



React Context



What is React Context?

- Context provides a way to pass data through the component tree without having to pass props down manually at every level
- In a typical React application, data is passed top-down (parent to child) via props, but such usage can be cumbersome for certain types of props (e.g. locale preference, UI theme) that are required by many components within an application. Context provides a way to share values like these between components without having to explicitly pass a prop through every level of the tree.



When to Use Context?

- Context is designed to share data that can be considered “global” for a tree of React components, such as the current authenticated user, theme, or preferred language.



Before You Use Context

- Context is primarily used when some data needs to be accessible by *many* components at different nesting levels. Apply it sparingly because it makes component reuse more difficult.
- If you only want to avoid passing some props through many levels, [component composition](#) is often a simpler solution than context.



How it works?



How it works?

<https://reactjs.org/docs/context.html#reactcreatecontext>

React lifting state up



Lifting state up

Often, several components need to reflect the same changing data. We recommend lifting the shared state up to their closest common ancestor. Let's see how this works in action.

<https://reactjs.org/docs/lifting-state-up.html>

Further reading



Further reading

- <https://reactjs.org/docs/context.html#reactcreatecontext>
- <https://medium.com/cleverprogrammer/the-react-context-api-364da590aa73>
- <https://www.freecodecamp.org/news/react-context-for-beginners/>
- <https://reactjs.org/docs/lifting-state-up.html>

Quiz



Quiz

- <https://play.kahoot.it/v2/?quizId=45f4a3ef-0965-42fc-bc76-43023856824c>

Assignment



Assignment

- <https://contactmentor.com/react-js-practice-exercises-solution/>
- From 1 to 6