Camosun College

ICS 211 - Web Applications

Lab 4 - React Components

Due Date: Lab Demo Due by Oct. 12 @ 2:20 PM

Lab Quiz (5 Questions) Due by Oct. 12 @ 11:59 PM

Background and Theory

React Components

React components are the "building blocks" of the React UI. You can think of them as LEGO bricks! Each are self-contained units that can be used throughout your application or even in other applications. Components allow you to group DOM elements together into a single cohesive unit. To create a React Component, you *extend* the `React.Component` base class:

```
class MyComponent extends React.Component {
  ...
}
```

Once you have created your Component, you instantiate an instance of it by passing it to `React.createElement()`. This creates the React Element and adds it to React's Virtual DOM. This function call can be passed as the first argument to `ReactDOM.render()` directly:

```
ReactDOM.render(
  React.createElement(MyComponent),
  document.getElementById('container')
);
```

Components are reusable. You can instantiate multiple copies of it on a page! For example:

```
ReactDOM.render(
  React.createElement('div', null,
    React.createElement(MyComponent),
    React.createElement(MyComponent),
    React.createElement(MyComponent)),
  document.getElementById('container')
);
```

A "wrapper" `<div>` is necessary because `ReactDOM.render()` accepts only a *single* element.

React Components also accept data. In React lingo, data is called *props*. Props are passed as the second argument to `React.createElement()`. They are passed as a JavaScript Object. For example:

```
ReactDOM.render(
  React.createElement(MyComponent, { aProp: 'a Prop value' }),
  document.getElementById('container')
);
```

In your Component class, you can use props with the `this.props` reference. For example, the above props value can be accessed by using `this.props.aProp`. Props can even be rendered as HTML attributes by passing `this.props` as data to `React.createElement()`.

See the associated lecture notes for more examples of React Components and using Components with props.

Task

Below is the template of the very simple website you are going to make with React Components. It will list at least *three* of your favorite internet sites (keep it PG):

Title

Subtitle

1. Site Name: http://www.example.com
2. Site Name: http://www.example.com
3. Site Name: http://www.example.com

For example, if the first site were Google, it would look like:

1. Google: https://www.google.ca/

Requirements for your React Website

1. The Title and Subtitle are up to you but are required. **Together, they should be a single React Component**.

2. In addition to the Title Component, you will make a **Site Component** that will accept Props. The two props it will accept are `siteLink` and `siteName`. It will use both in an `<li>` element that also contains an `<a>` child element. The `<a>` child element will have a `href` attribute that uses `siteLink`. The `href` attribute is the props for the `<a>` element.

3. `ReactDOM.render()` will render both the Title and all the Site Components. Thus, you will need to use `React.createElement()` to accomplish this. Use a `<div>` element as a wrapper around the Title Component and an `<ol>` element, which will be siblings. The `<ol>` element will have several Site

Components as its children (this **must** be an ordered list). Each child should pass values in for the `siteName` and `siteLink` props.

Lab Submission (see top of lab for due date)

1. Demo to the Lab Instructor (10 marks):
    - Your functioning website using `http-server`
    - Your React Code that shows:
        - The Title and Site Component Classes' implementations
        - The `ReactDOM.render()` method call
        - For full marks, all have to be implemented correctly and your results must match the template shown.
2. Do the Lab 4 Quiz in D2L (5 marks).