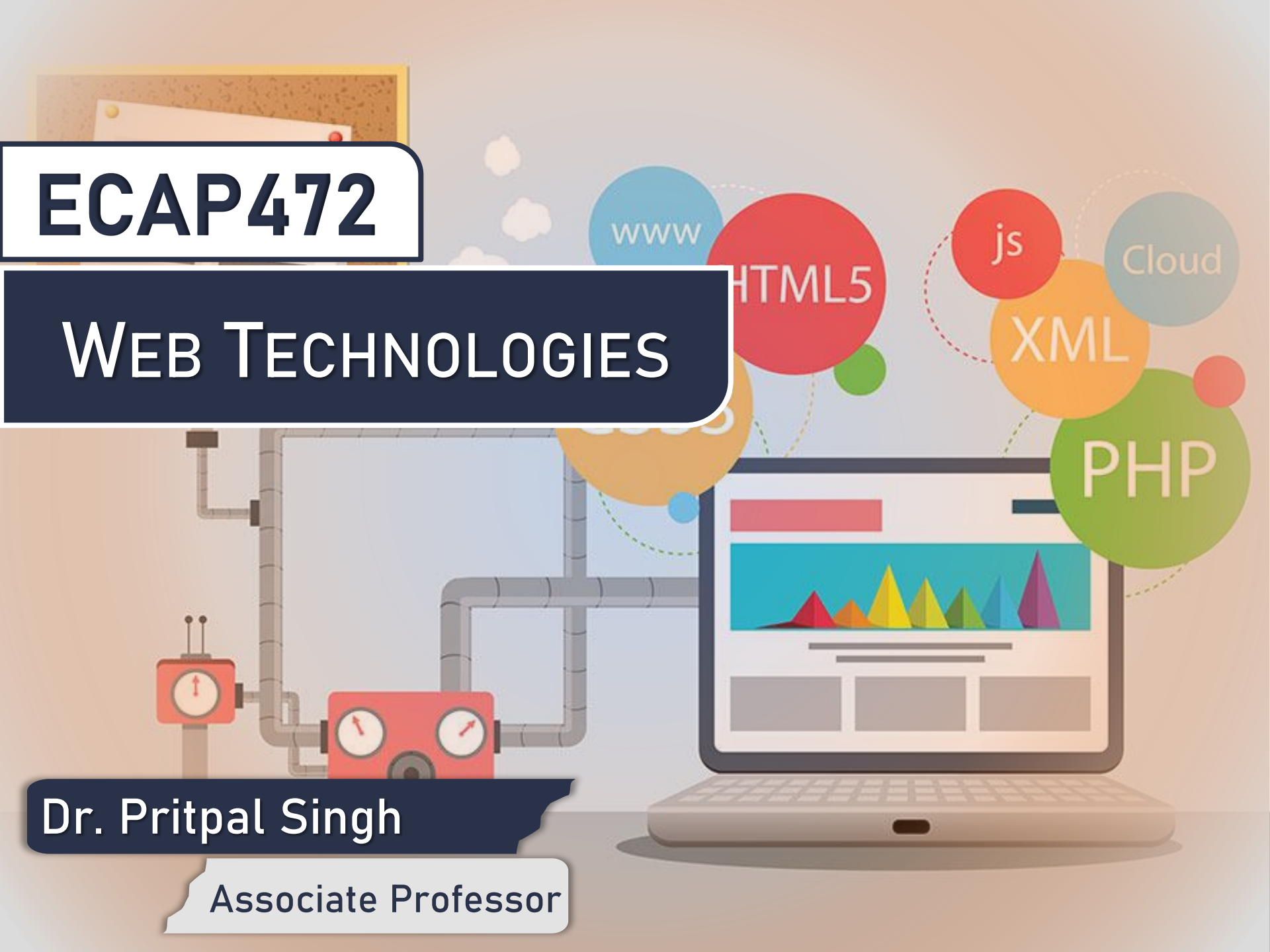
ECAP472

Web Technologies



Dr. Pritpal Singh

Associate Professor

Learning Outcomes

After this lecture, you will be able to

- Understand concept of React event management .
- Understand concept of React Conditional statements.

