React Primer

Bala Krishna Ragala

The Plan

- What is React
- JSX
- React Components
- Props
- Events
- State
- Lifecycle
- PropTypes
- Default Props

What is React?

- React is an Open Source **view library** created and maintained by Facebook

JSX

- Javascript in XML
- An alternative syntax to create user interfaces declaratively
- Syntactic sugar for React.createElement function

```
React.createElement(component, props, ...children)
```

```
<Text style={{color:'green'}}>Welcome to React Native</Text>
```

```
'use strict';

React.createElement(
  Text,
  { style: { color: 'green' } },
  'Welcome to React Native'
);
```

What’s valid in JSX ??

- Expressions
  - You can embed any JavaScript expression in JSX by wrapping it in curly braces

```
E.g., { ‘React is awesome’ } {3 * 2} {Math.random() * 100} { a ? true : false }
```

- Attributes

```
<div className=’container></div>
```

What’s valid in JSX ?? (contd..)

- Children

```
<ul><li>React</li><li>Angular</li></ul>
```

- Comments

```
{/* this is comment */}
```

It’s all about Components

A new mind set

Component - { Independent, Reuse & Testable }

React is all about building reusable components.

components make code reuse, testing, and separation of concerns easy

Component Thinking

- Look at large monolithic applications as set of reusable types composed to build a larger thing
- React is all about building reusable components

A car

A car – component thinking

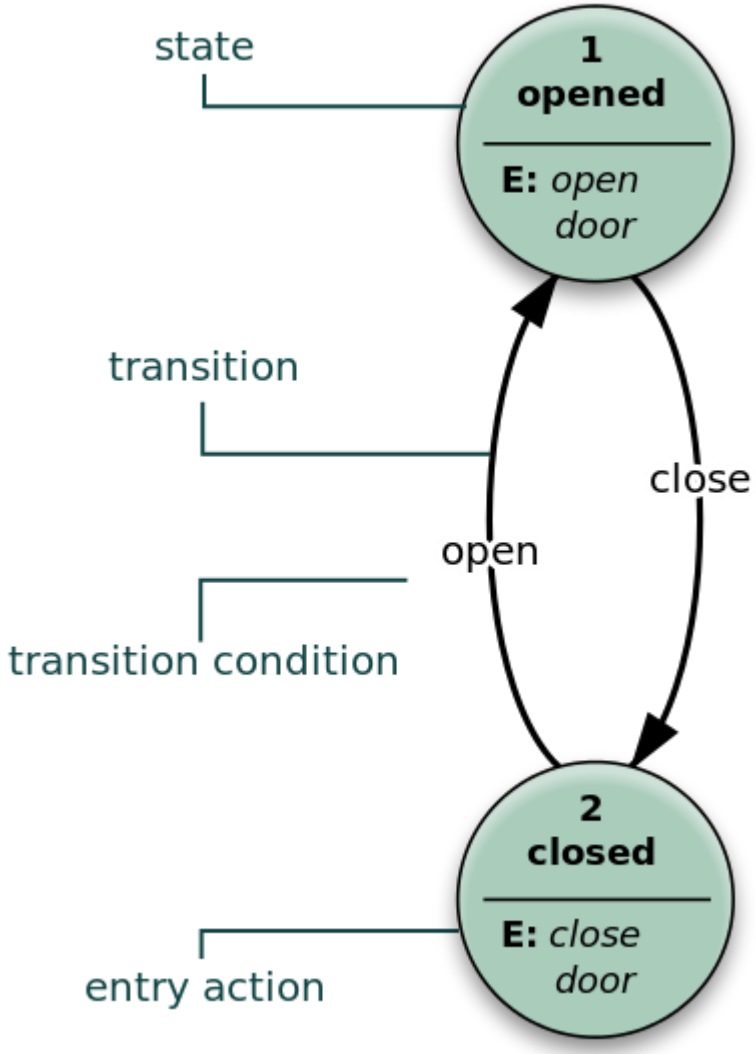
React Component – Class Based

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

class WhyReactNativeIsSoGreat extends Component {
  render() {
    return (
      <View>
        <Text>
          If you like React on the web, you'll like React Native.
        </Text>
        <Text>
          You just use native components like 'View' and 'Text',
          instead of web components like 'div' and 'span'.
        </Text>
      </View>
    );
  }
}
```

React Component - Functional

```
function HelloComponent({name}) {
      return <Text>Hello, {name}</Text>;
}

const HelloComponent = ({name}) => <Text>Hello, {name}</Text>;
```

Guess what?



https://en.wikipedia.org/wiki/Finite-state_machine#/media/File:Finite_state_machine_example_with_comments.svg

State

- UI is all about state, which represents data to be presented on the UI
- React provides this.state object to represent UI data
- Use this.state to get state and this.setState to set state
- React re renders UI for every state change

Props

- Way to pass data to components
- Similar to parameters for a function
- Every react component has this.props automatically created
- You can pass primitives, objects, functions
- Dynamic bag populated by react context
- Use this.props to access the props object

Events

- Events are like bells that notify something has happened
- UI is all about state and actions, actions trigger events
- Handling react events is similar to handling DOM events
- React events are named camelCase instead on lowercase
- Provide callback as handlers to do something for an event

Lifecycle

Mounting

These methods are called when an instance of a component is being created and inserted into the DOM:

- `constructor()`
- `componentWillMount()`
- `render()`
- `componentDidMount()`

Updating

An update can be caused by changes to props or state. These methods are called when a component is being re-rendered:

- `componentWillReceiveProps()`
- `shouldComponentUpdate()`
- `componentWillUpdate()`
- `render()`
- `componentDidUpdate()`

Unmounting

This method is called when a component is being removed from the DOM:

- `componentWillUnmount()`

Prop Types

- Provides type checking support
- https://reactjs.org/docs/typechecking-with-proptypes.html

```
import PropTypes from 'prop-types';

class Greeting extends React.Component {
    render() {
        return (
                <Text>Hello, {this.props.name}</Text>
        );
        }
}

Greeting.propTypes = {
        name: PropTypes.string
};
```

Default Props

- Supply default values for props

```
class Greeting extends React.Component {
    render() {
    return (
        <Text>Hello, {this.props.name}</Text>
    );
    }
}

// Specifies the default values for props:
Greeting.defaultProps = {
        name: 'Stranger'
};
```

Question Time

- In React, component markup is made of HTML and JS expressions?
- Which method of Component is responsible for painting the UI?
- What happens JSX is compiled?
- Does JSX support attributes?
- Component model improves reusability?
- What are the 2 ways to create components in React?
- Functional components are lighter than class based components?
- Is it possible to stop rendering a component?
- Which method do you use to re-render UI?
- Can you validate the prop type of function?