



# TABLE OF CONTENTS: THEORY LECTURES (CLICK THE TITLES)

- |   |  |   |
|---|--|---|
| <a href="#">1</a> Watch Before You Start!                         | <a href="#">14</a> State vs. Props                             | <a href="#">27</a> How Diffing Works                              |
| <a href="#">2</a> Why Do front-end Frameworks exist?              | <a href="#">15</a> What is "Thinking in React"?                | <a href="#">28</a> The Key Prop                                   |
| <a href="#">3</a> What is React?                                  | <a href="#">16</a> Fundamentals of State Management            | <a href="#">29</a> Rules for Render Logic: Pure components        |
| <a href="#">4</a> Setting Up a New React Project: The Two Options | <a href="#">17</a> Derived State                               | <a href="#">30</a> State Update Batching                          |
| <a href="#">5</a> Components as Building Blocks                   | <a href="#">18</a> The children Prop: Making a Reusable Button | <a href="#">31</a> How Events Work in React                       |
| <a href="#">6</a> What is JSX?                                    | <a href="#">19</a> How to Split a UI Into Components           | <a href="#">32</a> Libraries vs. Frameworks & The React Ecosystem |
| <a href="#">7</a> Separation of Concerns                          | <a href="#">20</a> Component Categories                        | <a href="#">33</a> Section Summary: Practical Takeaways           |
| <a href="#">8</a> Props, Immutability, and One-Way Data Flow      | <a href="#">21</a> Component Composition                       | <a href="#">34</a> The Component Lifecycle                        |
| <a href="#">9</a> The Rules of JSX                                | <a href="#">22</a> Props as a Component API                    | <a href="#">35</a> A First Look at Effects                        |
| <a href="#">10</a> Section Summary                                | <a href="#">23</a> Components, Instances, and Elements         | <a href="#">36</a> The useEffect Dependency Array                 |
| <a href="#">11</a> What is State in React?                        | <a href="#">24</a> How Rendering Works: Overview               | <a href="#">37</a> The useEffect Cleanup Function                 |
| <a href="#">12</a> The Mechanics of State                         | <a href="#">25</a> How Rendering Works: The Render Phase       | <a href="#">38</a> React Hooks and Their Rules                    |
| <a href="#">13</a> More Thoughts About State + State Guidelines   | <a href="#">26</a> How Rendering Works: The Commit Phase       | <a href="#">39</a> useState Summary                               |



# TABLE OF CONTENTS: THEORY LECTURES (CLICK THE TITLES)

**40** [Introducing Another Hook: useRef](#)

**41** [What are Custom Hooks? When to Create One?](#)

**42** [Class Components vs. Function Components](#)

**43** [Managing State With useReducer](#)

**44** [Section Summary: useState vs. useReducer](#)

**45** [Routing and Single-Page Applications \(SPAs\)](#)

**46** [Styling Options For React Applications](#)

**47** [What is the Context API?](#)

**48** [Thinking In React: Advanced State Management](#)

**49** [Performance Optimization and Wasted Renders](#)

**50** [Understanding memo](#)

**51** [Understanding useMemo and useCallback](#)

**52** [Optimizing Bundle Size With Code Splitting](#)

**53** [Don't Optimize Prematurely!](#)

**54** [useEffect Rules and Best Practices](#)

**55** [Introduction to Redux](#)

**56** [Redux Middleware and thunks](#)

**57** [What is Redux Toolkit \(RTK\)?](#)

**58** [Redux vs. Context API](#)

**59** [Application Planning \("Fast React Pizza Co."\)](#)

**60** [What is Tailwind CSS?](#)

**61** [Application Planning \("The Wild Oasis"\)](#)

**62** [What is Supabase?](#)

**63** [Modeling Application State](#)

**64** [What is React Query?](#)

**65** [An Overview of Reusability in React](#)