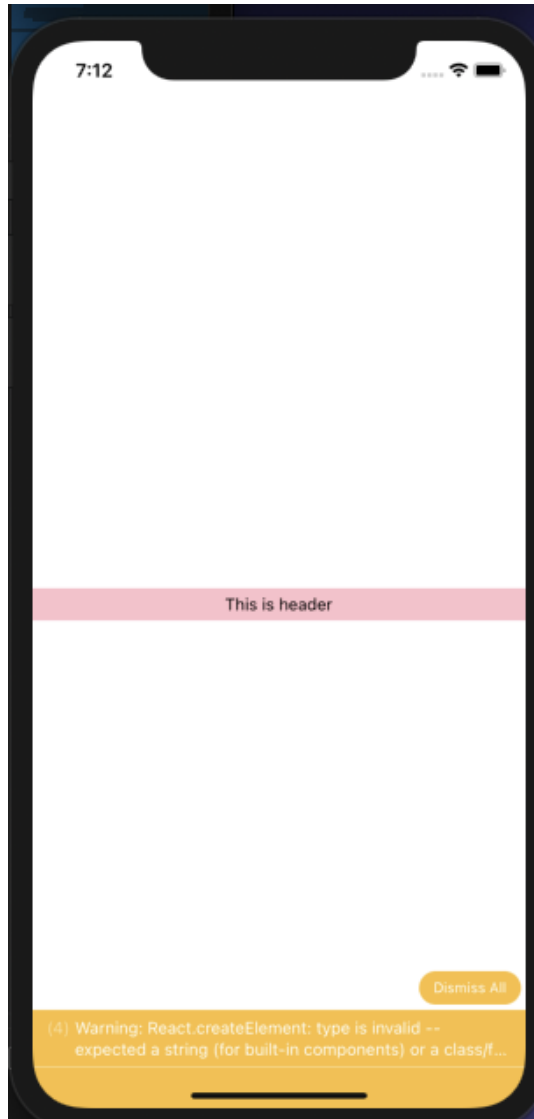
import React, { Component } from 'react';
import {
  View,
  Text,
  StyleSheet
} from 'react-native';
import Header from './src/header';
```

```

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

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

```



괄호 차이

```
exampleFunction = () => {  
  }  
  
exampleFunction = () => (  
  )
```

중괄호는 리턴되는 js 컴포넌트가 없는 것

소괄호는 jsx를 리턴할 수 있음

리턴되는 컴포넌트가 없으면 오류가 뜸

Jsx = javascript xml

State를 이용해서 텍스트 바꾸기

App.js

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

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

const Header = (props) => (
  <TouchableOpacity
    style={styles.header}
    onPress={()=>alert('Hello World')}
  >
    <View style={styles.header}>
      <Text>{props.name}</Text>
    </View>
  </TouchableOpacity>
)

const styles = StyleSheet.create({
  header: {
    backgroundColor: 'pink',
    alignItems: 'center',
    padding: 5,
    width: '100%'
  }
})
export default Header;
```

header.js

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

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

const Header = (props) => (
  <TouchableOpacity
    style={styles.header}
    onPress={()=>alert('Hello World')}
  >
    <View style={styles.header}>
      <Text>{props.name}</Text>
    </View>
  </TouchableOpacity>
)

const styles = StyleSheet.create({
  header: {
    backgroundColor: 'pink',
    alignItems: 'center',
    padding: 5,
    width: '100%'
  }
})
export default Header;
```

TouchableWithoutFeedback 으로 변경

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

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

const Header = (props) => (
  <TouchableWithoutFeedback

    onPress={()=>alert('Hello World')}
  >
    <View style={styles.header}>
      <Text>{props.name}</Text>
    </View>
  </TouchableWithoutFeedback>
)

const styles = StyleSheet.create({
  header: {
    backgroundColor: 'pink',
    alignItems: 'center',
```

```

padding: 5,
width: '100%'
}
})
export default Header;

```

onPress → onLongPress (길게 누르기로 변경)

```

const Header = (props) => (
  <TouchableOpacity
    style={styles.header}
    //onPress={()=>alert('Hello World')}
    onLongPress={()=>alert('Hello World')}
  >
    <View >
      <Text>{props.name}</Text>
    </View>
  </TouchableOpacity>
)

```

onLongPress → onPressIn (누르자마자 즉각 반응 변경)

```

const Header = (props) => (
  <TouchableOpacity
    style={styles.header}
    //onPress={()=>alert('Hello World')}
    //onLongPress={()=>alert('Hello World')}
    onPressIn={()=>alert('Hello World')}
  >
    <View >
      <Text>{props.name}</Text>
    </View>
  </TouchableOpacity>
)

```

onPressIn → onPressOut (손을 뗄 때 변경)

```

const Header = (props) => (
  <TouchableOpacity
    style={styles.header}
    //onPress={()=>alert('Hello World')}
    //onLongPress={()=>alert('Hello World')}
    //onPressIn={()=>alert('Hello World')}
    onPressOut={()=>alert('Hello World')}
  >
    <View >
      <Text>{props.name}</Text>
    </View>
  </TouchableOpacity>
)

```

```
    </TouchableOpacity>  
  )  
}
```

View 가 아닌 Text를 누를 때 반응 하도록 변경

App.js에서

```
render() {  
  return (  
    <View style={styles.mainView}>  
      { /* <Header name={this.state.appName}/> */}  
      <Text  
        style={styles.mainView}  
        onPress={()=>alert('text Touch event')}>  
        Hello World  
      </Text>  
    </View>  
  )  
}
```