ReactJS

Software System Development – Monsoon 2022

INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY

H Y D E R A B A D

💬 🔁 ♡ 4 ⬆️

Navigation (left sidebar)
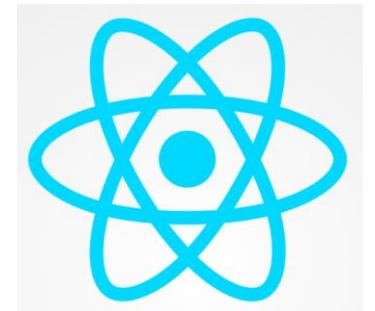
🏠 **Home**

Explore

🔔 Notifications

✉️ Messages

🔖 Bookmarks

📋 Lists

👤 Profile

⋯ More

Tweet

👤 **Sai Anirudh K**
@SaiAnirudhK1 ⋯

Main feed

Manu Awasthi ⋯
@mnwsth · 19h

Graduating from a PhD program in 4 years

💬 9   🔁 11   ♡ 304   ⬆️

♡ **Palki Sharma liked**

Francis Abban ✓ ⋯
@francis_abban · 10h

Am I the only one who misses @palkisu on @WIONews these days? When is she coming back on Gravitas Plus?

💬 23   🔁 28   ♡ 462   ⬆️

Deepak Sen ⋯

Right sidebar

🔍 Search Twitter

News · Last night
A video clip from 1901 has been misattributed to Queen Elizabeth II, fact-checkers report

#OnePlusNord
Celebrate Your Way with Nord | Buy now
↗ Promoted by OnePlus India

Trending in India   ⋯
#HardikPandya
Trending with #ViratKohli, #SuryakumarYadav

Trending   ⋯
Impeach
17.4K Tweets

War in Ukraine · LIVE
Latest updates on the war in Ukraine

Show more

Who to follow

🟢 **aramco** ✓   **Follow**
@aramco
↗ Promoted

Kashmiri Hindu   **Follow**
@BattaKashmiri

Maria Roussou   **Follow**
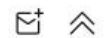@mroussou

Show more

Terms of Service   Privacy Policy
Accessibility   Ads info   More ⋯

Messages   ✉️ ⌃

Class Tweet {
        state = { } ;
        render() {   ⟶   React Element
        }
}

