**React.Js + Redux**

**What is React**?

* JS library created by Facebook
* Also used by Netflix & Instagram
* Used to create JS-Driven dynamic web apps
* In that regard, can be compared to angular & Vue

**What is Redux?**

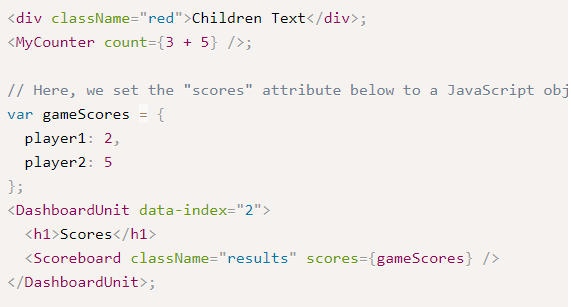
* A layer on-top of React
* Helps with state management of our app
  + Data in the app
  + UI state of the app
* Learn this in the 2nd half of the playlist

**What is JSX?**

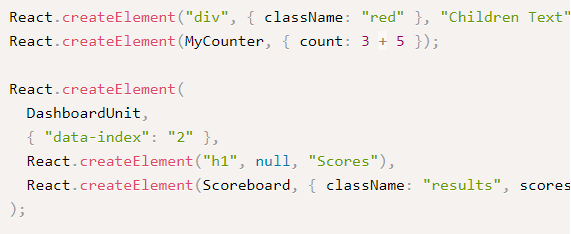
* In terms of React. We are writing XML with our React code.
* Jsx is a pre-processor step that adds XML syntax to JS
* So we can say that JSX makes react a lot more elegant

Example:

React code.



Compiled to the following without JSX:

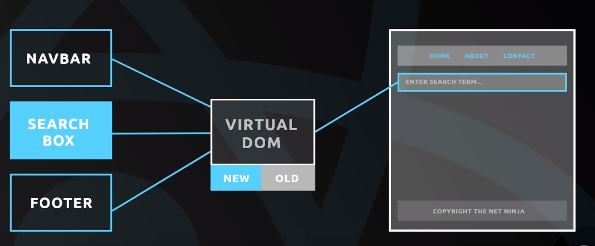


**What we will learn**

* React basics
  + Components, events, templates, props & forms
* React Router
  + Routes, route parameters, redirects
* Redux
  + Stores, actions & reducers

**Lesson 2**

Components are important!

Example: Navbar, search box, footer

Components & Templates

* Components look like HTML templates (JSX javascriptxml file)
* They can contain ‘state’(data or UI state)
* They also can contain JS for functionality

**Lesson 3**

**Set up react**

1. Get CDN from react official website

<script crossorigin src="https://unpkg.com/react@16/umd/react.development.js"></script>

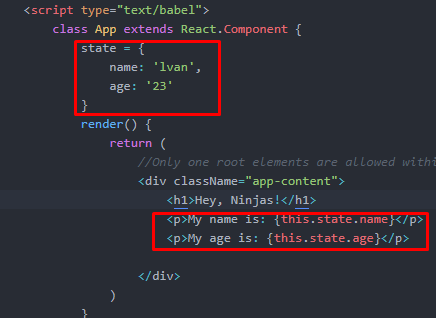
<script crossorigin src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>

1. Get the babel cdn from babel official website
2. 

**Lesson 4**

**Component State**

* JavaScript Object
* Describes the current state of the component
  + Data, UI-state
* The state of a component can be updated overtime
  + A modal could close
  + The data we output could change

Example: 

**Lesson 6**

\*There is a chrome extension called React dev tool which can manually change the state within the chrome browser

