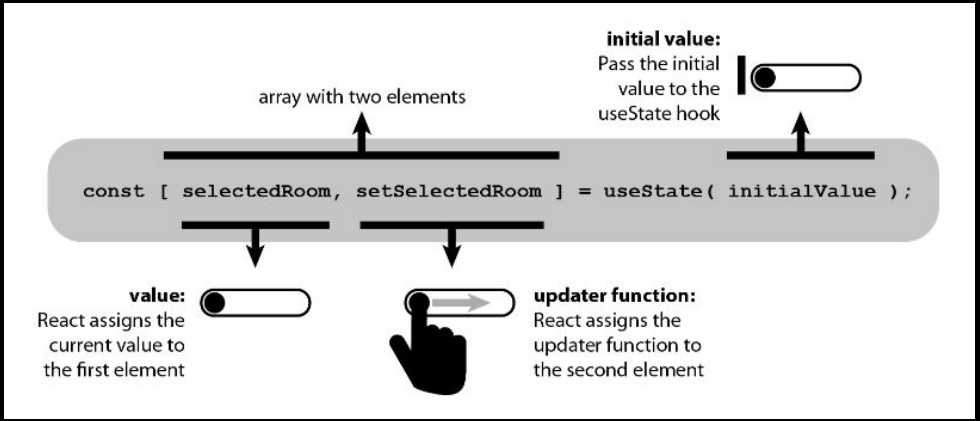
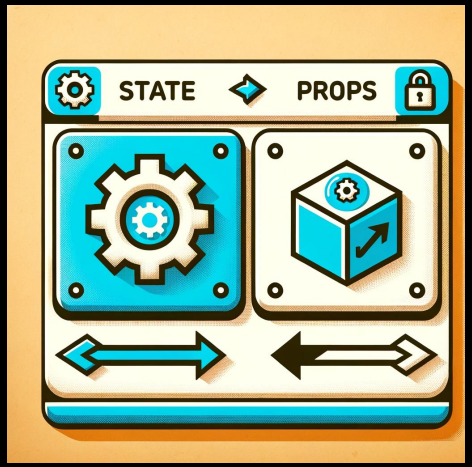
Managing State



1. State represents data that changes over time.
2. State is local and private to the component.
3. State changes cause the component to re-render.
4. For functional components, use the useState hook.
5. React Functions that start with word use are called hooks
6. Hooks should only be used inside components
7. Parent components can pass state down to children via props.
8. Lifting state up: share state between components by moving it to their closest common ancestor.

React

State vs Props



State:

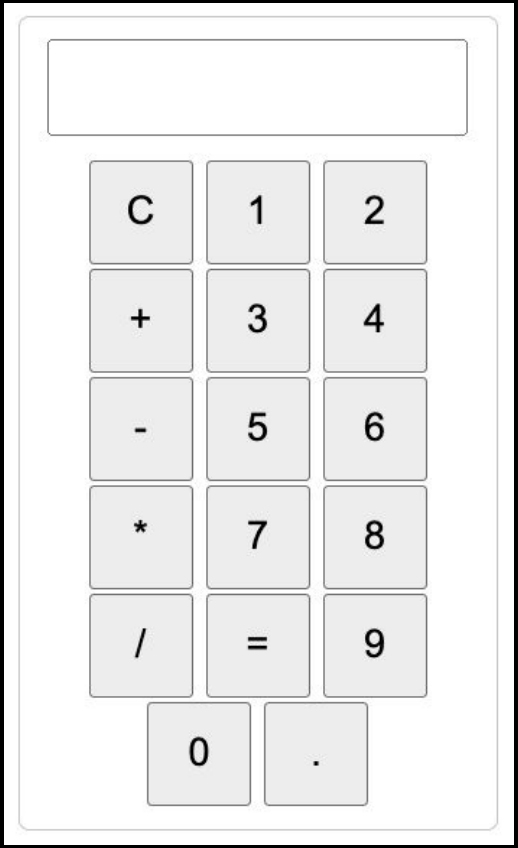
- Local and mutable data within a component.
- Initialized within the component.
- Can change over time.
- Causes re-render when updated.
- Managed using useState in functional components.