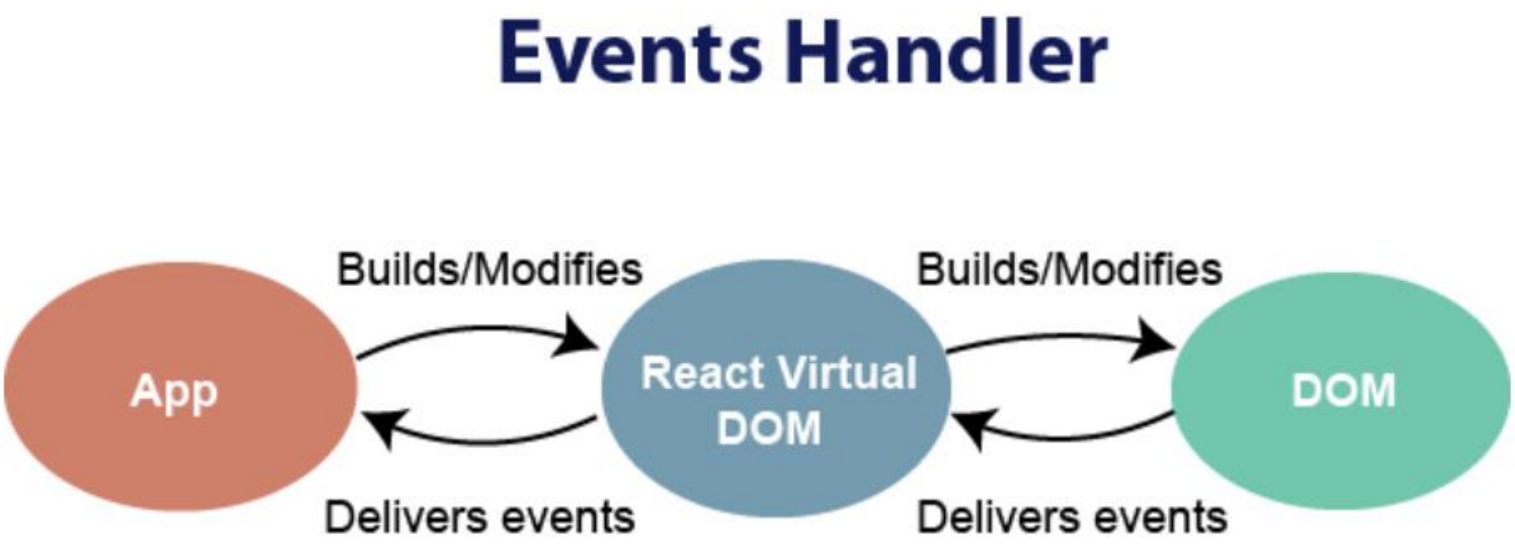
React Events

- An event is an action that could be triggered as a result of the user action or system generated event. For example, a mouse click, loading of a web page, pressing a key, window resizes, and other interactions are called events.

React Events

- React has its own event handling system which is very similar to handling events on DOM elements. The react event handling system is known as Synthetic Events. The synthetic event is a cross-browser wrapper of the browser's native event.

Event Handler



Handling Events

Handling events with react have some differences from handling events on DOM. These are:

- React events are named as camelCase instead of lowercase.
- With JSX, a function is passed as the event handler instead of a string

Event declaration in plain HTML:

```
<button onclick="showMessage()">

    Hello JavaTpoint

</button>
```

Event declaration in React:

```
<button onClick={showMessage}>

    Hello JavaTpoint

</button>
```

Adding Events

- React events are written in camelCase syntax:
- onClick instead of onclick.
- React event handlers are written inside curly braces:
- onClick={shoot} instead of onClick="shoot()".

Example:

Put the shoot function inside the Football component:

```
function Football()
 {
  const shoot = () => {
    alert("Great Shot!");
  }
return (
    <button onClick={shoot}>Take the shot!</button>
  );
}
ReactDOM.render(<Football />, document.getElementById('root'));
```

Passing Arguments

- To pass an argument to an event handler, use an arrow function.
- Send "Goal!" as a parameter to the shoot function, using arrow function:

```
function Football() {
  const shoot = (a) => {
    alert(a);
  }
  return (
    <button onClick={() => shoot("Goal!")}>Take the shot!</button>
  );
}

ReactDOM.render(<Football />, document.getElementById('root'));
```

React Conditional Rendering

- In React, you can conditionally render components.
- There are several ways to do this like:
- Using if Statement
- We can use the if JavaScript operator to decide which component to render.

