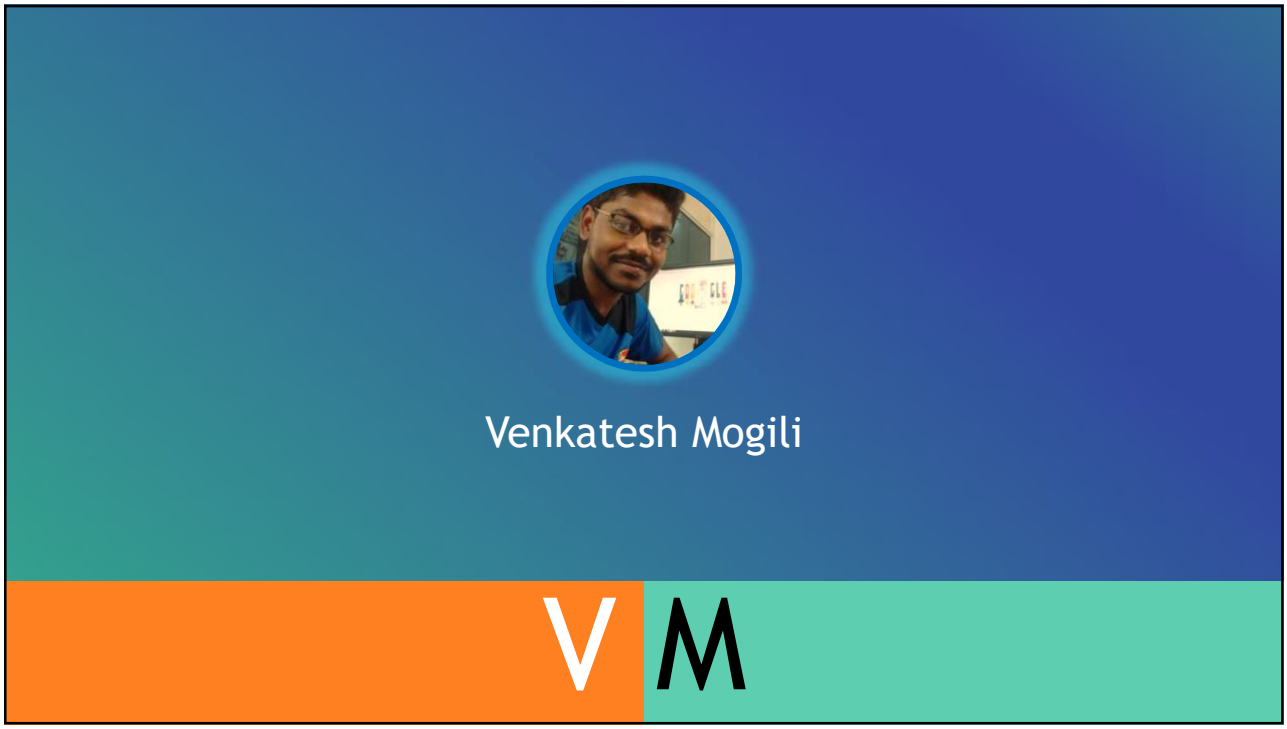

Venkatesh Mogili
V M


Venkatesh Mogili
#WebGuru
తెలుగు లో
React Native
Complete Course
Lecture-2

Contents

- React Fundamentals
- Core Components
- Basic Components
- View
- Text
- Image
- TextInput
- ScrollView
- StyleSheet

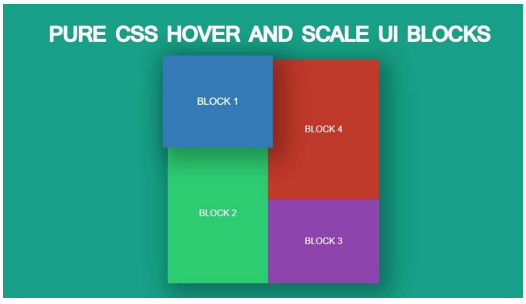
React Fundamentals – Components, JSX

- **Any UI block** can be considered as a **Component.**
- It can be reusable. Create once use anywhere.
- Easy to manage small to complex applications.
- React is everything about components.

```
import React from 'react'

export default function Button() {
    return (
        <div>
            <button>Submit</button>
        </div>
    )
}
```

JSX



React Fundamentals – Props

- Props will be used to pass the data between the components.
- We can pass any valid JavaScript expression as prop.
- Data can be passed in one way only. Unidirectional or top-down/bottom-top.

App Component

```
import React from 'react';
import Button from './components/Button';

function App() {
    return (
        <div>
            <Button title='Click here' />
        </div>
    );
}
export default App;
```

Button Component

```
import React from 'react';

export default function Button(props) {
    return (
        <div>
            <button>{props.title}</button>
        </div>
    );
}
```

Output: Click here

React Fundamentals –State

- State is a local variable for the component.
- Any state in one component won't be accessible to other components, which means by default states are encapsulated in components.