Props:

- Passed into a component from its parent.
- Read-only (immutable) within the receiving component.
- Allow parent-to-child component communication.
- Changes in props can also cause a re-render.

React

Project Calculator



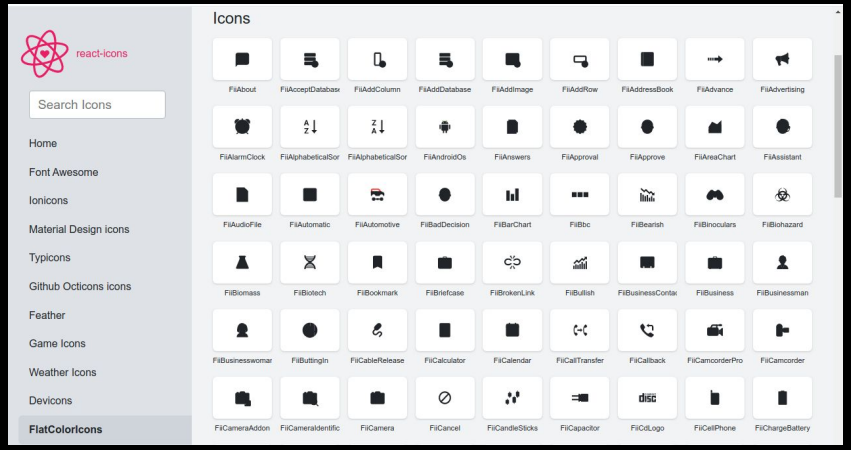
React-icon Library

1. You can use a lot of icons without managing them.
2. Install Package

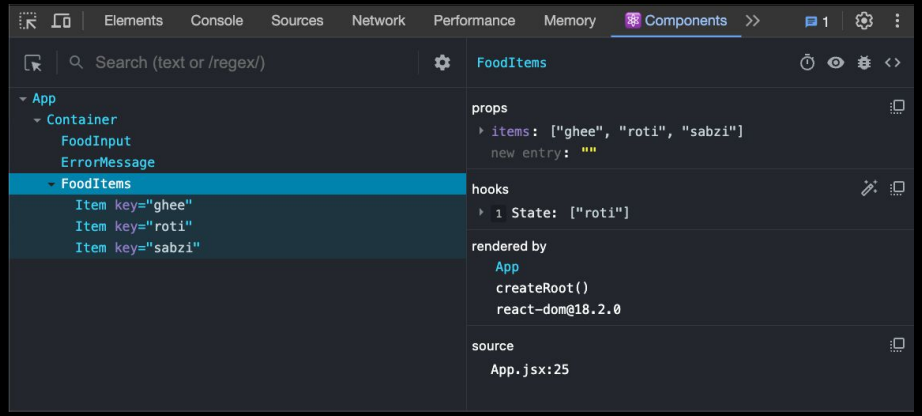
```
npm install react-icons -save
```

3. Use icon:

```
import {IconName} from
"react-icons/fc";
```



Inspecting with React Dev Tools



1. Inspection: Allows inspection of React component hierarchies.
2. State & Props: View and edit the current state and props of components.
3. Performance: Analyze component re-renders and performance bottlenecks.
4. Navigation: Conveniently navigate through the entire component tree.
5. Filtering: Filter components by name or source to locate them quickly.
6. Real-time Feedback: See live changes as you modify state or props.

How React Works

Root Component:

- The App is the main or root component of a React application.
- It's the starting point of your React component tree.

Virtual DOM:

- React creates an in-memory structure called the virtual DOM.
- Different from the actual browser DOM.
- It's a lightweight representation where each node stands for a component and its attributes.

Reconciliation Process:

- When component data changes, React updates the virtual DOM's state to mirror these changes.
- React then compares the current and previous versions of the virtual DOM.
- It identifies the specific nodes that need updating.
- Only these nodes are updated in the real browser DOM, making it efficient.