Using an if…else Statement

- An if…else statement will execute the actions contained in the if block when the condition is satisfied. Otherwise, it will execute the actions contained in the else block.
- In JSX, you are able to use JavaScript code with markup to render dynamic values within your application. JSX uses curly braces ({ and }) to signify expressions that need to be interpreted prior to rendering. The caveat, however, is that there is a limit to what can be done within such braces.

Using a switch Statement

- As shown previously, you can conditionally return different markup from a component based on set conditions using an if…else statement. The same could be achieved with a switch statement where you can specify the markup for various conditions.

Using Element Variables

- Element variables are similar to the approach to extract the conditional rendering into a function. Element variables are variables that hold JSX elements. You can conditionally assign elements or components to these variables outside the JSX and only render the variable within JSX.

Using Ternary Operators

- The conditional (ternary) operator is the only JavaScript operator that takes three operands. This operator is frequently used as a shortcut for the if statement

Using Logical && (Short Circuit Evaluation)

- Short circuit evaluation is a technique used to ensure that there are no side effects during the evaluation of operands in an expression. The logical && helps you specify that an action should be taken only on one condition, otherwise, it would be ignored entirely.

Recommendations:

Generally, keep in mind the following recommendations:

- For situations where there is only one expected outcome, the "short circuit evaluation" is possibly most applicable.
- For situations where there are two expected outcomes, an if…else statement, element variable, ternary operator, or "immediately invoked function expression" is probably most applicable.
- For situations where there are more than two outcomes, a switch statement, extracted function, or extracted functional component is probably most applicable.

Example:

We'll use these two components:

```
function MissedGoal()
{
  return <h1>MISSED!</h1>;
}
function MadeGoal()
{
  return <h1>Goal!</h1>;
}
```

Example:

Now, we'll create another component that chooses which component to render based on a condition.

Component based on condition

```
function Goal(props)
{
  const isGoal = props.isGoal;
  if (isGoal) {
    return <MadeGoal/>;
  }
  return <MissedGoal/>;
}
ReactDOM.render(
  <Goal isGoal={false} />,
  document.getElementById('root')
);
```

Ternary Operator

Return the MadeGoal component if isGoal is true, otherwise return the MissedGoal component:

```
function Goal(props) {
  const isGoal = props.isGoal;
  return (
    <>
      { isGoal ? <MadeGoal/> :
<MissedGoal/> }
    </>
  );
}
ReactDOM.render(
  <Goal isGoal={false} />,
  document.getElementById('root')
);
```

Angular vs React 2022 : Which JS Framework your Project Requires?

What is Angular?

- Angular is a development platform built on the typescript. It is a component-based framework for building scalable web apps. It has a collection of well-integrated libraries and features such as client-server communication, routing, and more. It has a suite of developer tools to develop and scale projects from single-developer size to enterprise-grade applications. Moreover, it's constantly updated technology with its latest developments led by the Angular team at Google

What is React?

- React is an open-source JavaScript library used for frontend development. It is used for building user interface or UI components. Its component-based and declarative traits let developers create interactive and complex UIs easily. Developers can build fast and scalable apps for all platforms due to its “learn once write anywhere” principle. React is managed by Facebook and a community of individual developers and communities.

Angular vs. React Comparison

- Angular is a Javascript framework built using Typescript, while Reactjs is a Javascript library and built using JSX.
- Angular is mostly used to build complex enterprise-grade apps like single-page apps and progressive web apps, while React is used to build UI components in any app with frequently variable data

Angular vs. React Comparison

- Angular's learning curve is steeper due to its too many in-built functionalities while React's smaller package size.

Angular vs. React Comparison

	Angular	ReactJs
Founded	Founded by Misko Hevery	Founded by Jordan Walke
Release Year	2009	2013
Ideal For	Creating highly active and interactive web applications	Large web applications with frequently variable data
DOM	Real	Virtual
App Size	Relatively Small	Relatively Small

