Week 9

240

Week 8 Review

- The React object contains this method that defines an object that is part of the page UI
- The ReactDOM object contains this method that displays a React object on the page
- In the call to React.createElement, there are three parameters. They are …

241

Creating a React element with “props”

- Create an element with React

```
<div id="myapp"></div>  <!– place to put the React element(s) -->
   <script>
     var hello = React.createElement("h1",{id:"main"},"Hello React World!");
     ReactDOM.render(hello, myapp);
   </script>
```

- Add more attributes

```
var link = React.createElement("a",
            {className:"link-button",  href: "https://extension.harvard.edu",
             target:"_blank"},
             "Learn @ Extension");
```

242

React with child elements

- Add “child” elements

```
var select = React.createElement("select",{name:"pick"},
        React.createElement("option",null,"Basic"),
        React.createElement("option",null,"Standard"),
        React.createElement("option",null,"Premium")
    );
```

- Creating a hierarchy

```
var nav = React.createElement("p",
            {id:"nav"},
            React.createElement("a",  {href: "./about"}, "About Us")
        ) //end “p” element
```

243

Array of child elements

- A set of child elements can be represented as an array as well as a comma delimited list
- That means that anything that produces an array (like .map() !) can be used to generate the set of child elements
- Recall the <select> element- implement the child elements as an array:

```
React.createElement("select",{name:"pick"},
        [     React.createElement("option",null,"Basic"),
              React.createElement("option",null,"Standard"),
              React.createElement("option",null,"Premium")
        ])
```

244

Try It Part 4

- Create a bulleted list of 5 names using an array of <li> elements (all created using React.createElement)
- View and test your page

245

React and Arrays

- **Question:** What advantage is there to using an array of elements for the child elements?
- **Answer**: We can take advantage of the inherent patterns using a loop or loop construct.

246

React and Arrays

- Anything that produces an array (such as .map() or filter()) can be used to generate the array of child elements
- Consider: What if we created the set of options for a select element in a separate array

```
options = ["Basic", "Standard", "Premium"];
```

- Now we can utilize map() to construct the createElement code:

```
var select = React.createElement("select",
    {name:"pick"},
    options.map((item)=>React.createElement("option",null,item))
    ); //end select element
```

247

Try It Part 5

- Update part 4 as follows:
- Put the names in a separate array
- Use .map() to create the array of <li> elements
- View and test your page

248

Unique Key requirement

Warning: Each child in an array or iterator should have a unique "key" prop.

- The React object has a "key" property that can be used to add a unique property to each item.
- The index for each item in the array could be used as a unique value for the key
- This means we need a way to iterate through the array elements and also preserve the index value.

249

Unique Key requirement

- Fortunately, map() passes a second parameter to the function which is the index
- Example:

```
a = [2,5,9];
list = a.map((number, idx)=> idx + " – " + number+ "<br>");
```

Will display:

0 – 2

1 – 5

2 - 9

250

Unique Key requirement

- We can use the index from map as the unique key

```
options = ["Basic", "Standard", "Premium"];
var select = React.createElement("select",{name:"pick"},
    options.map((item, i)=>React.createElement("option",{key:i},item)
                    ); //end map
); // end create Element
```

251

Try It- part 6

- Update **part 5** to add a unique key for each array element.
- View and test your page

252

Multiple Data Elements using an Object

- Sometimes you will need to parameterize more than one piece of data (for example, a name and a url)
- You can create an object to store multiple data items.
- Example – create a select element with both text and a value
  <option value='base'>Basic</option>

```
function Level(name, code)
{
        this.name = name;
        this.code = code;
}

options = [new Level("Basic","base"),
           new Level("Standard","std"),
           new Level("Premium","prem")];

var select = React.createElement("select",{name:"pick"},options.map(
        (item,i)=>React.createElement("option",{key:i,value:item.code},item.name)) );

ReactDOM.render(select, myapp);
```

