**Hello and welcome!**

My name is Ken, and I am here to talk about the React programming language. I’ll give you an introduction, then we will build an application, which is one of the best ways to learn.

Features from ES6 – a recent version of JavaScript will be used for this course.

Hopefully, you have some familiarity with HTML and JavaScript, but it is not required.

Please see other courses available on Skillshare.com for these classes if you can, because the more knowledge and experience you gain, the better.

--------------------------

What is React?

React is the visual aspect of websites. Think of it as the interior of a car or house, if you will. The structure is there, but you are changing the look and features.

React JavaScript is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components”.

In technical terms such as MVC – Model-View-Controller, React is the View.

Why use React?

When you need to have multiple parts/components such as active/inactive navigation items, accordion sections that expand and collapse, dynamic inputs, buttons that can become active or disabled, user login and permissions, React is a good choice. It helps to manage those changing states and with presenting different or dynamic views to the user based on the state or conditions.

Virtual DOM

The common practice was to directly manipulate the DOM (the Document Object Model). For those who need a refresh it’s the parts that make up a web page.

(show DOM image). React is very efficient because it keeps a virtual copy of the page’s DOM and only changes the parts of the DOM that actually changed and not the entire DOM. Frequent changes and complexity slow down full DOM update. With React it becomes much faster.

Reusable parts/components.

No React tutorial would be complete without mentioning reusable components. Components are first class members in React and component-based web development is here to stay. React supports the creation of components that you can call at any time, combine, and nest with other components. This saves a lot of time for the developer, since something that has been built once can get reused in multiple places throughout the application. It also speeds up the process of making changes or updates to a component. Think of components as the lego blocks that fit together to make up your web application.

Components can even be easily shared between multiple applications as your project scales out and grows.

## **JSX Syntax**

While you can use regular Javascript with React, JSX syntax allows for expanded functionality. Once you learn JSX, it actually becomes easier to write code for React, and React can easily transform that JSX code into a compatible format. JSX allows you to mix your HTML directly into the Javascript, without the need for separate template files. Doing so allows you to organize your code based on logical concerns (read: by component) and not by technology.

## **Redux & Other Libraries**

React gets especially powerful when you combine it with other libraries. One of the most popular is [Redux](https://redux.js.org/). Redux follows a simplified flux pattern and provides a single state container where you keep all your application data. It becomes a single source of truth for your application’s state, and it makes it easier to observe and manipulate relevant data. React then re-renders with any change in the data container.

Another valable library that supports the component driven design concept is [Storybook](https://github.com/storybooks/storybook). Storybook allows for the isolated design and hosting of individual components. Being able to do this allows developers to build highly reusable components that are not tightly bound to the context in which they will live in the application. It also provides a great playground for designers and developers to talk on an even playing field.

## **React Native**

An added bonus for learning React library is you can now [create mobile apps using React Native](https://www.intertech.com/Blog/react-native-pros-cons-building-mobile-apps-just-got-simpler/). There is a slight hurdle of a few added things you’ll need to learn. However, React Native’s mobile apps compile the Javascript you write to native code, making the apps you develop look and feel like other platform-native apps.

## **Community Support & Developer Tools**

Finally, React has a large user base and active support from the engineers at Facebook. It’s a growing and evolving library. If you have questions or need to borrow some code to get a component up and running, React’s open source community is great. It’s also easy to find a free React tutorial online to get started actually writing code. We recommend the video tutorial linked at the end of this article.

They also have developer tools for inspecting and debugging your applications. You can see exactly what is going on with your application in real time, including observing the virtual DOM. This makes it super easy to diagnose problems and double check that everything is working as it should,

Using CodePen.

Codepen a useful development environment for web design and development. The site is: <https://codepen.io/>

What is React

Why use React

Set up environment

Check for updated NodeJS or download: <https://nodejs.org/en/>

npm -g install npm

(the g is global so that you can have it on the machine and not the directory)  -needs verification

Now we

npm install -g create-react-app  
create-react-app my-app

react start my-app

Congratulations! You created a React Application! Time to open it up to see what is inside.

Now we move on to a more useful app. We will create a To Do List.

Find src and erase its contents?

What about a shopping list?

class ShoppingList extends React.Component {

render() {

return (

<div className="shopping-list">

<h1>Shopping List for {this.props.name}</h1>

<ul>

<li>Instagram</li>

<li>WhatsApp</li>

<li>Oculus</li>

</ul>

</div>

);

}

}