PES UNIVERSITY

WEB TECHNOLOGIES

React JS – Complex Components

Prof. Vinay Joshi and Dr. Sarasvathi V
Department of Computer Science and Engineering

Acknowledgement
The slides are created from various internet resources with valuable contributions from multiple professors and teaching assistants in the university.

React.JS – Complex Components

Agenda

- Introduction
- Approach followed
- Complex Component creation
- Sub component creation
- Properties from parent to sub component
- Modified parent and sub components
- Transferring properties

React.JS – Complex Components

Introduction

- Combining components to create the complex one
- Advantage is **composability**
- Approach
    - Identify the major visual elements
    - Breaking them into individual components

React.JS – Components and Properties

Transferring Properties

- Sending properties down the structural tree while working with multiple components in a hierarchy

- **transferPropsTo():**
  - Passes all the properties of a parent component to a child component unless we explicitly set the value of a child component's property

- Coding examples

React.JS – Components and Properties

Transferring Properties

In modern React (including React 17 and 18), the **transferPropsTo** method is outdated and has been replaced by a more standardized approach to passing props through components. **Instead of transferPropsTo, you should use explicit prop passing.**

```
//Parent Component
import React from 'react';
import ChildComponent from './ChildComponent'; // Assume ChildComponent is in the same directory

class ParentComponent extends React.Component {
  render() {
    return (
      <div>
        <ChildComponent name="React" age={8} />
      </div>
    );
  }
}

export default ParentComponent;
```

React.JS – Components and Properties

Transferring Properties

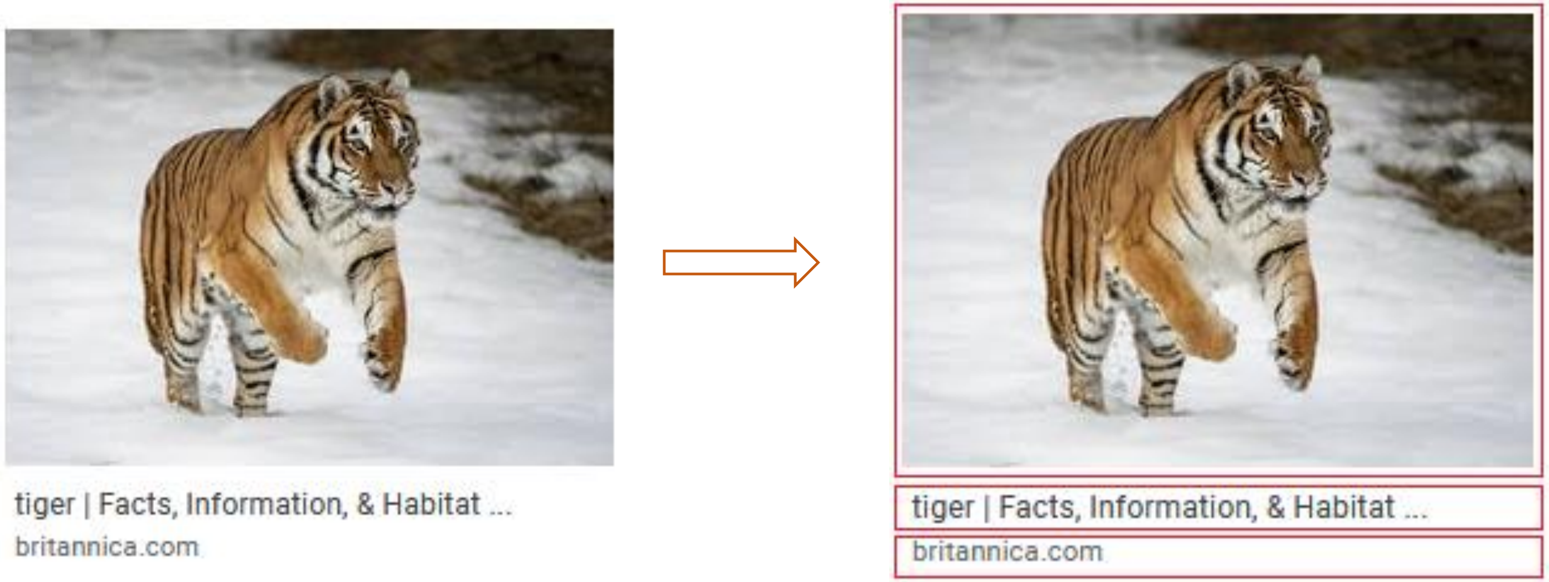
```
//Child Component
import React from 'react';

class ChildComponent extends React.Component {
  render() {
    return (
      <div>
        <h1>Name: {this.props.name}</h1>
        <h2>Age: {this.props.age}</h2>
      </div>
    );
  }
}

export default ChildComponent;
```

React.JS – Complex Components

Approach followed



tiger | Facts, Information, & Habitat ...

britannica.com

tiger | Facts, Information, & Habitat ...

britannica.com

- The complex component consists of **Image rendering, caption rendering and link rendering** on the web page

React.JS – Complex Components

Creation of complex component

```
class SrchResult extends React.Component
{
    render() {
        return (
            <div>
                <ResImage/>
                <ResCaption/>
                <ResLink/>
            </div>
        );
    }
}
```

class component names should be capitalized, and methods like render() should be correctly defined

React 18 still supports class components, though functional components with hooks are more commonly used in recent versions.