253

Try It - Finale

- Rework the Finance/Weather/Sports problem to use an array of objects that include the item name, the corresponding url.
- Utilize the array with .map() to build the element using React.
- Challenge: Add a boolean that is true when this is an external link (and therefore needs target='_blank')

254

Demo: Color Grid

- We want to create something like this:

1. Create an object to represent the key value and the background color
2. Create an array of the objects
3. Use React.createElement to set up an array of all of the "keys"
4. Add styling and render

255

React Components

- A component is a React element that can be used as a building block
- Use case: a site with multiple nav menus. The nav menu element can be constructed as a parameterized component
- Components can be created with functions or classes
- We can pass parameters to the component through key/value pairs in createElement (props)

256

Function Components

- *Create the React element as usual but wrap it in a function that returns the element*

```
function MyPageTitle()
{
     return React.createElement("h1",null,"Welcome to My Page");
}
```

- *Render the element using ReactDOM.render() using the name of the component*

```
var theTitle = React.createElement(MyPageTitle,null,null);
ReactDOM.render(theTitle, myapp);
```

257

Try It- Components Part 1

1. Create a component called MyStaffMember that displays a name for an employee named "Bill Bailey". The name should be an h3
   - Render the component
2. Create a component called MyStaffMember that displays a name and title for an employee named "Bill Bailey" who is a "Sales Manager"
   - Render the component
3. Create a component called MyStaffName and MyStaffTitle. Use those as child elements for MyStaffMember

258

Add a parameter via props

```
function myPageTitle(props)
{
      return React.createElement("h1",null,props.title);
}

ReactDOM.render(
     React.createElement(
           myPageTitle,
           {title: "Welcome to my better Page"},
           null),
     myapp);
```

259

The component can have other properties

```
function myPageTitle(props)
{
    return React.createElement("h1",{className:"green"},props.title);
}
ReactDOM.render(
      React.createElement(
            myPageTitle,{title: "Welcome to my even better Page"},null),
      myapp);
```

260

Parameterize additional properties

```
function myPageTitle(props)
{
      return React.createElement(
            "h1",
            {className:props.myclass},
            props.title);
}
ReactDOM.render(
      React.createElement(myPageTitle,
            {title: "Welcome to my really great Page",
             myclass:"green"},
                  null),
      myapp);
```

261

Try it – Components part 2

1. Create a component called MyStaffMember that displays a name for an employee named "Bill Bailey". The name should be an h3. The name should be parameterized via props
   - Render the component
2. Create a component called MyStaffMember that displays a name and title for an employee named "Bill Bailey" who is a "Sales Manager". The name and title should be parameterized via props
   - Render the component
3. Create components called MyStaffName and MyStaffTitle. Use those as child elements for MyStaffMember. You will need to pass the props to all components

262

Destructure the props object

```
function myPageTitle({title, myclass})
{
      return React.createElement("h1",{className: myclass}, title);
}

ReactDOM.render(React.createElement(
            myPageTitle,
            {title: "Welcome to my really great  Page", myclass:"green"},
            null),
         myapp);
```

263

Try it – Components Part 2 - 4

- Create components called MyStaffName and MyStaffTitle. Use those as child elements for MyStaffMember. You will need to pass the props to all components
- Use destructuring for the props parameter

264

Creating Components Using Classes

- Components can also be created using a class (this is an older way of creating a component)
- You will need to derive your class from React.Component

```
class myPageTitle extends React.Component {
      render() {
            return React.createElement("h1",null,"Welcome to My Page");
      }
}

myTitle= React.createElement(myPageTitle,null,null);
```

265

Try it

- Rework the color grid to use components

266

Try it

Create an object to represent a staff member

- Each staff member should have a name and a title

1. Create a React component to render one staff member – the parameter should be an instance of the staff member object.
2. Create an array of 3 staff members

   Create a React component to render the 3 staff members using the array and the staff member component

267