App Component

```
import React, { useState } from 'react';
import Button from './components/Button';

function App() {
    const [ title ] = useState('Click here');
    return (
        <div>
            <button>{title}</button>
            <Button />
        </div>
    );
}
export default App;
```

Output: Click here Submit

Button Component

```
import React, { Component } from 'react';

export default class Button extends Component {
    state = {
        title: 'Submit',
    };
    render() {
        return (
            <div>
                <button>{this.state.title}</button>
            </div>
        );
    }
}
```

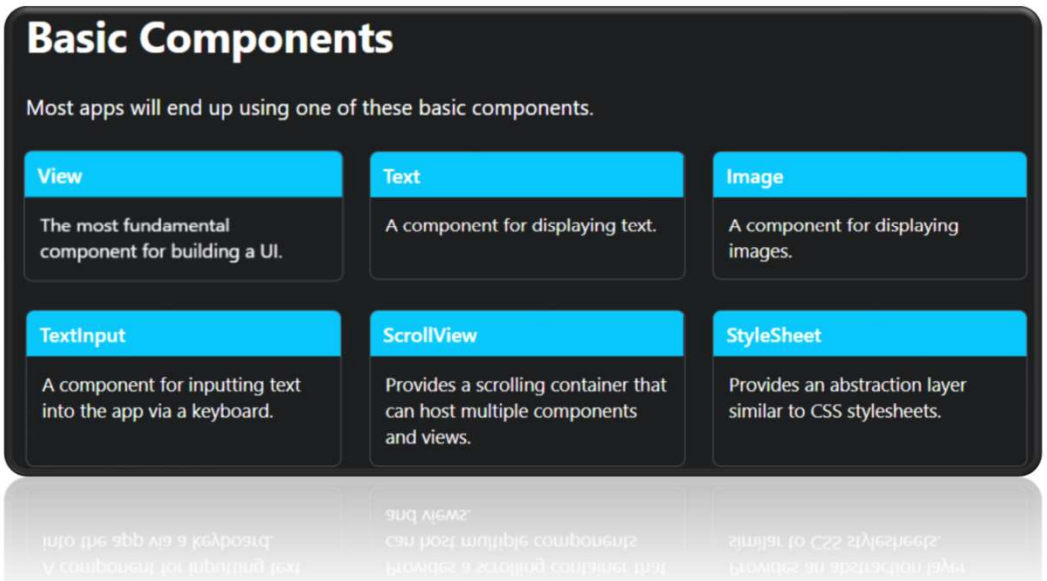
Output: Submit

Core Components

- React Native contains the below core components which are equivalent to the native modules.

1. Basic Components
2. UI
3. List Views
4. Android Components
5. iOS Components
6. Other Components

Basic Components



UI Components

User Interface

These common user interface controls will render on any platform.

Button

A basic button component for handling touches that should render nicely on any platform.

Switch

Renders a boolean input.

List Views

List Views

Unlike the more generic `ScrollView`, the following list view components only render elements that are currently showing on the screen. This makes them a performant choice for displaying long lists of data.

FlatList

A component for rendering performant scrollable lists.

SectionList

Like `FlatList`, but for sectioned lists.

Android Components

Android Components and APIs

Many of the following components provide wrappers for commonly used Android classes.

BackHandler

Detect hardware button presses for back navigation.

DrawerLayoutAndroid

Renders a `DrawerLayout` on Android.

PermissionsAndroid

Provides access to the permissions model introduced in Android M.

ToastAndroid

Create an Android Toast alert.

iOS Components

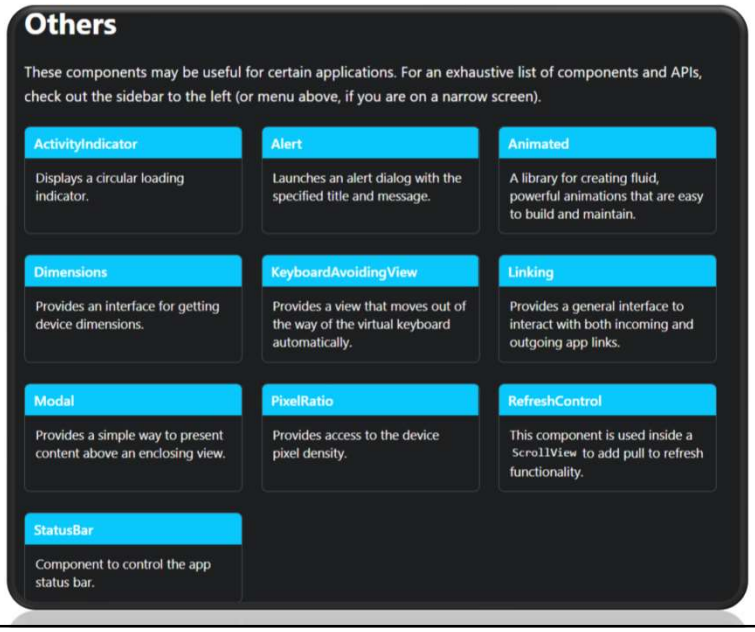
iOS Components and APIs

Many of the following components provide wrappers for commonly used UIKit classes.

