

## ADVANTAGES OF USING HOOKS OVER CASS:

Using functional components and hooks in React.js has several advantages over class components:

1. **Simplicity and readability:** Functional components are typically shorter and easier to read compared to class components. They focus only on rendering and behavior, without the need for managing complex lifecycle methods or dealing with **this** binding.
2. **Improved performance:** Functional components are generally more performant than class components. They don't have the overhead of creating and managing instances of class components, resulting in faster rendering and updates.
3. **Code reusability:** Functional components promote code reusability through the use of custom hooks. Hooks allow you to extract and reuse stateful logic, making it easier to share and compose logic across components.
4. **Easier testing:** Functional components are easier to test because they are essentially just JavaScript functions. You can test them by passing props and mocking dependencies without the need for complex setup or dealing with the component instance.
5. **Better compatibility with React ecosystem:** As React evolves, the React team is focusing more on functional components and hooks. Many new features and improvements are being introduced specifically for functional components, making it the recommended approach for writing React components.
6. **Improved developer experience:** Hooks, introduced in React 16.8, make it easier to manage state and side effects in functional components. They provide a simpler and more intuitive way to handle component logic, reducing the cognitive load and making development more enjoyable.

While there are some cases where class components are still necessary, functional components and hooks provide a more modern and efficient way to build React applications, offering benefits such as simplicity, performance, reusability, testability, compatibility, and a better developer experience.