

React

—

VS Code Plugins

- Simple React Snippets
- Prettier - Code Formatter

What is React?

- React is a JavaScript library for building user interfaces.
- React is maintained by Facebook and a community of individual developers.
- React is primarily use for developing single page or mobile applications. (SPA)
- React is also referred to as ReactJS and React Native.
 - ReactJS is the one we will be learning in class.
 - React Native is used for mobile development and exposes native IOS and Android feature. Like GPS, Camera, etc.
- Although it is not required React uses JSX syntax to render HTML to the DOM.

What is JSX?

- JSX stands for JavaScript XML.
- JSX is a syntax extension to JavaScript.
- JSX looks similar to HTML.
- React recommends using JSX when programming with React.
- JSX tags look like HTML tags that transpile to JavaScript code that adds HTML to the DOM.

```
var element = <h1>Hello World!</h1>;
```

Why use React?

- Reusable Components
- Faster Rendering to the DOM using a virtual DOM
- Clean Abstraction
- Mobile Support with React Native

React Virtual DOM

- A browser DOM is in most cases a representation of your HTML elements in memory.
- The DOM gives a layer of abstraction to your browser.
- The problem with the DOM is it is very inefficient when modifying HTML elements frequently.
- React adds another layer of abstraction using a Virtual DOM and only pushes things that have changes from the Virtual DOM to the browsers DOM.
- This makes changes very fast and efficient.

A close-up shot of Leonardo DiCaprio from the chest up, wearing a dark suit and white shirt. He is looking slightly to his right with a serious expression. The background is blurred, showing what appears to be an office or meeting room setting.

**I CAN'T WAIT FOR ABSTRACTION
OF VIRTUAL DOM**

WE NEED TO GO DEEPER

Let's set up our first React app!

- First we'll need a couple things
 - We'll need npm installed on our computer.
 - To Test run: `npm -v` (you should see a version number appear like `6.1.0`)
 - We'll also need to install a package called: `create-react-app`
 - To test whether you have it installed run: `create-react-app --help` this should not error if you have it installed.
 - If you do not have it installed run the command: `npm i -g create-react-app`
- At this point we should have enough to make our first app. To do so run the command:
`create-react-app <project name>`
 - Example: `create-react-app hello-world`
- This will install a boiler plate project for running a react project!

Running our React app

- Once npm is finished downloading required files you can now navigate into the project directory and run: `npm start` to start your React app.
- This should open a browser connecting to <http://localhost:3000/> and show a template React page.
- Port 3000 is the default port React runs on.

React's Entry Point

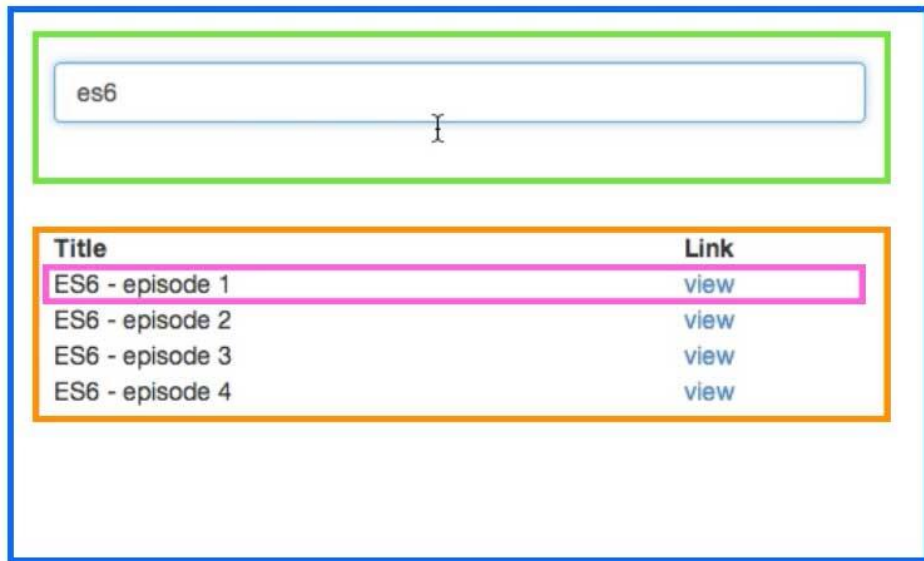
- In a simple description, React loads a single html page and then modifies that page using JavaScript.
- It does this by interacting with the Virtual DOM.
- This is imported from the react-dom library.
- ReactDOM is the magic happening behind the scenes.

```
import React from 'react';
import ReactDOM from 'react-dom';

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

Components

- Everything in react is build with components.
- There are two types of components, functional and class components.
- Components are like the building blocks of react.
- Think of components as any piece of reusable code that makes sense grouped together.



State

- State is the place in your component to put your variables or anything that your component relies on to function.
- State should always be immutable data.
- They are used similar to how HTML attributes are associated to HTML elements and is used by calling: `this.state.<stateVariable>`

Props

- Props are how components communicate with each other.
- They are used similar to how HTML attributes are associated to HTML elements.
- They get passed into the component and are used by calling: `this.props.<propertyName>`

Exercise

Create a Component that holds information about a User.

The component should contains a header tag with the name of the user, a section for their bio, a user profile image, and their email.

The component should accept all of the user's information as properties.

Duplicate the component a couple times with different users.