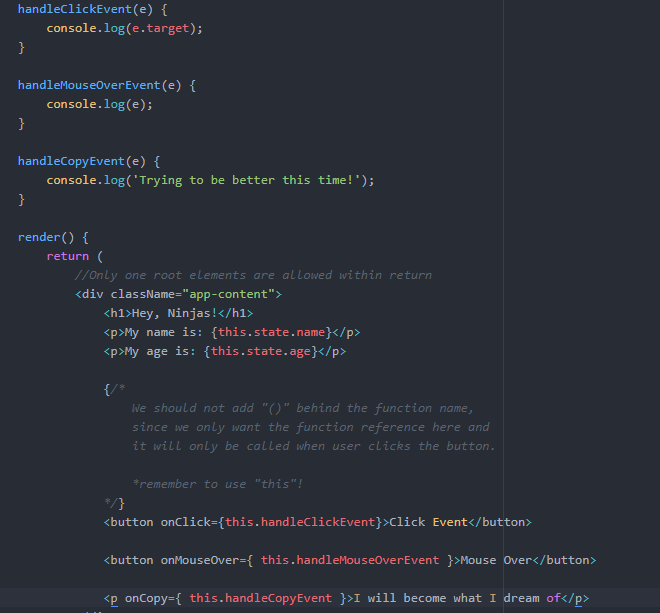
**Lesson 7 EVENT**

**Notes\***

Remember to add “this” before refencing the function

Remember not to add “()” behind the function references

Remember not to use this.state inside the function implementation



**LESSON 8 Change State Value**

Don’t use this.state.value = “new value”

But use

this.setState({

value: “Some new value”

});

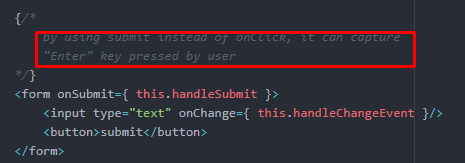
**LESSON 9 FORM**

Notes\*

1. The default action for website when submitting form is to refresh the page.

To Prevent this default event behaviour use

e.preventDefault()

1. 

**LESSON 10 Create React App**

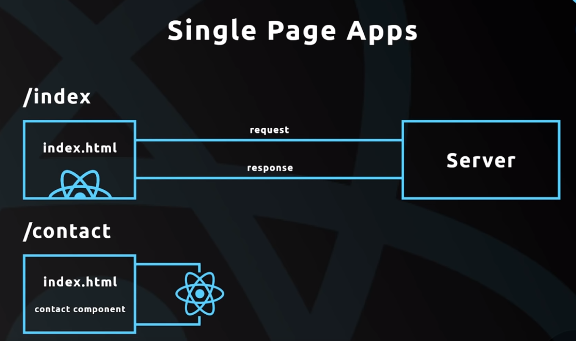
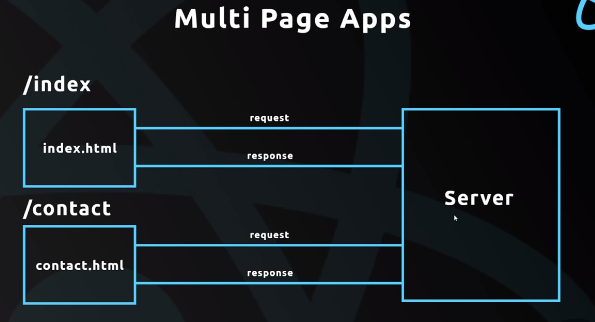
* A command line tool to create React apps
  + Development server
  + Use **ES6(Wut is this???)** features which are not normally supported
  + Keep our code modular
  + Use build tools to create optimized code

Check links:

<https://github.com/facebook/create-react-app>

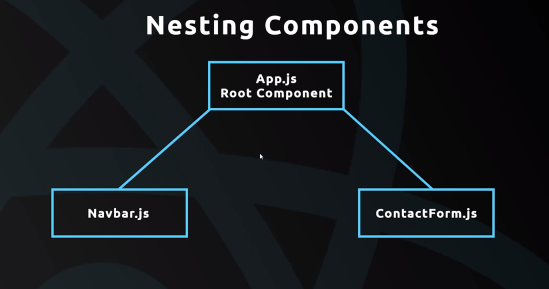
**LESSON 11 Single page React App**

* React apps are typically SPA’s
* Only ever one html page served to the browser
* React then controls what a user sees on that page