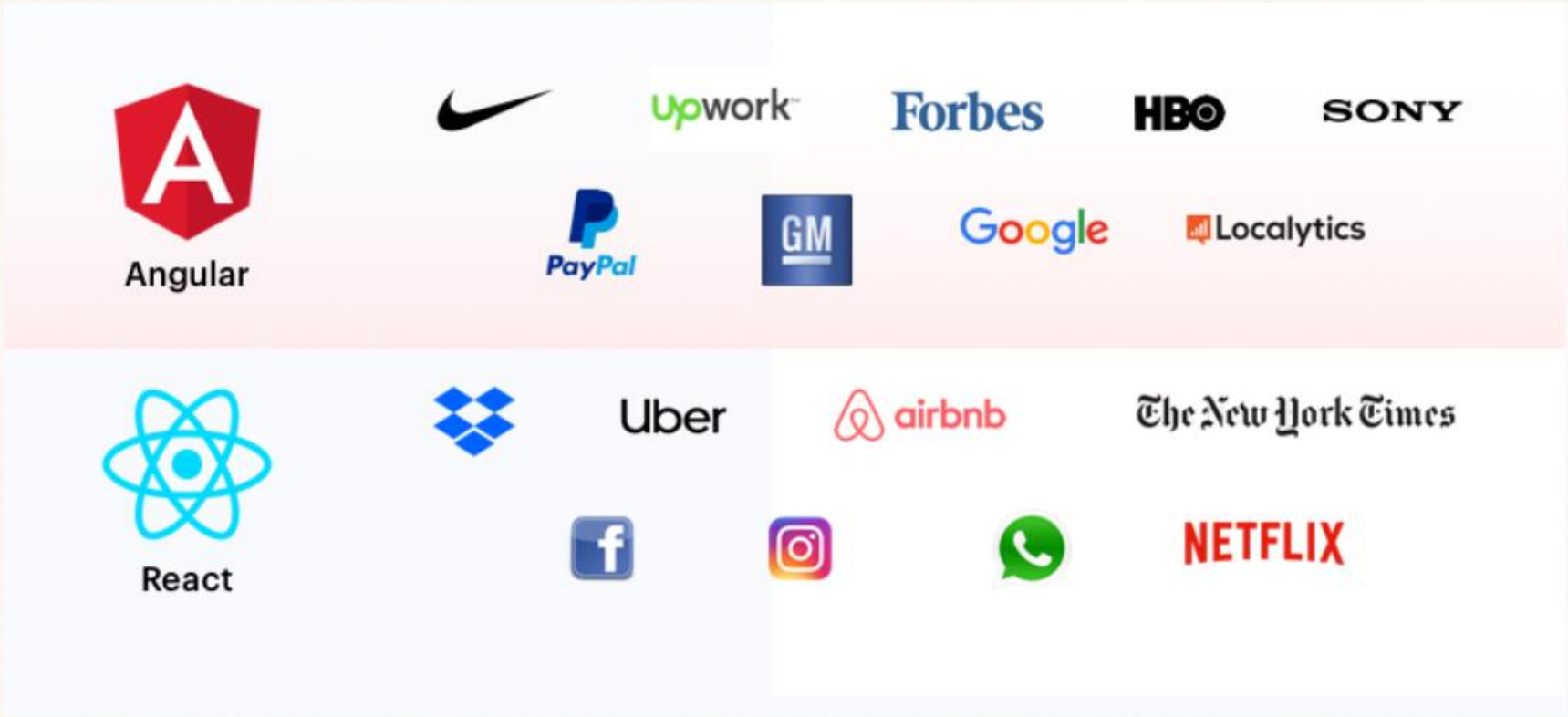
Angular vs. React Comparison

Performance	High	High
Dynamic UI Binding	UI binding at plain object or property level	Direct linking of states to the UI
Data Binding	Two-way	One-way
Learning Curve	Steep	Moderate
GitHub Stars	41,871	113,719
Opinionation	Considerably less opinionated	Flexible opinionation

Angular vs. React Comparison

UI Rendering	Client/Server side	Client/Server side
Price	Open source	Open source
What Should I Choose?	• TypeScript • Huge community support • Your app is really large? • Clean HTML? • Object-oriented-programming (OOP)	• Flexibility • Big ecosystems • If you love Javascript • Small team • Good at choosing among the best options (packages)

Angular and Reactjs Use Cases



What is Angular Good for?

- Angular speeds up the frontend development and is known as the standalone solution for building quickest data-driven web applications. Since it is built up by Google engineers, you can be sure of its reliability and efficiency of code backed up by the massive community of experts. Angular is known for its model-view-controller capability that augments the functionalities of browser-based applications by reducing the javascript code needed to keep the application functional and robust.

What is Reactjs Good for?

- Reactjs is an open-source library. It is used for handling the view layer of the application, and developers prefer it to build reusable UI components. It is fast and scalable to change the data without reloading the page. Reactjs was first used on Facebook and later was implemented in the Instagram news feed. Mostly the large scale applications are using the Reactjs for loading the data without refreshing the pages.

That’s all for now…