

Difference between state and props:

State: The state represents parts of an application that can change. Each component can have its State. The state is Mutable and It is local to the component only.

Props: Props are known as properties it can be used to pass data from one component to another. Props cannot be modified, read-only, and Immutable.

Props and state are related. The state of one component will often become the props of a child component. Props are passed to the child within the render method of the parent as the second argument to `React.createElement()` or, if you're using JSX, the more familiar tag attributes.

Note:

- Use state to store the data your current page needs in your controller-view.
- Use props to pass data & event handlers down to your child components.

Key Points:

Props

- are immutable
- which lets React do fast reference checks
- are used to pass data down from your view-controller
- your top-level component
- have better performance
- use this to pass data to child components

State

- should be managed in your view-controller
- your top-level component
- is mutable
- has worse performance
- should not be accessed from child components

- pass it down with props instead

To Simply put,

State is the local state of the component which cannot be accessed and modified outside of the component. It's equivalent to local variables in a function.

Props, on the other hand, make components reusable by giving components the ability to receive data from their parent component in the form of props. They are equivalent to function parameters.

For further information:

props

Props (short for properties) are a component's configuration. They are received from above and immutable as far as the Component receiving them is concerned. A Component cannot change its props, but it is responsible for putting together the props of its child Components. Props do not have to just be data -- callback functions may be passed in as props.

state

The state is a data structure that starts with a default value when a Component mounts. It may be mutated across time, mostly as a result of user events.

A Component manages its own state internally. Besides setting an initial state, it has no business fiddling with the state of its children. You might conceptualize state as private to that component.