**LESSON 12 Nesting Components**

App.js => root component(mother of all the components)



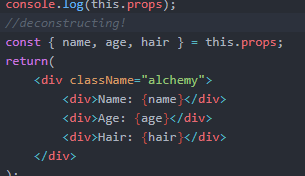


**Lesson 13 PROPS**

**App.js:**



**Alchemy.js:**

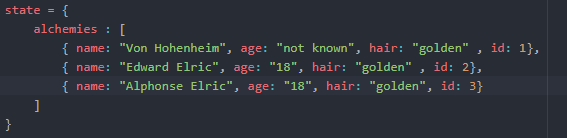


**Lesson 14 outputting lists**

**How to pass multiple values Alchemies?**

Utilizing the JS Map() function:

1. Creating state that contains alchemies list in App.js



1. Get the props value and utilize map() function to rendering all alchemy



**Lesson 15 Stateless Components**

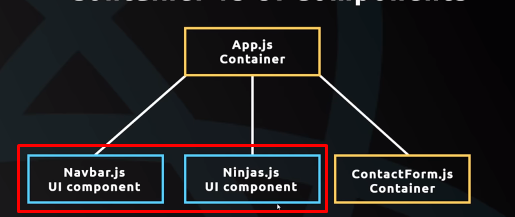
**Container vs UI Components**

Container:

1. Contain state
2. Contain lifecycle hooks
3. Not concerned with UI
4. Use classes to create

UI Components

1. Don’t contain state
2. Receive data from props
3. Only concerned with UI
4. Use functions to create



Blue components are UI Components which doesn’t require data and don’t have state. And we usually use function to create. Example:

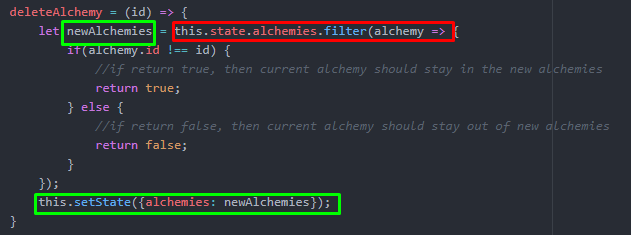


Orange components are Container

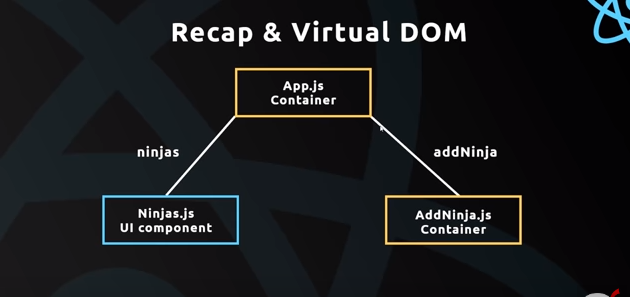
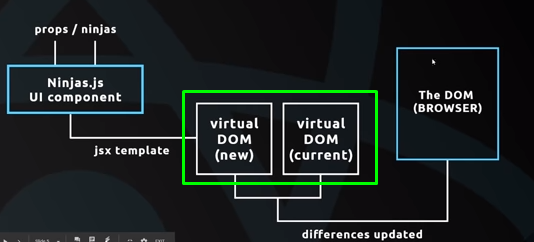
**Lesson 17 Form revisited**

**Lesson 18 Functions as Props**

**Lesson 19 Deleting Data**

Using filter() function to create the new state whose unwanted date set have been removed from the original collection

**Lesson 20 Recap & Virtual DOM**

So when whenever we change the the state and ninjas will have a new props which will cause differences between **virtual dom(new)** and **virtual DOM(old)** so that the current virtual dom got updated which cause the browser dom got updated at the same time.