immutability

Validates against mutating props, state, and other values that are immutable.

Rule Details

A component's props and state are immutable snapshots. Never mutate them directly. Instead, pass new props down, and use the setter function from `useState`.

Common Violations

Invalid

```
// ❌ Array push mutation
function Component() {
  const [items, setItems] = useState([1, 2, 3]);

  const addItem = () => {
    items.push(4); // Mutating!
    setItems(items); // Same reference, no re-render
  };
}

// ❌ Object property assignment
function Component() {
  const [user, setUser] = useState({name: 'Alice'});

  const updateName = () => {
    user.name = 'Bob'; // Mutating!
    setUser(user); // Same reference
  };
}

// ❌ Sort without spreading
```

```
function Component() {
  const [items, setItems] = useState([3, 1, 2]);

  const sortItems = () => {
    setItems(items.sort()); // sort mutates!
  };
}
```

Valid

```
// ✅ Create new array
function Component() {
  const [items, setItems] = useState([1, 2, 3]);

  const addItem = () => {
    setItems([...items, 4]); // New array
  };
}

// ✅ Create new object
function Component() {
  const [user, setUser] = useState({name: 'Alice'});

  const updateName = () => {
    setUser({...user, name: 'Bob'}); // New object
  };
}
```

Troubleshooting

I need to add items to an array

Mutating arrays with methods like `push()` won't trigger re-renders:

```
// ❌ Wrong: Mutating the array
function TodoList() {
  const [todos, setTodos] = useState([]);
```

```
  const addTodo = (id, text) => {
    todos.push({id, text});
    setTodos(todos); // Same array reference!
  };

  return (
    <ul>
      {todos.map(todo => <li key={todo.id}>{todo.text}</li>)}
    </ul>
  );
}
```

Create a new array instead:

```
// ✅ Better: Create a new array
function TodoList() {
  const [todos, setTodos] = useState([]);

  const addTodo = (id, text) => {
    setTodos([...todos, {id, text}]);
    // Or: setTodos(todos => [...todos, {id: Date.now(), text}])
  };

  return (
    <ul>
      {todos.map(todo => <li key={todo.id}>{todo.text}</li>)}
    </ul>
  );
}
```

I need to update nested objects

Mutating nested properties doesn't trigger re-renders:

```
// ❌ Wrong: Mutating nested object
function UserProfile() {
  const [user, setUser] = useState({
    name: 'Alice',
```

```
    settings: {
      theme: 'light',
      notifications: true
    }
  });

  const toggleTheme = () => {
    user.settings.theme = 'dark'; // Mutation!
    setUser(user); // Same object reference
  };
}
```

Spread at each level that needs updating:

```
// ✅ Better: Create new objects at each level
function UserProfile() {
  const [user, setUser] = useState({
    name: 'Alice',
    settings: {
      theme: 'light',
      notifications: true
    }
  });

  const toggleTheme = () => {
    setUser({
      ...user,
      settings: {
        ...user.settings,
        theme: 'dark'
      }
    });
  };
}
```

이전
globals

다음
incompatible-library

uwu?