React.JS – Complex Components

Creation of sub components

```
class ResImage extends React.Component
{
 render() {
  return (
   <div>
    <img src="tiger.jpg" alt="Tiger" />
   </div>
  );
 }
}
```

In JSX, the <img> tag should be self-closing (<img />) because it doesn't have any child elements.

```
class ResLink extends React.Component {
      render() {
            return (
                  <div> <a
                  href="https://www.britannica.com/animal/tiger">
                  britannica.com </a> </div> );
      }
}
```

```
class ResCaption extends React.Component {
      render() {
            return (
                  <div> <p>tiger | Facts, Information and
                  Habitat…</p> </div> );
      }
}
```

React.JS – Complex Components

Properties from parent to sub component

To avoid hard coding, properties have to be passed from parent Components to its sub components

- src
- href
- linktext
- caption

```
<SrchResult
  src="tiger.jpg"
  href="https://www.britannica.com/animal/tiger"
  linktext="britannica.com"
  caption="tiger | Facts, Information and Habitat..."
/>
```

React.JS – Complex Components

Modified Parent and sub components

```
class SrchResult extends React.Component {
 render() {
  return (
   <div>
    <ResImage src={this.props.src} />
    <ResCaption caption={this.props.caption} />
    <ResLink               href={this.props.href}
linktext={this.props.linktext} />
   </div>
  );
 }
}
```

```
class ResImage extends React.Component {
 render() {
  return (
   <div>
    <img src={this.props.src} alt="Tiger" />
   </div>
  );
 }
}
```

React.JS – Complex Components

Modified Parent and sub components

```
class ResCaption extends React.Component {
  render() {
    return (
      <div>
        <p>{this.props.caption}</p>
      </div>
    );
  }
}
```

```
class ResLink extends React.Component {
  render() {
    return (
      <div>
        <a href={this.props.href}>{this.props.linktext}</a>
      </div>
    );
  }
}
```

We are passing src, href, linktext and caption as props from the parent (SrchResult) to the sub-components (ResImage, ResCaption, ResLink).

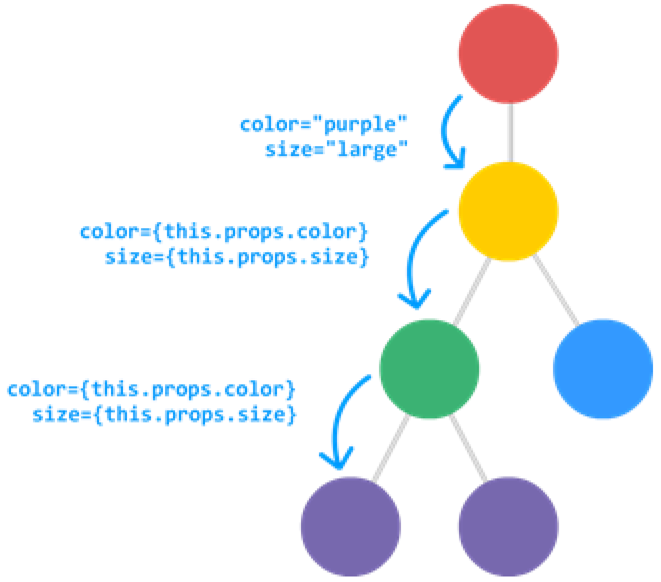
Inside class components, use **this.props** to access the properties passed from the parent.

React.JS – Complex Components

Transferring Properties

- Transferring properties was not a tedious as there was only one level of hierarchy
- If there are multiple levels of components, transferring properties will be very cumbersome
- To overcome that, we use the **spread operator (…)**

<Display {...this.props}/>



React.JS – Complex Components

Transferring Properties - Modified Parent and sub components

```
class SrchResult extends React.Component {
  render() {
    return (
      <div>
        <ResImage {...this.props} />
        <ResCaption {...this.props} />
        <ResLink {...this.props} />
      </div>
    );
  }
}
```

```
class ResImage extends React.Component {
  render() {
    return (
      <div>
        <img src={this.props.src} alt="Tiger" />
      </div>
    );
  }
}
```

Spread Operator: The {...this.props} pattern passes all props from SrchResult to its child components (ResImage, ResCaption, ResLink), avoiding the need to manually pass each prop.

Simplified Prop Transfer: This reduces redundancy and complexity, especially when there are many props or multiple levels of components.

React.JS – Complex Components

Transferring Properties - Modified Parent and sub components

```
class ResCaption extends React.Component {
  render() {
    return (
      <div>
        <p>{this.props.caption}</p>
      </div>
    );
  }
}
```

```
class ResLink extends React.Component {
  render() {
    return (
      <div>
        <a href={this.props.href}>{this.props.linktext}</a>
      </div>
    );
  }
}
```

We are passing src, href, linktext and caption as props from the parent (SrchResult) to the sub-components (ResImage, ResCaption, ResLink).

Inside class components, use **this.props** to access the properties passed from the parent.

To streamline passing props down through multiple levels of components, the spread operator (...) can be used to automatically pass all the props from the parent to the child components.

THANK YOU

Vinay Joshi and Dr.Sarasvathi V

Department of Computer Science and Engineering

vinayj@pes.edu

sarsvathiv@pes.edu

Acknowledgement

The slides are created from various internet resources with valuable contributions from multiple professors and teaching assistants in the university.