ActionSheetIOS

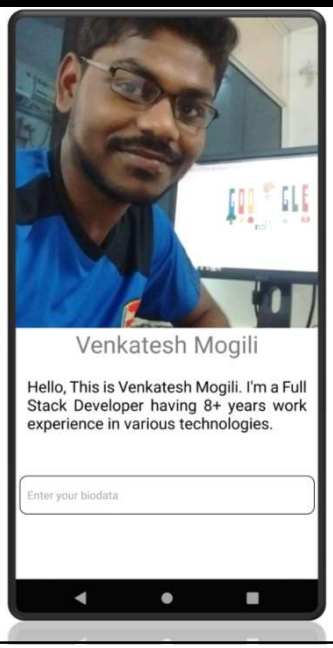
API to display an iOS action sheet or share sheet.

Other Components



Basic Components Practical

- View
- Text
- Image
- TextInput
- ScrollView
- StyleSheet



View

- View is the most fundamental component for building a UI.
- It is a container that supports layout with flexbox, style, some touch handling, and accessibility controls.
- View maps directly to the native view equivalent on whatever platform React Native is running on, whether that is a UIView, , android.view, etc.,
- View is designed to be nested inside other views and can have 0 to many children of any type.

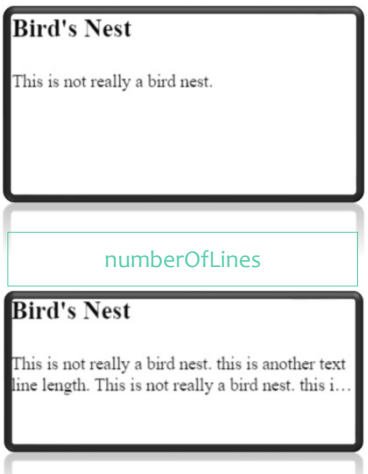
Text

- Text is a React component for displaying text.
- Text supports nesting, styling, and touch handling.

```
<View>
  Some text
</View>
```

```
// GOOD
<View>
  <Text>
    Some text
  </Text>
</View>
```

```
return (
  <Text style={styles.baseText}>
    <Text style={styles.titleText} onPress={onPressTitle}>
      {titleText}
      {"\n"}
      {"\n"}
    </Text>
    <Text numberOfLines={5}>{bodyText}</Text>
  </Text>
);
```



Image

- A React component for displaying different types of images, including network images, static resources, temporary local images, and images from local disk, such as the camera roll.

source

require

uri

blurRadius

fadeDuration

resizeMode

TextInput

- A foundational component for inputting text into the app via a keyboard.
- Props provide configurability for several features, such as auto-correction, auto-capitalization, placeholder text, and different keyboard types, such as a numeric keypad.

Input Text

placeholder

onChangeText

onFocus

onSubmitEditing

Input Text

placeholder

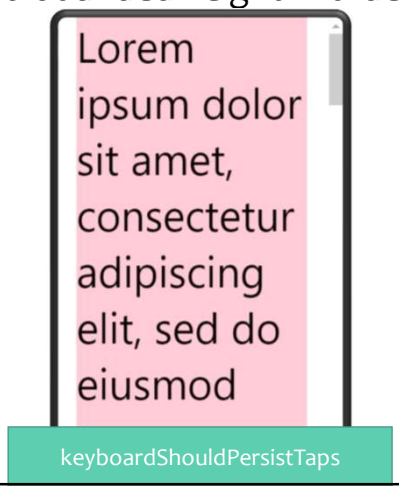
multiline

editable

keyboardType

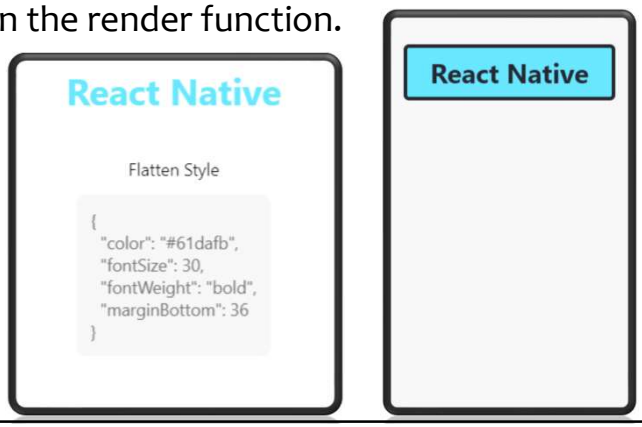
ScrollView

- Component that wraps platform ScrollView while providing integration with touch locking "responder" system.
- ScrollViews must have a bounded height in order to work.



StyleSheet

- A StyleSheet is an abstraction similar to CSS StyleSheets.
- By moving styles away from the render function, you're making the code easier to understand.
- Naming the styles is a good way to add meaning to the low level components in the render function.



Summary

- ✓ React Fundamentals
- ✓ Core Components
- ✓ Basic Components
- ✓ View
- ✓ Text
- ✓ Image
- ✓ TextInput
- ✓ ScrollView
- ✓ StyleSheet

Thank You

VVatch IMplement Try Experiments

VM Online Training