BASICS OF REACT COMPONENTS

Prof. Andrew Sheehan

Boston University/MET
Computer Science Department

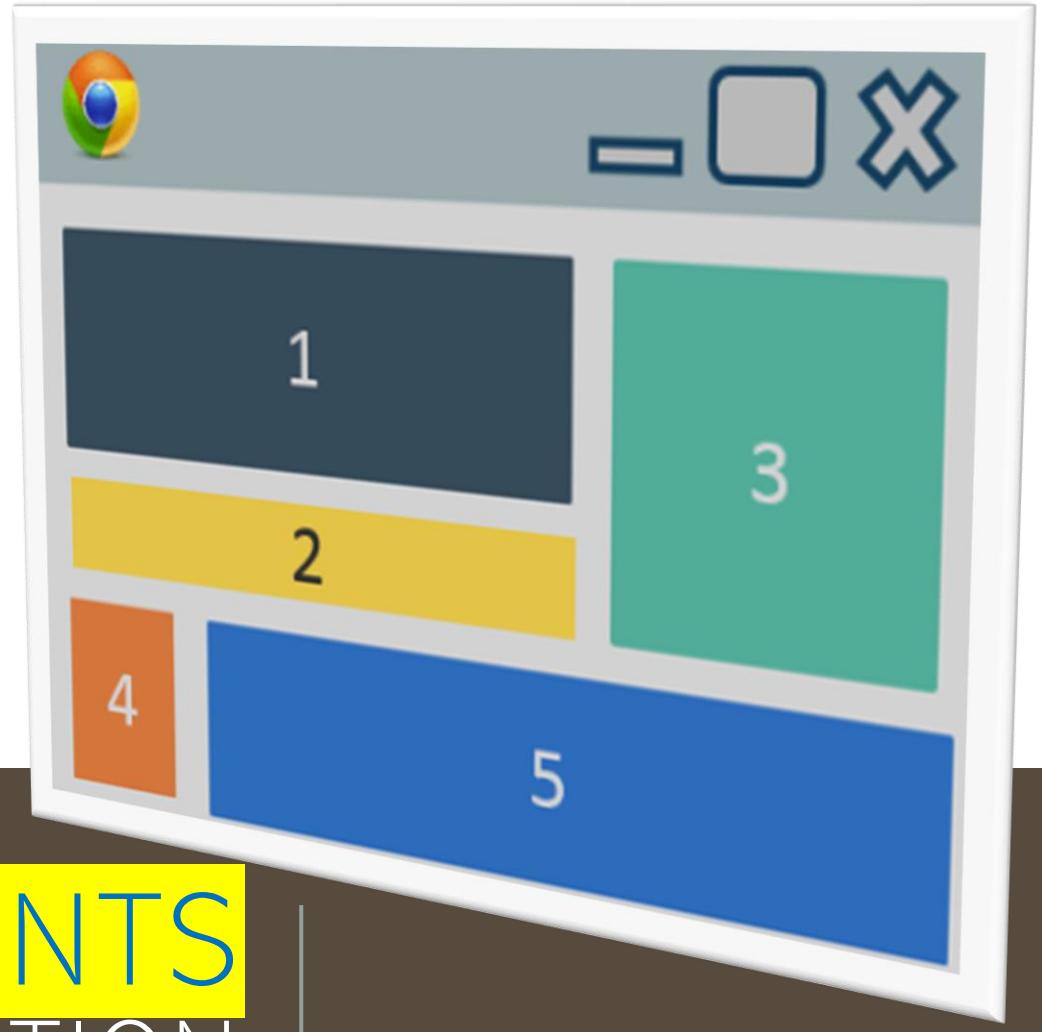
WHAT ARE COMPONENTS

Components independent, reusable pieces of code.

Components are the foundation of your application.

Can be just a button or have more functionality.

Allows you to separate the user interface into functional parts.



