ACADEMIND

User Interaction & State

Making Apps Interactive & Reactive

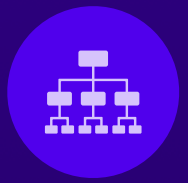
- Handling Events
- Updating the UI & Working with “State”
- A Closer Look At Components & State

Updating Data via “State”

By default, React does not care about changes of variables inside of components.

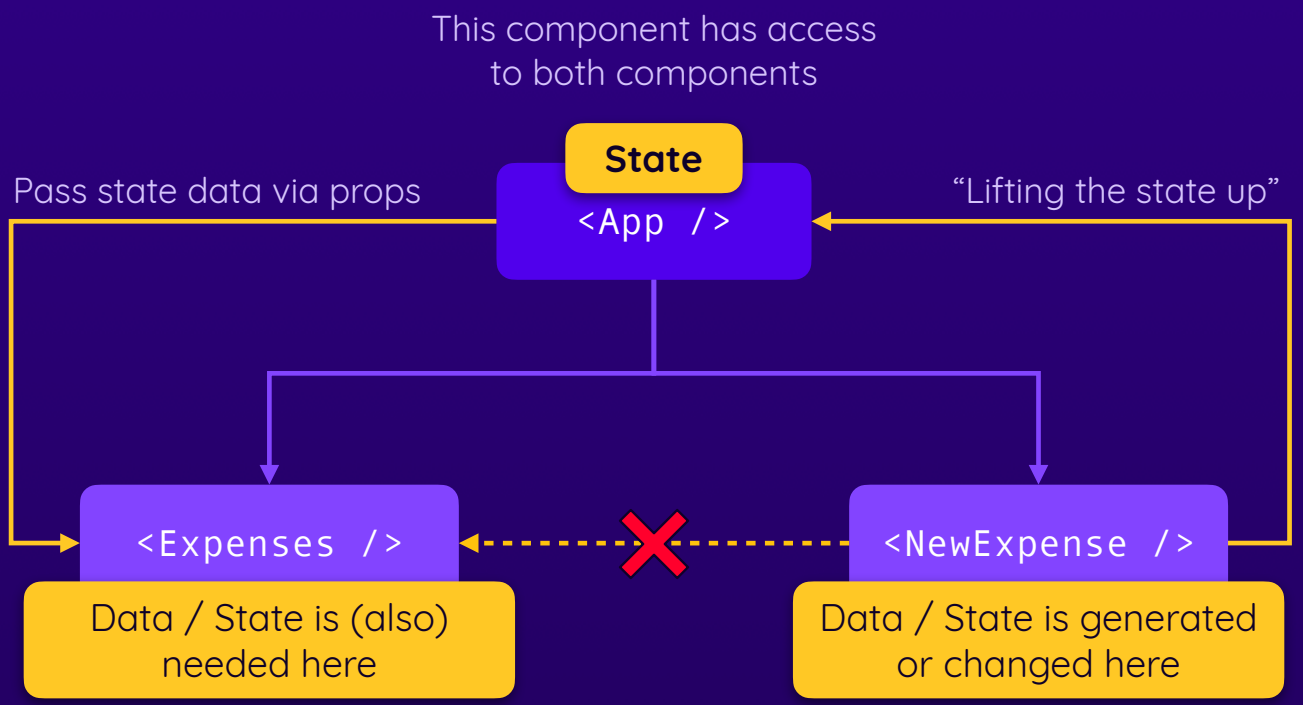
It does not re-evaluate the components’s JSX markup.

“State” is data managed by React, where changes of the data do force React to re-evaluate (“re-render”) the component where the data changed.



Child components of components where state changed are also re-evaluated.

Lifting State Up



Stateful vs Stateless Components

Stateful Components

- React components that manage internal state
- Typically, you have only a couple of these
- Also called “smart” components or “containers”

Stateless Components

- React components which only use props and output JSX
- Typically, you have plenty of these
- Also called “dumb” or “presentational” components

Alternative Way Of Building Components

Functional Components

- JavaScript functions which return JSX code
- React executes them for you (initially & upon state changes)
- Use “React Hooks” for state management

Class-based Components

- JavaScript classes as blueprints for components
- render() method for outputting JSX (called by React)
- Historically (React < 16.8), the only way of managing state!