

# Button (2020.12.12)

src/generaor.js 생성

Button 에는 Tilt이라는 속성이 필수적으로 들어가 있어야 함. (버튼에 새겨질 글자)

IOS와 ANDROID의 버튼 디폴트 모양이 다름

## App.js

```
/**
 * 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';
import Generator from './src/generator';

class App extends Component {

  state = {
    appName: 'My First App',
    random: [36, 999]
  }

  onAddRandomNum = () => {
    alert('add random number!!!')
  }

  render() {
    return (
      <View style={styles.mainView}>
        <Header name={this.state.appName}/>

```

```

        <View>
          <Text style={styles.mainText}
            onPress={()=>alert('text Touch event')}>
            Hello World</Text>
        </View>

        <Generator add={this.onAddRandomNum}/>
      </View>
    )
  }
}

const styles = StyleSheet.create({
  mainView: {
    flex: 1,
    backgroundColor: 'white',

    paddingTop: 50,
    alignContent: 'center',
    justifyContent: 'center'
  },

  mainText: {
    fontSize: 20,
    fontWeight: 'normal',
    color: 'red',
    padding: 20,
    alignContent: 'center',
    justifyContent: 'center'
  }
})
export default App;

```

## generaotr.js

```

/**
 * 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';

```

```

import Generator from './src/generator';

class App extends Component {

  state = {
    appName: 'My First App',
    random: [36, 999]
  }

  onAddRandomNum = () => {
    alert('add random number!!!')
  }

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

        <Generator add={this.onAddRandomNum}/>
      </View>
    )
  }
}

const styles = StyleSheet.create({
  mainView: {
    flex: 1,
    backgroundColor: 'white',

    paddingTop: 50,
    alignContent: 'center',
    justifyContent: 'center'
  },

  mainText: {
    fontSize: 20,
    fontWeight: 'normal',
    color: 'red',
    padding: 20,
    alignContent: 'center',
    justifyContent: 'center'
  }
})
export default App;

```

## Button 2

난수 띄우기

### numlist.js 를 src 디렉토리에 추가

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

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

const NumList = (props) => {
  return (
    <View style={styles.numList}>
      <Text>Hello Again</Text>
    </View>
  )
}

const styles = StyleSheet.create({
  numList: {
    backgroundColor: '#cecece',
    alignItems: "center",
    padding: 5,
    width: '100%',
    marginTop: 5
  }
})
export default NumList;
```

App.js 에서 numlist.js 사용을 위해 import

```
import NumList from './src/numlist';
```

render 함수 안에 넣어준다.

```

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

      <Generator add={this.onAddRandomNum}/>
      <NumList num={this.state.random}/>
    </View>
  )
}

```

그 다음 numlist.js 에서 App.js 의 state의 random 배열에 있는 요소들을 화면에 출력하기 위해 props로 받아온다.

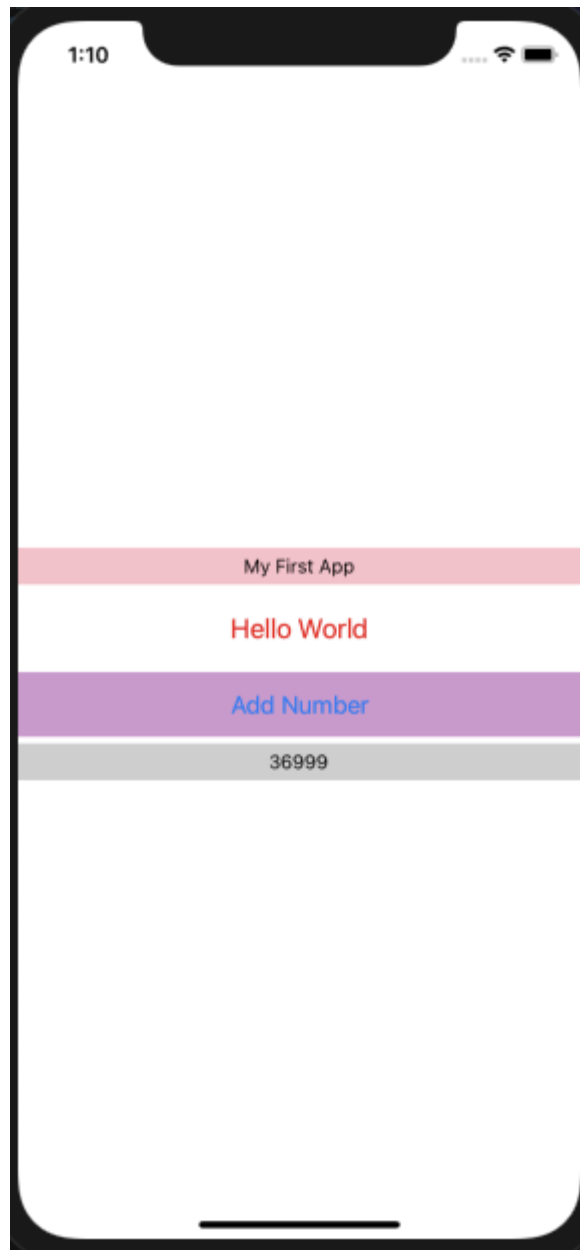
## numlist.js

