Useful Resources for Part 1

Here are some **resources** that you might find helpful while studying this Part 1.

👉 [React](https://react.dev/?ref=jonas.io) (Documentation that you should keep open at all times)

👉 [Create React App](https://create-react-app.dev/?ref=jonas.io) (This is how we'll setup our first app)

👉 [Vite: Getting Started](https://vitejs.dev/guide/?ref=jonas.io) (For real-world React apps)

👉 [Adding React URL to an HTML Document](https://gist.githubusercontent.com/gaearon/0275b1e1518599bbeafcde4722e79ed1/raw/db72dcbf3384ee1708c4a07d3be79860db04bff0/example.html) (For the "Pure React" lecture)

Useful Resources for Part 2

Here are some **high-quality resources** that you might find helpful while studying this Part 2. (The list follows the order of sections in this part)

***These resources are 100% optional, but might give you additional insights or provide ever deeper dives into some topics.***

👉 [Writing Resilient Components](https://overreacted.io/writing-resilient-components/?ref=jonas.io) (By Dan Abramov from the React team)

👉 [Things I think about when I write React code](https://github.com/mithi/react-philosophies?ref=jonas.io) (GitHub repository)

👉 [A (Mostly) Complete Guide to React Rendering Behavior](https://blog.isquaredsoftware.com/2020/05/blogged-answers-a-mostly-complete-guide-to-react-rendering-behavior/?ref=jonas.io) (By Mark Erikson from the redux team)

👉 [A Visual Guide to React Rendering](https://alexsidorenko.com/blog/react-render-always-rerenders/?ref=jonas.io) (A multi-part series, check out the other ones)

👉 [Inside Fiber: in-depth overview of the new reconciliation algorithm in React](https://indepth.dev/posts/1008/inside-fiber-in-depth-overview-of-the-new-reconciliation-algorithm-in-react?ref=jonas.io)

👉 [A Cartoon Intro to Fiber](https://www.youtube.com/watch?v=ZCuYPiUIONs?ref=jonas.io) (YouTube video)

👉 [What Is React Fiber? React.js Deep Dive](https://www.youtube.com/watch?v=0ympFIwQFJw?ref=jonas.io) (YouTube video)

👉 [The React and React Native Event System Explained](https://levelup.gitconnected.com/how-exactly-does-react-handles-events-71e8b5e359f2?ref=jonas.io)

👉 [Under the hood of event listeners in React](https://gist.github.com/romain-trotard/76313af8170809970daa7ff9d87b0dd5?ref=jonas.io)

👉 [A DIY guide to build your own React](https://github.com/pomber/didact?ref=jonas.io)

👉 [useSyncExternalStore First Look](https://julesblom.com/writing/usesyncexternalstore?ref=jonas.io)

👉 [Under the hood of React's hooks system](https://the-guild.dev/blog/react-hooks-system?ref=jonas.io)

👉 [Why Do React Hooks Rely on Call Order?](https://overreacted.io/why-do-hooks-rely-on-call-order/?ref=jonas.io) (By Dan Abramov

👉 [So you think you know everything about React refs](https://blog.thoughtspile.tech/2021/05/17/everything-about-react-refs/?ref=jonas.io)

👉 [react-use: Reusable React Hook Library](https://github.com/streamich/react-use?ref=jonas.io) (GitHub repository)

👉 [react-hookz: React hooks done right](https://github.com/react-hookz/web?ref=jonas.io) (GitHub repository)

Useful Resources for Part 3

Here are some **high-quality resources** that you might find helpful while studying this Part 3. (The list follows the order of sections in this part)

***These resources are 100% optional, but might give you additional insights or provide ever deeper dives into some topics.***

👉 [Tao of React - Software Design, Architecture & Best Practices](https://alexkondov.com/tao-of-react/?ref=jonas.io)

👉 [The new wave of React state management](https://frontendmastery.com/posts/the-new-wave-of-react-state-management/?ref=jonas.io) (Excellent read!)

👉 [A Visual Guide to React Rendering - useMemo](https://alexsidorenko.com/blog/react-render-usememo?ref=jonas.io)

👉 [React as a UI Runtime](https://overreacted.io/react-as-a-ui-runtime/?ref=jonas.io) (By Dan Abramov from the React team)

👉 [You Might Not Need an Effect](https://react.dev/learn/you-might-not-need-an-effect?ref=jonas.io) (Official React docs)

👉 [A Complete Guide to useEffect](https://overreacted.io/a-complete-guide-to-useeffect/?ref=jonas.io) (By Dan Abramov)

👉 [useEffect sometimes fires before paint](https://blog.thoughtspile.tech/2021/11/15/unintentional-layout-effect/?ref=jonas.io)

👉 [Making setInterval Declarative with React Hooks](https://overreacted.io/making-setinterval-declarative-with-react-hooks/?ref=jonas.io) (By Dan Abramov)

👉 [Redux - Not Dead Yet!](https://blog.isquaredsoftware.com/2018/03/redux-not-dead-yet/?ref=jonas.io) (By Mark Erikson from the Redux team)

👉 [Why React Context is Not a "State Management" Tool](https://blog.isquaredsoftware.com/2021/01/context-redux-differences/) (By Mark Erikson)

**Library documentation:**

👉 [Vite](https://vitejs.dev/guide/why.html) (Why Vite is so fast)

👉 [CSS Modules](https://github.com/css-modules/css-modules)

👉 [React Router](https://reactrouter.com/en/main?ref=jonas.io)

👉 [React Leaflet: Installation](https://react-leaflet.js.org/docs/start-installation/?ref=jonas.io)

👉 [Redux: Style Guide](https://redux.js.org/style-guide/?ref=jonas.io) (A must-read for Redux users!)

👉 [Redux Toolkit](https://redux-toolkit.js.org/tutorials/overview?ref=jonas.io)

👉 [React Redux](https://react-redux.js.org/api/hooks?ref=jonas.io)