React codeercursus

Leer coderen met React met behulp van screencasts en oefeningen.

Deze tutorial bevat 48 interactieve screencasts en is het perfecte startpunt voor aspirant React-ontwikkelaars. Je leert alle kernbegrippen, terwijl je tegelijkertijd twee apps bouwt en coderingsuitdagingen aan het doen bent.

48 lessen | 302 minuten

Na het voltooien van deze cursus, weet je alles wat je nodig hebt om webtoepassingen te bouwen in React. We behandelen:

JSX
Props and state
Conditional rendering
Styling components
Lifecycle methods
Fetching data from an API
Handling events
Forms and controlled components
Writing modern React JS code
Setting up an environment outside of Scrimba

1. Course Introduction & Learning Philosophy

Welcome to an Introduction to React! This course introduction video will give some of my background and a preview of what you'll be learning throughout this course!

Philosophy

- The easiest way to learn is by doing it the hard way
- Learn by doing. Scrimba is *perfect*for this
- Spaced learning and repetition are key
- Don't binge the course. Rest often. Review courses

What you'll learn

- Components
- JSX
- Styling components
- Props & state
- Event handling
- Lifecycle methods
- HTTP
- Forms
- More!

What you should know beforehand

- HTML
- CSS
- JavaScript
- ES2016/ES6

1.1 What we'll be building

We take a look at the projects we'll be building throughout the course: the standard (required?) todo list app and an (only-sometimes-funny) meme generator.

1.2 Why use React?

We discuss some of the reasons React has become so popular and why

it's worth learning as a front end web applications framework.

2. React Container & Component Architecture

Sometimes the amount of work a single component is in charge of doing can become unruly, as in the component is in charge of doing too many things. Using a container/component approach to splitting apart the business logic from the display logic can be a great way to separate the concerns of a given component.

3. ReactDOM & JSX

We learn how to spin up the most basic React app possible using ReactDOM and touch on some key points about JSX.

3.1 ReactDOM & JSX Practice

Now it's your turn - put your new knowledge from the last screencast to the test by creating the boilerplate React code needed to render a few list items in an unordered list.

4. React Functional Components

Learn how to create your first React component using functional components.

4.1 React Functional Components Practice

Practice from the previous lesson on functional components. You'll build a simple functional component that returns some UI with your name, a paragraph about yourself, and a list of your most-desired vacation spots.

5. Move Components into Separate Files

We learn to use import/exports in ES6 to separate our components into their own files. We also spend a little time talking about project organization.

6. React Parent/Child Components

We dive deeper into the concept of creating parent/child components. Each piece of HTML can (and often should) be separated into its own component.

6.1 React Parent/Child Component Practice

This is a chance to practice the parent/child component hierarchy.

7. React Todo App - Phase 1

We're starting a new todo app from scratch! Throughout the course, as we learn more about React, we'll come back to this app and add to it until it's everything we ever wanted it to be!

7.1 Styling React with CSS Classes

Learn how to add style to your React applications using CSS classes.

7.2 Some Caveats

Before moving on, I cover a few quirks of my own way of writing JavaScript and React code. I wanted to address them now so it isn't bugging (or confusing) anyone as we continue to learn.

7.3 JSX to JavaScript and Back

JSX is great and all, but sometimes you just need to put some regular JavaScript inside your JSX. This lesson will prepare us for the next one about inline styling, but is used all over React in general.

7.4 React Inline Styles with the Style Property

React gives us an easy way to dynamically style our components based on data through inline styling. We cover some examples of how we can use dynamic data and inline styles to make an application that can change its own style.

8. React Todo App - Phase 2

Now that we've learned about parent/child components and styling, let's add some of this to our ongoing todo list project!

8.1 React Props Part 1: Understanding the Concept

A quick introduction to the concept of props in React by looking at something we're already familiar with - HTML element attributes.

8.2 React Props Part 2: Reusable Components

One more conceptual look at the Youtube.com homepage and how (if it were developed in React) it might be broken down in to reusable components.

8.3 React Props

This lesson covers the primary way to pass data from a parent component down to a child component, and in the process allows us to create components that are more and more reusable.

8.4 React Props and Styling Practice

In this practice, you'll create 5 <Joke/> components, pass data down to each one, and try your hand at adding some styling to them.

8.5 Mapping Components in React

8.6 Mapping Components Practice

Given an array of products, display each product in its own components on the page using JavaScript's array .map() method. Good luck!

9. React Todo App: Phase 3

Using what we learned about mapping and props, we're going to turn our hard-coded <TodoItem /> components into dynamic components that can display their own data. Essentially, this means we're declaratively using raw data and easily turning it into something that gets displayed on the screen!

