

Hoja de Atajos de React

Tu referencia rápida para iniciar y trabajar con React.

Iniciar un Proyecto

Usando Create React App (Recomendado)

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

Usando Vite

```
npm create vite@latest my-app -- --template react  
cd my-app  
npm install  
npm run dev
```

Usar React desde un CDN

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <script src="https://unpkg.com/react@17/umd/react.development.js"></script>  
  <script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>  
</head>  
<body>  
  <div id="root"></div>  
  <script>  
    const App = () => React.createElement('h1', null, 'Hello React!');  
    ReactDOM.render(  
      React.createElement(App),  
      document.getElementById('root')  
    );  
  </script>  
</body>  
</html>
```

JSX y Renderizado

JSX Básico

```
const element = <h1>Hello, world!</h1>;  
ReactDOM.render(element, document.getElementById('root'));
```

Componentes Funcionales

```
function Welcome(props) {  
  return <h1>Hello, {props.name}!</h1>  
}  
const element = <Welcome name="React" />;  
ReactDOM.render(element, document.getElementById('root'));
```

Componentes con Hooks

```
import React, { useState } from 'react';  
  
function Counter() {  
  const [count, setCount] = useState(0);  
  return (  
    <div>  
      <p>Count: {count}</p>  
      <button onClick={() => setCount(count + 1)}>  
        Increment  
      </button>  
    </div>  
  );  
}
```

Props y Estado

Pasar Props

```
function App() {  
  return <Greeting name="John" />;  
}  
  
function Greeting(props) {  
  return <h1>Hello, {props.name}!</h1>;  
}
```

Estado con useState

```
import React, { useState } from 'react';

function App() {
  const [message, setMessage] = useState('Hello!');
  return (
    <div>
      <p>{message}</p>
      <button onClick={() => setMessage('Hi!')}>
        Change Message
      </button>
    </div>
  );
}
```

Manejo de Eventos

Eventos en React

```
function App() {
  function handleClick() {
    alert('Button clicked!');
  }
  return <button onClick={handleClick}>Click Me</button>;
}
```

Eventos Sintéticos

```
function App() {
  function handleInputChange(event) {
    console.log(event.target.value);
  }
  return <input type="text" onChange={handleInputChange} />;
}
```

Renderizado Condicional

If/Else

```
function App({ isLoggedIn }) {
  if (isLoggedIn) {
    return <h1>Welcome back!</h1>;
  }
}
```

```
    }  
    return <h1>Please log in.</h1>;  
  }  
}
```

Operador Ternario

```
function App({ isLoggedIn }) {  
  return <h1>{isLoggedIn ? 'Welcome back!' : 'Please log in.'}</h1>;  
}
```

Listas y Keys

Renderizar una Lista

```
function App() {  
  const numbers = [1, 2, 3, 4, 5];  
  return (  
    <ul>  
      {numbers.map((number) => (  
        <li key={number}>{number}</li>  
      ))}  
    </ul>  
  );  
}
```

Estilos

Inline Styles

```
function App() {  
  const style = { color: 'blue', fontSize: '20px' };  
  return <h1 style={style}>Hello, styled React!</h1>;  
}
```

CSS Modules

```
/* App.module.css */  
.title {  
  color: red;  
}
```

```
““javascript // En el componente: import styles from './App.module.css';  
function App() return <h1 className=styles.title>Hello, CSS Modules!</h1>;
```

React Router

Configuración Básica

```
\begin{minted}{javascript} import { BrowserRouter as Router, Route, Routes } from  
  ↳ 'react-router-dom';  
  
function App() { return ( <Router> <Routes> <Route path="/" element={<Home />} /> <Route  
  ↳ path="/about" element={<About />} /> </Routes> </Router> ); }  
  
function Home() { return <h1>Home Page</h1>; }
```

Manejo de Estado Global

Context API

```
function App() { const [value, setValue] = useState('Hello Context!'); return (  
  ↳ <MyContext.Provider value={{ value, setValue }}> <Child />  
  ↳ </MyContext.Provider> ); }
```

Redux (opcional)

```
\begin{minted}{javascript} import { createStore } from 'redux'; import { Provider,  
  ↳ useDispatch, useSelector } from 'react-redux';  
  
const reducer = (state = { count: 0 }, action) => { switch (action.type) { case  
  ↳ 'INCREMENT': return { count: state.count + 1 }; default: return state; } };  
  
const store = createStore(reducer);  
  
function App() { return ( <Provider store={store}> <Counter /> </Provider> ); }  
  
function Counter() { const count = useSelector((state) => state.count); const dispatch =  
  ↳ useDispatch();
```

Sitios útiles

- [React Docs](#)
- [Vite](#)
- [React Router](#)