7 React

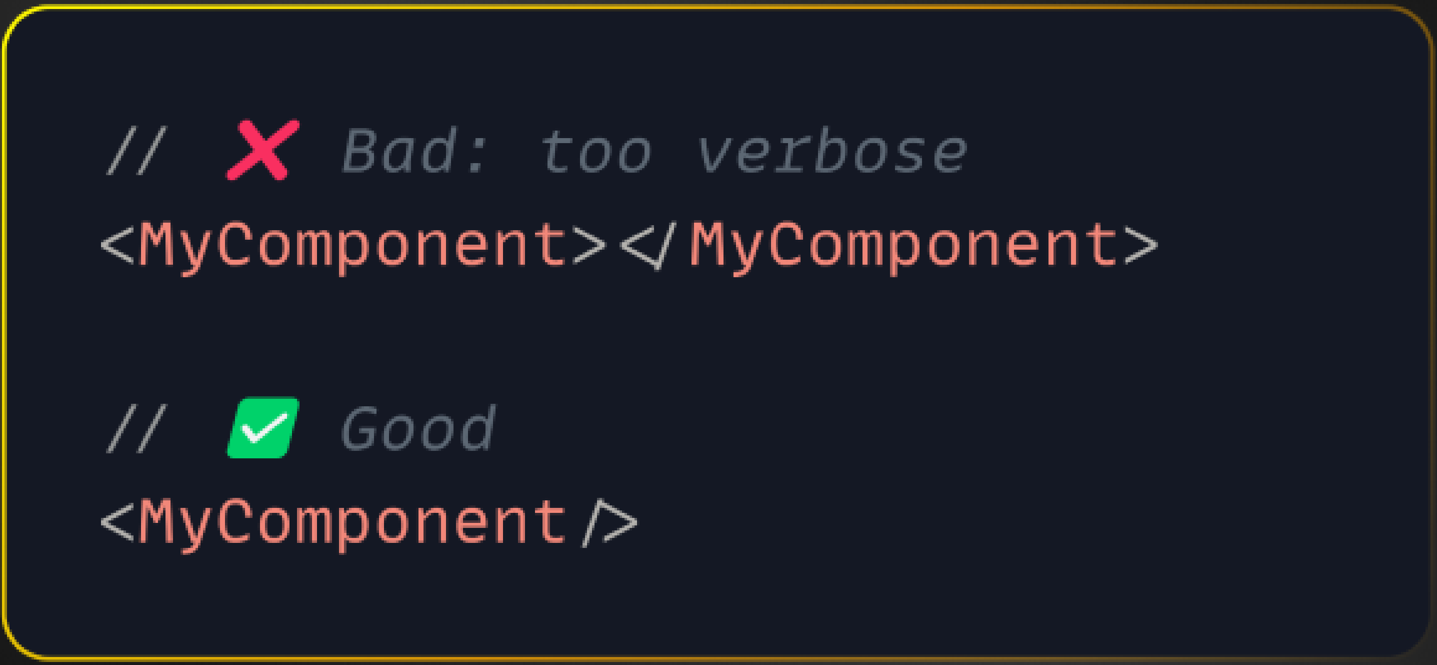
TRICKS FOR BEGINNERS

Halim Shams

01

Use Self-Closing Tags

Keep your code clean by using self-closing tags.



```
// ❌ Bad: too verbose
<MyComponent></MyComponent>

// ✅ Good
<MyComponent />
```

Halim Shams

Prefer Fragments

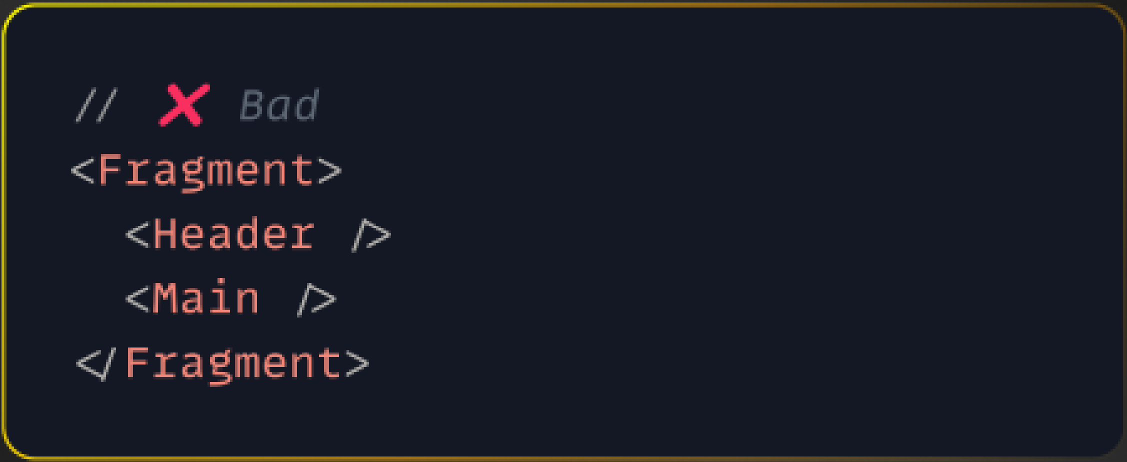
Avoid unnecessary `<div>` tags. Use `<Fragment>` to keep the DOM clean.

```
// ❌ Bad
<div>
  <Header />
  <Main />
</div>
```