COMPONENTS COMPRISE THE APPLICATION

```
function OnOffComponent() {    // OnOffComponent.js
    return (
        <>
          <strong>Activate your profile?</strong>
          <input type="radio" name="activated" value="on"> Yes <br>
          <input type="radio" name="activated" value="off"> No <br>
        </>
    );
}
function ProfileComponent() {    // ProfileComponent.js
    return (
       <>
        <h1>Profile Settings:</h1>
        <OnOffComponent/>
       </>
   );
}
```

TYPES OF COMPONENTS

1. Class (Legacy)
2. Function

```
function Greetings() {
    return (
        <h1>Hi There!</h1>;
    );
}
```

```
let Welcome = (name = 'Guest') => {
    return (
        <h1>Welcome, {name}!</h1>;
    );
}
```

FUNCTION-BASED COMPONENTS

```
class HiThere extends React.Component {
  render() {
    return (
      <h1>Hi</h1>
    );
  }
}
```

CLASS COMPONENT (LEGACY)

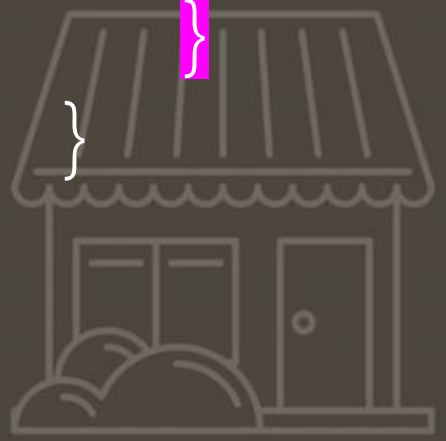
Note: Always start component names with a capital letter.

React treats components starting with lowercase letters as DOM tags. For example, `<div />` represents an HTML div tag, but `<Welcome />` represents a component and requires `Welcome` to be in scope.

MUST USE CAPITAL LETTERS

```
function Outer()
{
  return ( <h1>Hello</h1> );

    function Inner() {
        return ( <h2>There</h2> );
    }
}
```




NEVER NEST COMPONENTS

IMPORTING & EXPORTING

```
export function Component()  {
}

// using it in another component
import { Component } from
      "./components/Component.js";
```

Fragments allows you to group elements under a parent node, without having to add more HTML.

```
<Fragment>
   <span> {temp} </span>
</Fragment>
```

```
<>
   <span> {temp} </span>
</>
```