9.1 Class-based Components

In this lesson you'll learn about class-based components (components created using the JavaScript class keyword instead of a functional component) and when/why you'd want to use them.

9.2 Class-based Components

Take some time to practice converting functional components into class components.

9.3 React State

State is an extraordinarily important concept in React. Any time a component needs to be in charge of maintaining its own information (as opposed to being passed that information from a parent component via props), you'll use state. Let's take a look at the very basics of state, and soon we'll learn how to change state.

9.4 React State Practice

Here's a chance to do some debugging. If you run in to an error you don't understand, take time to Google the error before just watching the solution! The debugging process is really important for you to be able to handle these kinds of errors in the real world.

9.5 React State Practice 2

Another chance to practice what you've learned!

10. React Todo App: Phase 4

In this phase, we move the todos data we've imported into state so that we can actually make modifications to the data in a future phase.

10.1 Handling Events in React

Handling events are what drive a web application and set it apart from simple static websites. Handling events in React is fairly simple, and looks very similar to the way it's done in regular HTML. Methods like onclick and onsubmit are just camelCased and put into React the same way (onClick, onSubmit).

11. React Todo App: Phase 5

Let's make another small addition to our todo app. When we added the checked property to our <TodoItem /> component, we were getting an error that we weren't providing a way to change the state. We'll write the code to actually change the state in an upcoming lesson, but for now let's get rid of the warning by simply adding an onChange handler to our <TodoItem />.

11.1 React setState: Changing the State

State by itself isn't very useful unless it can be changed. But unlike props which are immutable, state was intended to be changed whenever necessary. However, you should never directly change state directly, but instead should use the setState method to do so. This lesson covers the different ways you can use setState to make changes to your component's data saved in state.

12. React Todo App: Phase 6

We finally get to the bulk of the todo app, implementing the logic we need to change the completed state of whichever todo item we click.

12.1 React Lifecycle Methods Part 1

Lifecycle methods are a crucial tool given to developers so we can trigger certain actions when a component undergoes a specific part of its lifecycle.

12.2 React Lifecycle Methods Part 2

When React 16.3 was released, they announced the removal of 3 lifecycle methods (componentWillMount, componentWillReceiveProps, and componentWillUpdate) and the introduction of 2 new methods (static

getDerivedStateFromProps and getSnapshotBeforeUpdate). Although we won't be using these much in this course, let's take a few to better understand these new lifecycle methods.

12.3 React Conditional Render

Conditional Rendering is used anytime you sometimes want something to display on the screen. In this lesson, we see how we might pop up a loading screen while something else is happening in the background, and then replace it with something else when the data has loaded.

12.4 React Conditional Render Part 2

Using the conditional AND (&&) is a useful way to simply render something or nothing at all, depending on the truthiness of the condition you're checking against.

12.5 React Conditional Render Practice

This is your chance to practice conditional rendering as well as numerous other lessons we've covered recently. Good luck!

13. React Todo App: Phase 7

We finish off our todo app by adding some conditional styling to the

completed items. If you haven't had a chance to work through these phases of the todo app, this would be a great time to either go back and work through them, or to even try building a todo app from scratch by yourself. Remember - learning how to do something happens when you actually try to do it.

13.1 Fetching data from an API with React

One of the most common things your web app will need to do is be able to grab data from an external source of some kind. This lesson will cover how to do that in React using the native fetch function and the Star Wars API

14. React Forms Part 1

The way you build forms in React is a bit different than how you may be used to doing so in vanilla JavaScript, so the next 2 lessons are dedicated to how you can build forms in an easy and reusable way.

14.1 React Form Part 2

In this lesson we cover some of the other most common form elements: text areas, checkboxes, radio buttons, and select boxes.

14.2 React Form Practice

Let's build a travel form that gathers your personal and trip information. Feel free to refer back to the previous lessons on forms if you get stuck.

15. React Meme Generator Capstone Project

In this screencast we're going to build the Meme Generator I talked about in the start of the course. We'll start almost from scratch (except from some basic styling). It'll be a mix of teaching and challenges, so this is going to be really fun!

16. Writing Modern React Apps

We cover some of the newer features in React, and I include links to both new and advanced React features and patterns that will be helpful for you to learn as you dive deeper into React

17. React Project Ideas for Practicing

Since the best way to learn React is to actually create lots of React apps, I've included links to some of Free Code Camp's blog posts that enumerate a ton of ideas you could use as toy projects to help you learn.

18. Conclusion - React Tutorial

You've made it! Let's recap some of the topics we've learned about in this course.