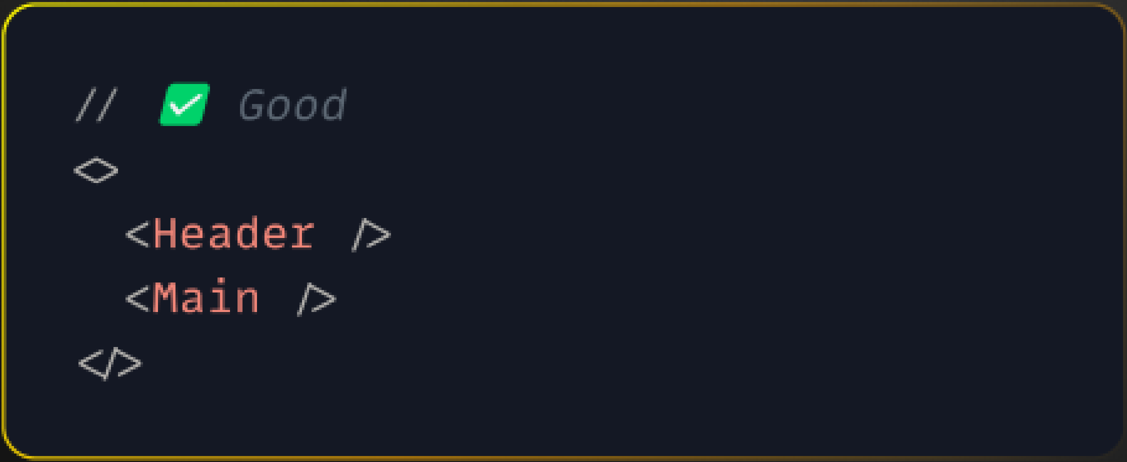
```
// ✅ Good
<Fragment key={id}>
  <Header />
  <Main />
</Fragment>
```

Use Fragment Shorthand

Leverage Fragment shorthand if you don’t need to pass key.



```
// ❌ Bad
<Fragment>
  <Header />
  <Main />
</Fragment>
```

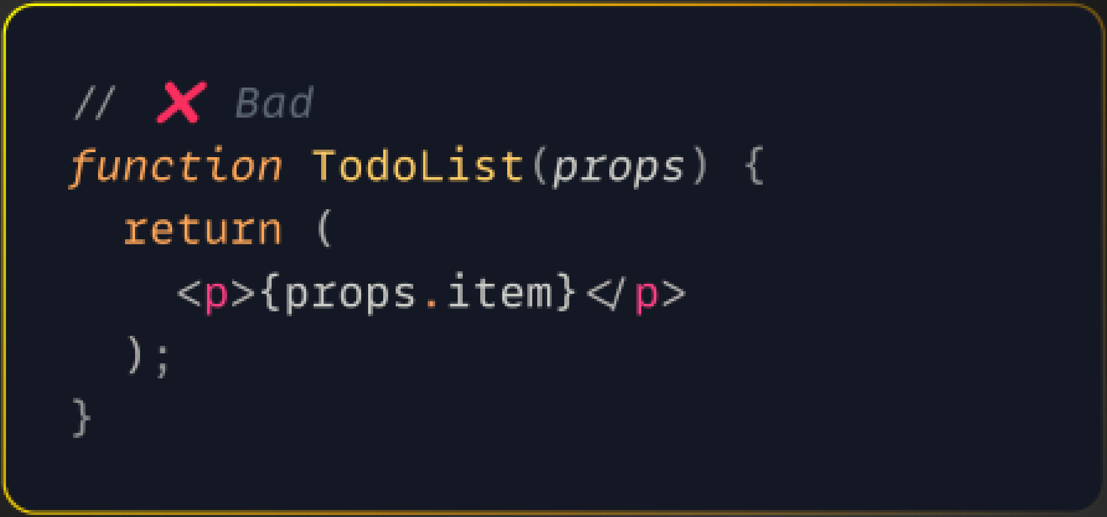


```
// ✅ Good
<>
  <Header />
  <Main />
</>
```

