

React Native Cheatsheet

Creating an app

```
npm react-native init AppName
//or
npm create-react-native-app AppName
```

Components

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

const App = () => {
  return (
    <View style={styles.container}>
      <Text>Hello World</Text>
    </View>
  )
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
  },
});

export default App;
```

View

```
function App() {
  return (
    /* Base layout structure */
    <View style={{ flex: 1 }}>
      /* Simple background color */
      <View style={{ color: 'blue' }}>
        <Text>Text with bg color</Text>
      </View>
      /* Space layout structure */
      <View style={{ margin: 10 }} />
    </View>
  );
}
```

Text

```
function App() {
  return (
    <Text style={{ margin: 10 }}>
      Hello World
    </Text>
  );
}
```

TouchableHighlight / TouchableOpacity

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

const App = () => {
  const [count, setCount] = useState(0);
  const onIncrement =
    () => setCount(count + 1);
  const onDecrement =
    () => setCount(count - 1);

  return (
    <View
      style={{
        flex: 1,
        margin: 20,
        alignItems: 'center',
        justifyContent: 'center',
      }}>
      <Text>Count: {count}</Text>
      <TouchableOpacity
        onPress={onIncrement}
        style={{backgroundColor: '#3F3F3F',
        margin: 20, padding: 20}}>
        <Text>+</Text>
      </TouchableOpacity>
      <TouchableHighlight
        onPress={onDecrement}
        style={{backgroundColor: '#F3F3F3',
        margin: 20, padding: 20}}>
        <Text>-</Text>
      </TouchableHighlight>
    </View>
  );
};

export default App;
```

StyleSheet

```
function App() {
  return (
    <Text style={styles.heading}>
      React Native
    </Text>
    <Text style={styles.message}>
      Hello World
    </Text>
  );
}

const styles = StyleSheet.create({
  heading: {
    fontSize: 16,
  },
  message: {
    fontSize: 11,
    textTransform: 'uppercase',
  },
});
```

Navigation

```
import {View, Text} from 'react-native';
import {NavigatorContainer}
from '@react-navigation/native';
import {createNativeStackNavigator}
from '@react-navigation/native-stack';

function HomeScreen() {
  return (
    <View>
      <Text>Home Screen</Text>
    </View>
  );
}

const Stack =
createNativeStackNavigator();

function App() {
  return (
    <NavigatorContainer>
      <Stack.Navigator>
        <Stack.Screen name="Home"
          component={HomeScreen} />
      </Stack.Navigator>
    </NavigatorContainer>
  );
}

export default App;
```

Vishnu Sivan

Immersive Tech Lead,
Game Developer,
Mobile App Developer,
Full Stack Developer,
Web Developer

Seasoned professional, forward looking
software engineer with 3+ years of
experience in creating and executing
innovative solutions in Immersive field
to enhance business productivity.

<https://codemaker2015.github.io>
<www.linkedin.com/in/codemaker2015>

Running the app

```
cd AppName
npm react-native run-android
//or
npm react-native run-android
```

State and props

```
import {useState} from "react";
import {
  StyleSheet,
  Text,
  View
} from 'react-native';

export default App = (props) => {
  const [count, setCount] = useState(0);

  return (
    <View>
      <Text style={styles.header}>
        {props.title?.toUpperCase()}
      </Text>
      <Text>Count: {count}</Text>
    </View>
  );
}

App.defaultProps = {
  title: 'Hello React Native'
}

const styles = StyleSheet.create({
  header: {
    fontFamily: 'Roboto',
    fontSize: 20,
    fontWeight: 'bold',
    textAlign: 'center',
  },
});
```

Image

```
function App() {
  return (
    <>
      <Image
        source={require('./logo.png')} />
      <Image
        source={{ uri: 'https://...' />
      <Image
        source={{ uri:
          'data:image/png;base64,...'
        }} />
    </>
  );
}
```