Node package manager

~sudo npm -i -g create-react-app@1.5.2

React Element   State   DOM Element
                       Change

Visual Studio Code Plugin

ReactSimpleSnippets – Bruke Holland
Prettier – Code formatter – Esben Peterson
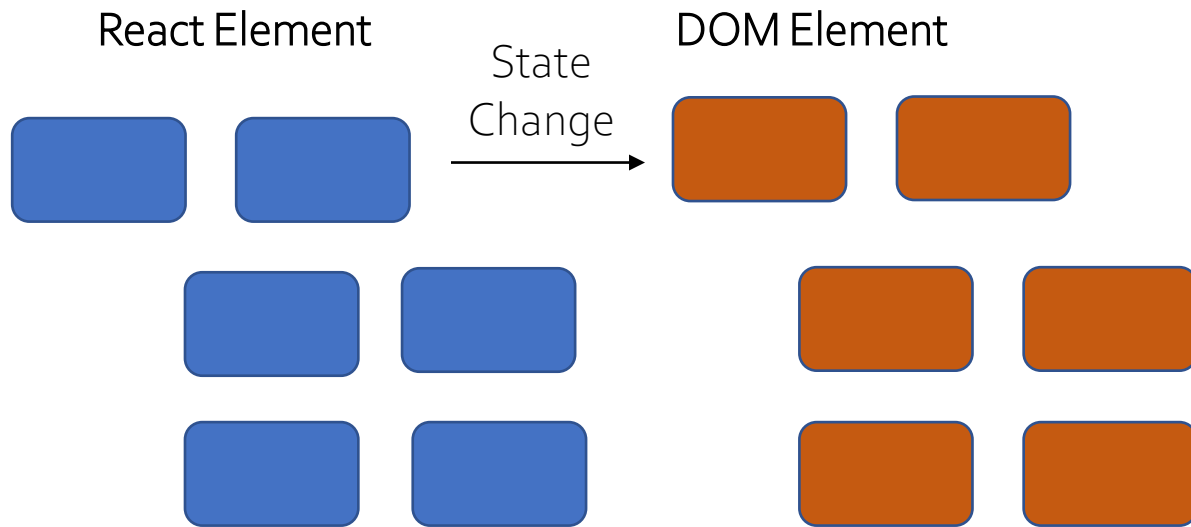
const element = document.querySelector();
element.classList.add();
element.addEventListener ();
//React to update the State Change of DOM rather updating DOM

```
<!doctype html>
<html>
<head>
<title>SSD Example</title>
</head>
<body>
<div id="reactapp"></div>
<script src="./webpackOutput/reactApp.bundle.js"></script>
</body>
</html>
```
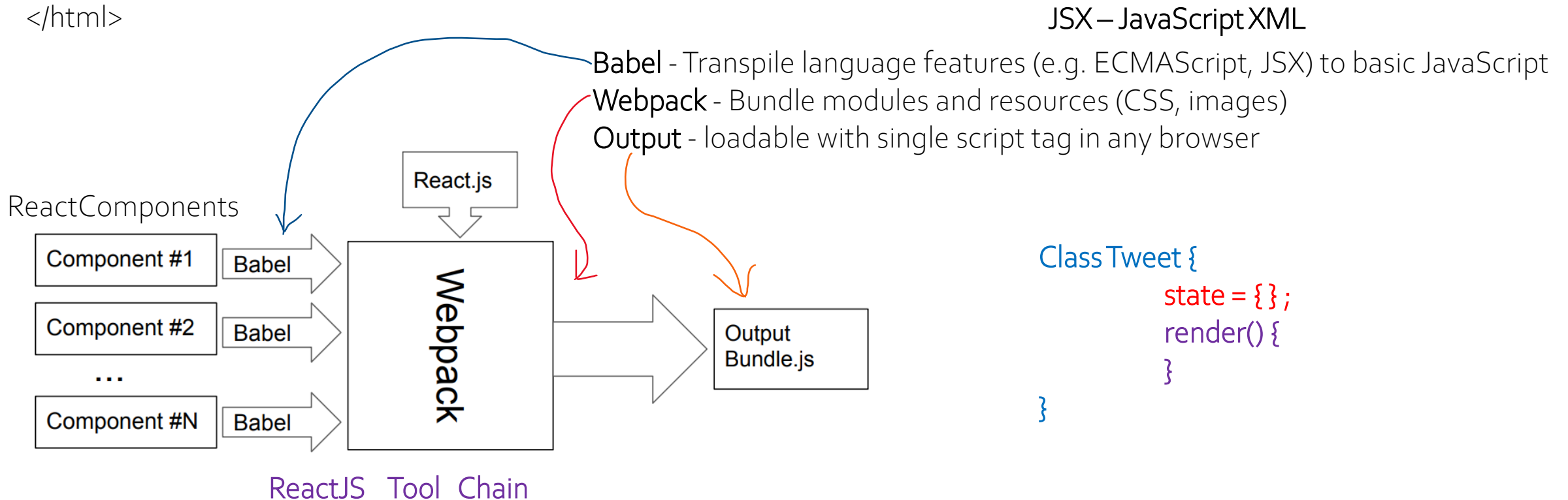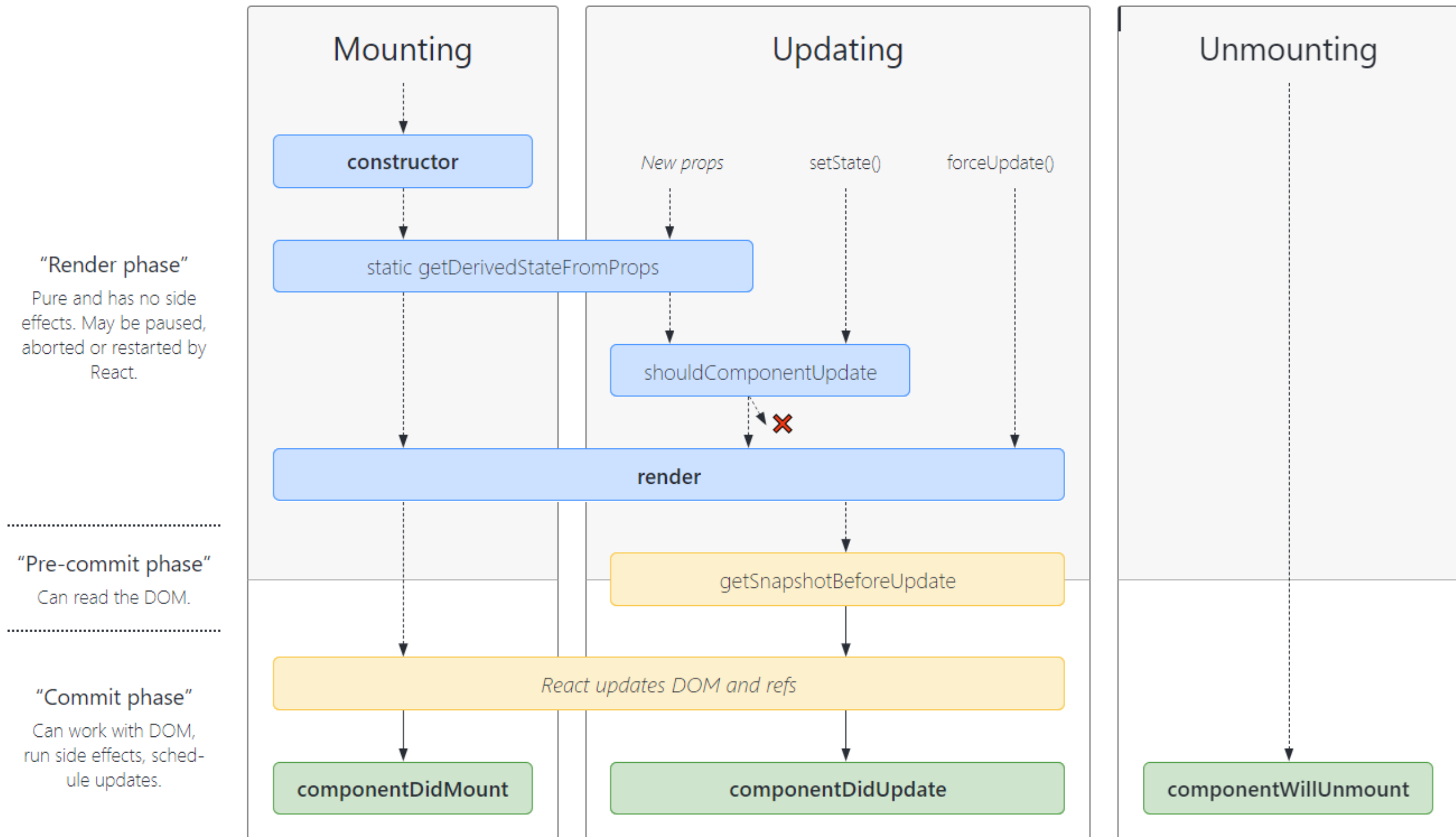
ReactJS applications come as a JavaScript blob that will use the DOM interface to write the view into the div.