```

const NumList = (props) => {
  return (
    <View style={styles.numList}>
      <Text>{props.num}</Text>
    </View>
  )
}

```

하지만 이렇게 하면 배열의 모든 요소가 출력이 된다.

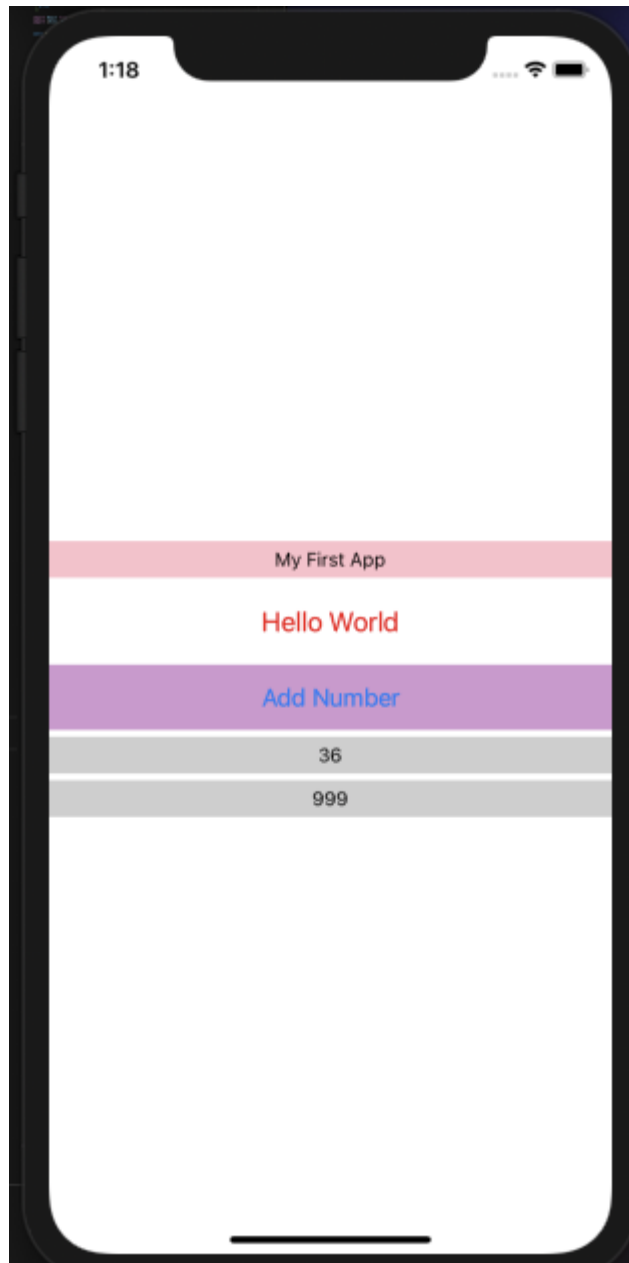


이를 방지하기 위해 배열 객체 내장 함수 map를 사용해야 함

루프문 같은 것. 요소 수만큼 루프를 돌면서 그때의 요소값이 첫번째 인자로 오고 요소의  
인덱스 값이 두번째 인자로옴

```
const NumList = (props) => {  
  return (  
    props.num.map((item, idx)=>(  
      <View style={styles.numList} key={idx}>  
        <Text>{item}</Text>  
      </View>))  
  )  
}
```

```
} )
```



요소에 있는 값들이 각각 출력이 됨.