## useReducer





```
function used to dispatch an action that
state and an action, and returns
will trigger a change in state

the next state

const [state, dispatch] = useReducer<T>(reducer, initialArg, initializer);

type of reducer function

initial value of state

function that receives previous
state and an action, and returns
the next state

used to initializer);

function that receives previous
state and an action, and returns
the next state

used to initializer);

function that receives previous
state and an action, and returns

(optional) function that can be
used to initialize state

type of reducer function

initial value of state
```

## Dispatch an action to update state

```
import React, { useReducer } from "react";
interface State {
                                                                                     interface for state object
  superhero: string;
type Action =
                                                                               -> type alias for the different actions that can be dispatched
   { type: "UPDATE_SUPERHERO"; payload: string }
  { type: "RESET_SUPERHERO" };
const reducer = (_state: State, action: Action): State => {
  switch (action.type) {
    case "UPDATE SUPERHERO":
       return { superhero: action.payload };
                                                                                    reducer function that returns
                                                                                     a new state value when an
    case "RESET_SUPERHERO":
                                                                                       action is dispatched
       return { superhero: "Spider-Man" };
     default:
       throw new Error();
export const FunctionComponent = () => {
                                                                                     state variable and dispatch function returned from
  const [state, dispatch] = useReducer(reducer, {
                                                                                        useReducer Hook
    superhero: "Spider-Man"
                                                                                      reducer function and an
                                                                                     initial state object passed in as arguments of the
                                                                                        useReducer Hook
  return
    <div>
                                                                               -> render field from state object
       <h2>{state.superhero}</h2>
       <br/>button
         onClick={() =>
            dispatch({
              type: "UPDATE_SUPERHERO",
                                                                                   dispatch action to update state
              payload: "Iron Man"
         Update Superhero!
       </br>
     </div>
```