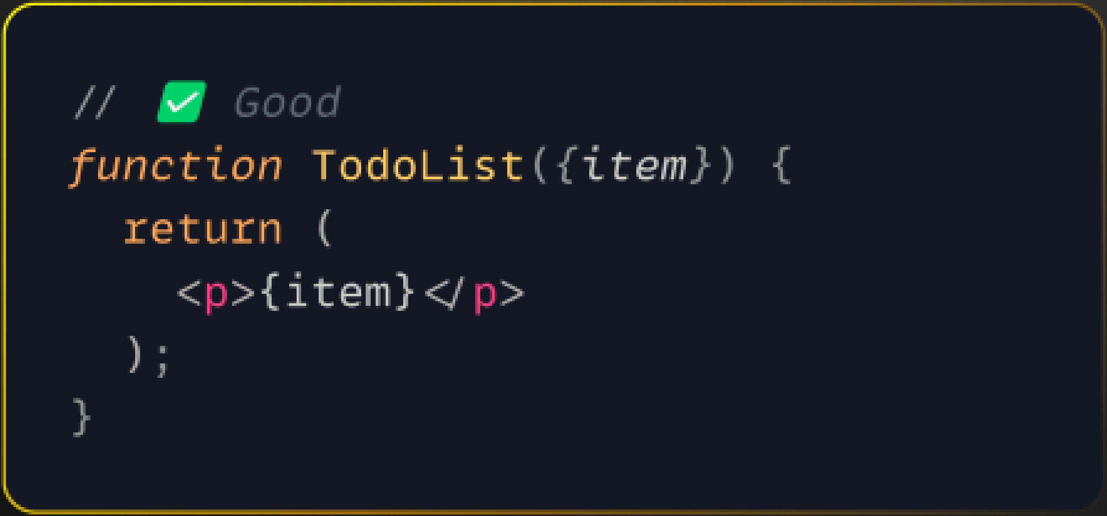
in
Halim Shams

Spread Props

Destructure props for more readable and concise components.



```
// ❌ Bad
function TodoList(props) {
  return (
    <p>{props.item}</p>
  );
}
```

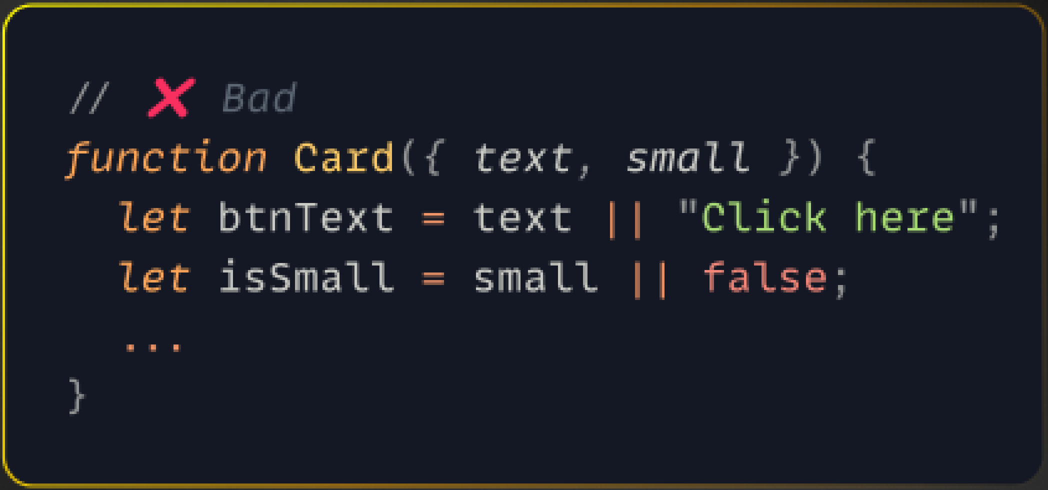


```
// ✅ Good
function TodoList({item}) {
  return (
    <p>{item}</p>
  );
}
```