JSX – JavaScript XML

Babel - Transpile language features (e.g. ECMAScript, JSX) to basic JavaScript
Webpack - Bundle modules and resources (CSS, images)
Output - loadable with single script tag in any browser

React.js

ReactComponents

Component #1	Babel
Component #2	Babel
...	
Component #N	Babel

Webpack

Output Bundle.js

ReactJS Tool Chain

Class Tweet {
    state = { } ;
    render() {
    }
}

Mounting

constructor

↓

static getDerivedStateFromProps

↓

render

↓

componentDidMount

Updating

New props   setState()   forceUpdate()

static getDerivedStateFromProps

↓

shouldComponentUpdate

✕

render

↓

getSnapshotBeforeUpdate

↓

React updates DOM and refs

↓

componentDidUpdate

Unmounting

componentWillUnmount

"Render phase"

Pure and has no side effects. May be paused, aborted or restarted by React.

"Pre-commit phase"

Can read the DOM.

"Commit phase"

Can work with DOM, run side effects, schedule updates.

Component

Components are self-contained reusable building blocks of web application.

React components are basically just idempotent functions (same input produces same output) i.e. $F(f(x)) = f(x)$

They describe your UI at any point in time, just like a server-rendered app.

Created using React.createClass()
The only required method is render()
Inserted into DOM
using React.renderComponent()

```
var React = require('react'),
    SimpleView = React.createClass({
        render: function () {
            return <h1><strong>Example
1:</strong> A simple component</h1>;
        }
    });

React.renderComponent(SimpleView(
),
document.getElementById('example'))
;
```

Props

Passed down to component from parent component and represents data for the component. It is accessed via this.props

```
render: function() {

    var someProp = 'bar';
    console.log('component render()', this.props);


    return <div>
        <AnotherComponent foo={someProp} model={this.props.model} />
    </div>;
}
```

State

Represents internal state of the component.

Accessed via this.state

When a component's state data changes, the rendered markup will be updated by re-invoking render() method

```
render: function() {
    return <h3>Click count:
        <span className='label label-default'>{this.state.clicks}</span>
    </h3>;
}
```

JSX

XML-like syntax for generating component's HTML

Easier to read and understand large DOM trees

Translates to plain JavaScript using react-tools

```
/** @jsx React.DOM */

render: function () {
    return <div>
        <h2>
            <strong>Example 4:</strong>  React App
        </h2>
    </div>;
}


/** regular DOM */

render: function () {
    return React.DOM.div(null,
        React.DOM.h2(null, React.DOM.strong(null, "Example 4:"), " React App")
    );
}
```

Virtual DOM

- The virtual DOM is used for efficient re-rendering of the DOM

- React aims to re-render the virtual tree only when the state changes

- Uses 2 virtual trees (new and previous) to find differences and batch update real DOM

- Observes data changes (setState) and does dirty-checking to know when to re-render component

- Whenever possible, does not update entire component in real DOM - only computes a patch operation that updates part of the DOM