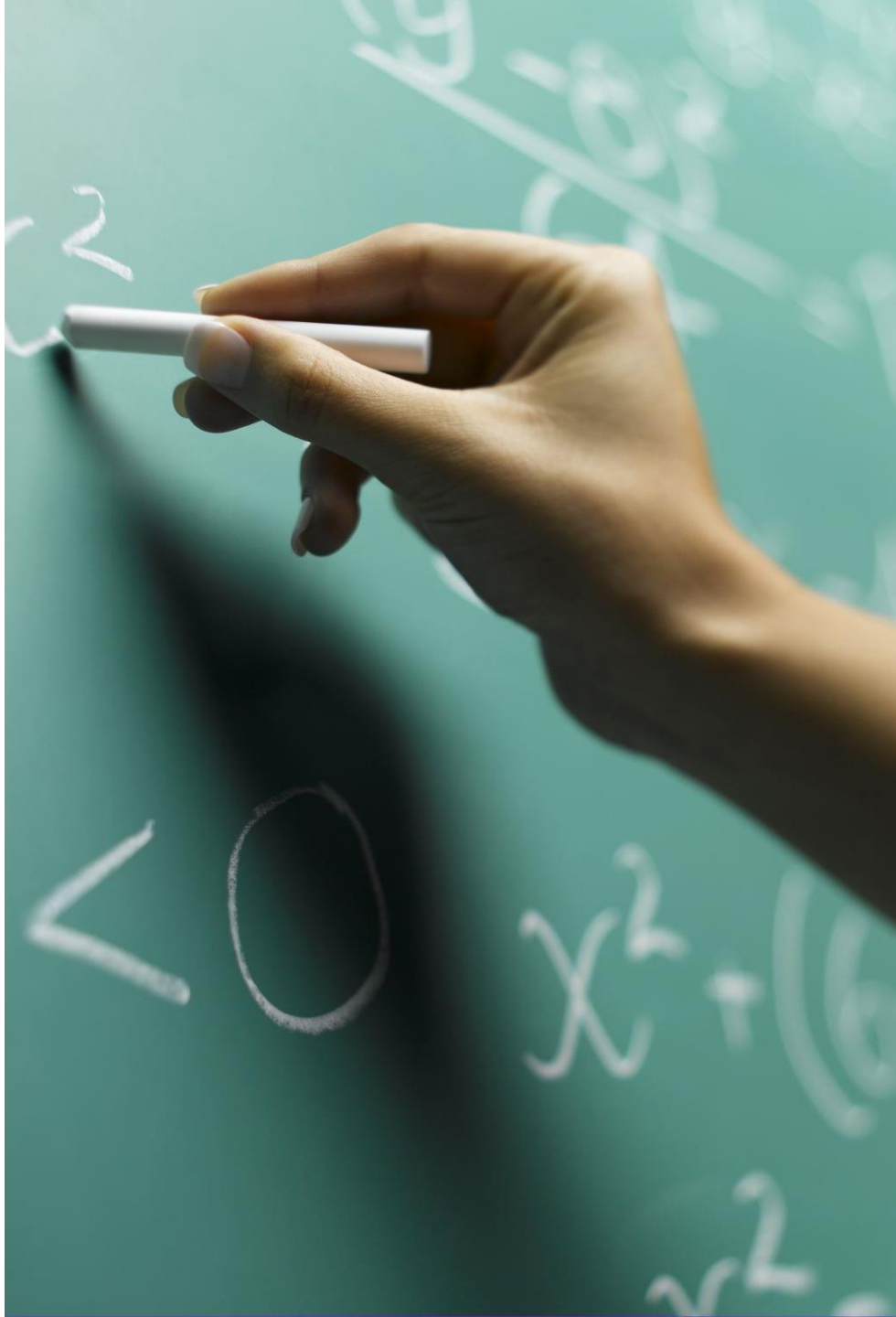
FRAGMENTS

https://react.dev/reference/react/Fragment

WHAT ARE ‘PROPS’

It means properties

props to a component are what attributes are to HTML.



The constructor function will be called when your component is being initialized

```
class Test extends React.Component{
  constructor(props){
    super(props);
  }
}
```

CONSTRUCTOR (LEGACY)

PROPS EXAMPLE (Legacy)

```
ReactDOM.render(<LegacyApp headerProp = “Header”
                           contentProp = “Content”/>,
   document.getElementById('app’));


class LegacyApp extends React.Component {
    constructor(props) {
       super(props);
    }

    render() {
      return (
         <div>
            <h1>{this.props.headerProp}</h1>
            <h2>{this.props.contentProp}</h2>
         </div>
      );
   }
}
```

React components use props to communicate with each other.

Every parent component can pass information to its child components by giving them props.

Props might remind you of HTML attributes, but you can pass any JavaScript value through them, including objects, arrays, and functions.

PROPS WITH FUNCTIONS

PROPS WITH FUNCTIONS

```
function Weather(actual, feels, measure, low, high ) {
   return  (
     <section>
        <span>Weather: {actual} (feels {feels} - {measure}</span>
        <div>High: {high} /  Low: {low}</div>
      </section>
 )};

 //use

 <Weather actual="23" feels="27" measure="C" low="20" high="30"/>
```

```
<h1 class=“t”> is
      <h1 className=“t”>

<label for=“targetId”> is
      <label htmlFor=“targetId”>
```