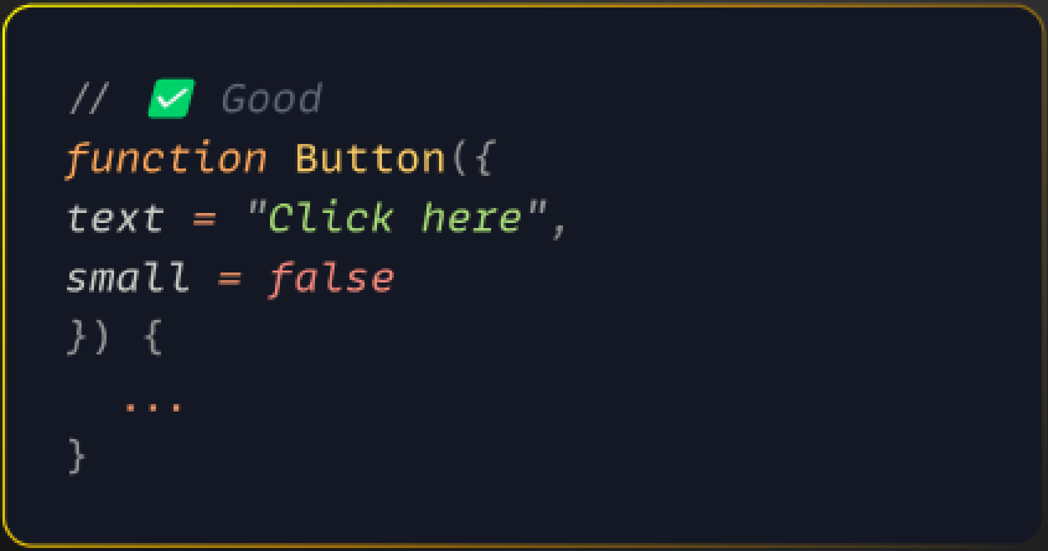
Halim Shams

Setting Default Props

Define default values directly within props for easy management.



```
// ❌ Bad
function Card({ text, small }) {
  let btnText = text || "Click here";
  let isSmall = small || false;
  ...
}
```

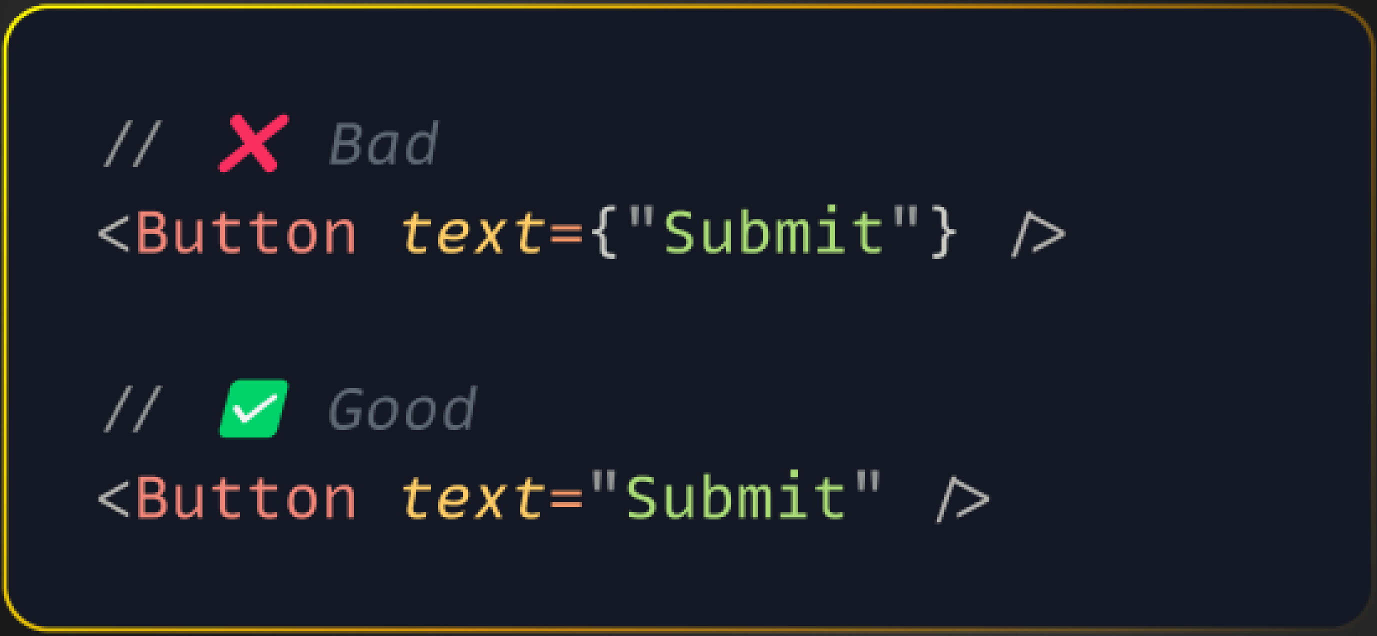


```
// ✅ Good
function Button({
text = "Click here",
small = false
}) {
  ...
}
```

Halim Shams

Simplify String Props

Pass string props without curly braces for cleaner syntax.



