

The State of State Management in React Hooks

// Front End Small Talk 8/20/2020

__Joe Seifi__

<https://seifi.org>

1. useState

```
// define
const [count, increment] = useState(0)

// use
<main>{count}</main>

// mutate
increment(previous => previous + 1)
```

2. useContext

```
// define
const Context = createContext(0)
<Context.Provider>
  <Main />
</Context.Provider>

// use in Main
const count = useContext(Context)
<main>{count}</main>

// mutate
```



3. useReducer

```
// define
const aReducer = (state, action) => {
  if (action.type === 'INCREMENT') {
    return state + 1
  }
}

// use - in the same file
const [count, dispatch] = useReducer(aReducer, 0)
<main>{count}</main>

// mutate
dispatch({ type: "INCREMENT" });
```

4. useReducer + useContext

```
// define
const aReducer = (state, action) => {
  if (action.type === 'INCREMENT') {
    return state + 1
  }
}

const [count, dispatch] = useReducer(aReducer, 0)

const Context = createContext(0)
<Context.Provider>
  <Main count={count} dispatch={dispatch} />
</Context.Provider>

// use in Main
const count = useContext(Context)
const {dispatch, count} = props
<main>{count}</main>

// mutate in Main
dispatch({ type: "INCREMENT" });
```

5. useImmer

```
// define in Counter
import { useImmer } from "use-immer";

// use
const [count, increment] = useImmer(0);

<main>{count}</main>

// mutate
{increment(draft => draft++)}
```

<https://github.com/immerjs/use-immer>

6. Hookstate

```
// define in Counter
import { createState } from '@hookstate/core';
const state = createState(0);

// use in Main
import { useState } from '@hookstate/core';
const counter = useState(state);
<main>{counter.get()}</main>

// mutate
counter.set(c => c + 1)
```

<https://hookstate.js.org/>

Notable mentions

Rest Hooks

<https://github.com/coinbase/rest-hooks>
Data Fetching, Caching, Suspense,
Error boundaries

Zustand

<https://github.com/react-spring/zustand>
Tiny API, no context providers