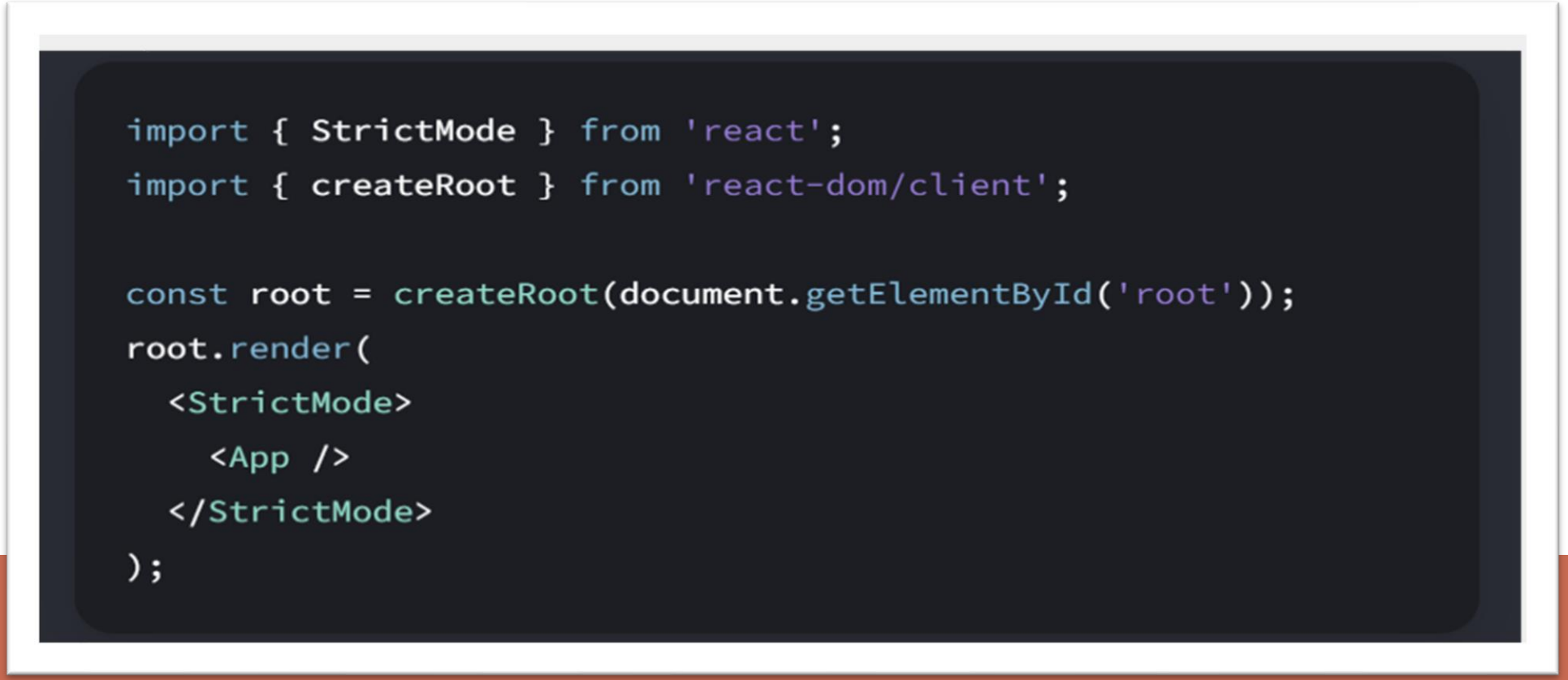
Renaming example of common HTML attributes

```
// stateless
function Welcome(client) {
  return (
    <h1>Welcome, {client}</h1>;
  );
}

// stateful
const [count, setCount] = useState(0);
function Welcome(props) {
  return (
    <h1>{count}</h1>;
  );
}
```

STATELESS VS. STATEFUL

StrictMode allows you to see more diagnostics while your application is running.



```
import { StrictMode } from 'react';
import { createRoot } from 'react-dom/client';

const root = createRoot(document.getElementById('root'));
root.render(
  <StrictMode>
    <App />
  </StrictMode>
);
```

STRICT MODE

https://react.dev/reference/react/StrictMode