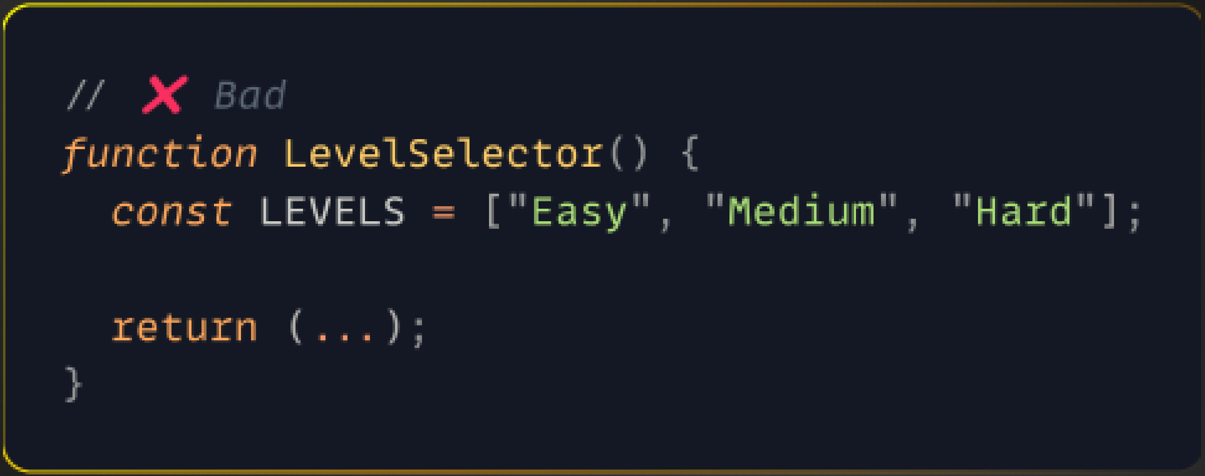
```
// ❌ Bad
<Button text={"Submit"} />

// ✅ Good
<Button text="Submit" />
```

Halim Shams

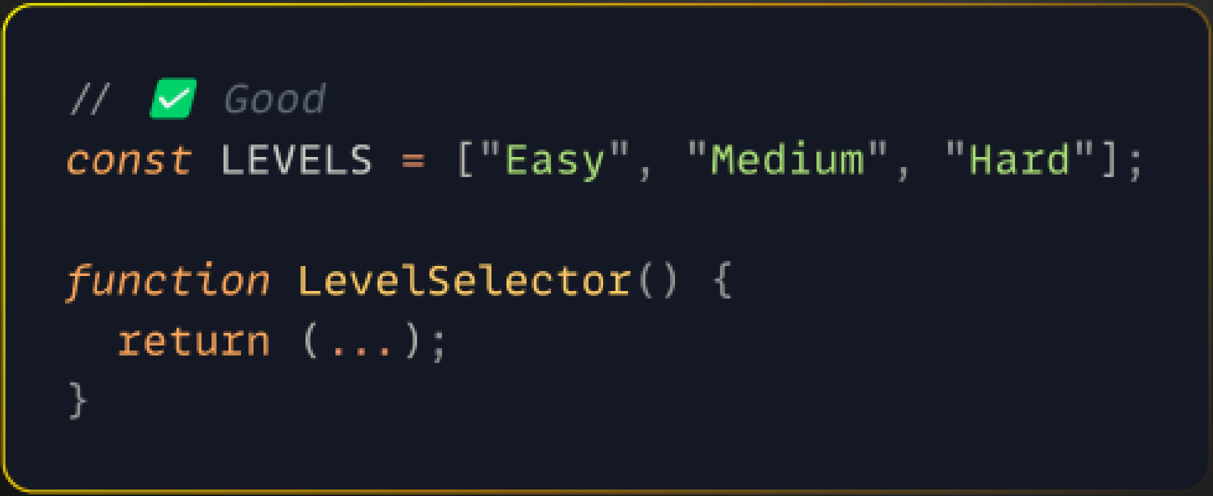
Move Static Data Out

Keep components efficient by moving static data outside.



```
// ❌ Bad
function LevelSelector() {
  const LEVELS = ["Easy", "Medium", "Hard"];

  return (...);
}
```



```
// ✅ Good
const LEVELS = ["Easy", "Medium", "Hard"];

function LevelSelector() {
  return (...);
}
```

Halim Shams

Follow for more

Halim Shams