TextInput

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

const App = () => {
  const [text, setText] = useState("");
  return (
    <View style={{padding: 15}}>
      <TextInput
        style={{height: 50}}
        placeholder="Enter name"
        onChange={text =>
          setText(text)}
        defaultValue={text}
      />
      <Text style={{padding: 10}}>
        {text}
      </Text>
    </View>
  );
};

export default App;
```

Inline styling

```
function App() {
  const [color, setColor] =
    useState("red");
  return (
    <View>
      <Text style={{
        "border": `1px solid ${color}`
      }}>
        {color}
      </Text>
    </View>
  );
}
```

Detecting Screen Size

```
import {Dimensions}
from 'react-native';

const {width, height} =
Dimensions.get('window');

alert(Screen size: ${width}x${height});
```

useNavigation

```
import * as React from 'react';
import {Button} from 'react-native';
import {useNavigation}
from '@react-navigation/native';

function App() {
  const navigation = useNavigation();

  return (
    <Button
      title="Back"
      onPress={() => {
        navigation.goBack();
      }}
    />
  );
}

export default App;
```

Networking

```
import React, {useEffect, useState} from 'react';
import {ActivityIndicator, FlatList, Text, View} from 'react-native';

function App() {
  const [isLoading, setLoading] = useState(true);
  const [data, setData] = useState([]);

  const getMovies = async () => {
    try {
      const response = await fetch('https://reactnative.dev/movies.json');
      const json = await response.json();
      setData(json.movies);
    } catch (error) {
      console.error(error);
    } finally {
      setLoading(false);
    }
  }

  useEffect(() => {
    getMovies();
  }, []);

  return (
    <View style={{ flex: 1, padding: 20 }}>
      {isLoading ? <ActivityIndicator /> : (
        <FlatList
          data={data}
          keyExtractor={item => item.id}
          renderItem={({ item }) => (
            <Text>{item.title}, {item.releaseYear}</Text>
          )}
        />
      )}
    </View>
  );
}

export default App;
```

Upgrading React Native

```
npm install react-native@latest;
```

useState and useEffect

```
import {useState, useEffect}
from 'react';

function App() {
  const [count, setCount] = useState(0);

  useEffect(() => {
    setTimeout(() => {
      setCount((count) => count + 1);
    }, 1000);
  });

  return <h1>Count: {count}</h1>;
}
```

ScrollView

```
function App() {
  return (
    <ScrollView>
      <Text style={{margin: 10}}>
        Hello World
      </Text>
      <View style={{marginTop: 512}} />
      <Text style={{margin: 10}}>
        Welcome to React Native
      </Text>
    </ScrollView>
  );
}
```

Button

```
function App() {
  return (
    <Button
      onPress={onPressSubmit}
      title="Submit"
      color="#FFFFFF"
    />
  );
}
```

FlatList

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

const App = () => {
  return (
    <View>
      <FlatList
        data={
          [
            {key: 'January'},
            {key: 'February'},
            {key: 'March'},
          ]
        }
        renderItem={({item}) => <Text>{item.key}</Text>
      />
    </View>
  );
};

export default App;
```

Flexbox

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

const App = () => {
  return (
    <View style={styles.container, [
      // Set flexDirection to "column".
      flexDirection: "column"
    ]}>
      <View
        style={{
          flex: 1, backgroundColor: "red"
        }}
      />
      <View
        style={{
          flex: 2, backgroundColor: "orange"
        }}
      />
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 10,
  },
});

export default App;
```

BackHandler android

```
import {useEffect} from 'react';
import {BackHandler} from 'react-native';

function App() {
  useEffect(() => {
    const backAction = () => {
      console.log("back button pressed");
      return false; // back button is enabled
      return true; // back button is disabled
    };

    // Register for hardware back event
    // and attach a handler to it
    const backHandler =
      BackHandler.addEventListener(
        'hardwareBackPress',
        backAction,
      );

    return () => backHandler.remove();
  }, []);
};

export default App;
```