

Documentation Tool: React(Javascript/Jsx)

Introduction:

React is an open-source JavaScript library developed by Facebook for building user interfaces, particularly for single-page applications. It allows developers to create reusable UI components and manage the view layer efficiently.

React uses **JSX (JavaScript XML)** for building components with a syntax similar to HTML and supports a **component-based architecture**, making it a top choice for frontend development.

Why Use React?

- Component-Based: Break down UI into independent, reusable pieces.
- Fast Rendering: Uses a virtual DOM for efficient updates.
- Unidirectional Data Flow: Easier to debug and control.
- Declarative Syntax: Write UI by describing the final state.
- Rich Ecosystem: Strong community, tools, and libraries.
- Popular Tools: Storybook, React Dev Tools, Create React App, etc

It's widely used in modern frontend development for building scalable and maintainable applications.

Documentation:

Official Docs: <https://react.dev/learn>

Storybook Docs: <https://storybook.js.org/docs/react/get-started/introduction>

Installation & Usage:

1. Installing React

Using Create React App (Simple Setup):

```
bash
```

```
npx create-react-app my-app  
cd my-app  
npm start
```

Documenting React Components:

React components can be documented using Storybook or other tools.

2. Using Storybook.

Install Storybook:

```
bash
```

```
npx storybook init
```

Creating Stories:

Create a .stories.jsx or .stories.tsx file for each component:

```
jsx
```

```
import MyComponent from './MyComponent';  
  
export default {  
  title: 'MyComponent',  
  component: MyComponent,  
};  
  
export const Default = () => <MyComponent />;
```

Run storybook:

```
bash
```

```
npm run storybook
```

Deployment

You can deploy React documentation (e.g., Storybook) to platforms like:

- Firebase Hosting.

```
bash  
  
npm run build-storybook  
firebase init  
firebase deploy
```

- Vercel.
- Github pages.

Pros & Cons of React:

Why React?

- Reusable, testable components
- Efficient updates with Virtual DOM
- Rich ecosystem (Next.js, Redux, etc.)
- Large community and extensive support
- Easy integration with Storybook for UI documentation

Why Not React?

- Not a full framework (needs routing & state libraries)
- Learning curve for beginners
- Overkill for very simple static sites

Why Use StoryBook for React Docs?

- View components in isolation
- Auto-generates documentation from component metadata
- Addons for accessibility, viewport testing, etc.
- Perfect for building and testing design systems and component libraries

StoryBook Website: <https://storybook.js.org>