

React Cheat Sheet

Iniciar un proyecto de React con Vite

1. **Instalar Vite**

- Si no tienes Node.js, instálalo primero desde <https://nodejs.org>.
- Usar el siguiente comando para crear un proyecto de React con Vite:

```
```bash
npm create vite@latest nombre-del-proyecto --template react
```
```

2. **Instalar dependencias**

- Entra en la carpeta del proyecto:

```
```bash
cd nombre-del-proyecto
```
```

- Instala las dependencias necesarias con:

```
```bash
npm install
```
```

3. **Iniciar el servidor de desarrollo**

- Para iniciar el servidor de desarrollo, ejecuta:

```
```bash  

npm run dev

```
```

4. ****Abrir el proyecto en el navegador****

- Abre el navegador y accede a `http://localhost:5173/` para ver el proyecto en ejecución.

Componentes

- ****Componente funcional****

```
```jsx  

const MyComponent = () => {

 return <div>Hello, World!</div>;

};

```
```

- ****Componente de clase****

```
```jsx  

class MyComponent extends React.Component {

 render() {

 return <div>Hello, World!</div>;

 }

}

```
```

Props y State

- **Props (propiedades)**

```
```jsx

const MyComponent = (props) => {

 return <div>{props.name}</div>;

};

...

```

## - **State (estado)**

```
```jsx

const MyComponent = () => {

  const [count, setCount] = React.useState(0);

  return <div onClick={() => setCount(count + 1)}>{count}</div>;

};

...

```

Eventos

- **Evento onClick**

```
```jsx

const handleClick = () => {

 alert('Clicked!');

};

return <button onClick={handleClick}>Click me</button>;

...

```

## ### useEffect (Efectos secundarios)

### - **Ejemplo básico**

```
```jsx

React.useEffect(() => {

```

```
    console.log('Component mounted or updated');  
  }, []); // Se ejecuta una vez al montar el componente  
  ...
```

useState (Manejo del estado)

- **Ejemplo básico**

```
``jsx  
  
const [state, setState] = React.useState(initialState);  
...
```

useContext (Contexto)

- **Crear contexto**

```
``jsx  
  
const MyContext = React.createContext();  
...
```

- **Proveer valor**

```
``jsx  
  
<MyContext.Provider value={someValue}>  
  <MyComponent />  
</MyContext.Provider>  
...
```

- **Consumir contexto**

```
``jsx  
  
const value = React.useContext(MyContext);  
...
```

Conditional Rendering

- ****if-else****

```
```jsx
if (isLoggedIn) {
 return <div>Welcome!</div>;
} else {
 return <div>Please log in</div>;
}
```
```

- ****Operador ternario****

```
```jsx
return isLoggedIn ? <div>Welcome!</div> : <div>Please log in</div>;
```
```

Listas y Keys

- ****Renderizar lista****

```
```jsx
const list = ['apple', 'banana', 'orange'];
return (

 {list.map((item, index) => (
 <li key={index}>{item}
))}

);
```
```

Ref y DOM

- ****Crear un ref****

```
```jsx

const inputRef = React.useRef();

const handleFocus = () => {

 inputRef.current.focus();

};

return (

 <div>

 <input ref={inputRef} />

 <button onClick={handleFocus}>Focus Input</button>

 </div>

);

...`
```

### ### Hooks personalizados

- **\*\*Ejemplo de hook personalizado\*\***

```
```jsx

function useCounter(initialValue) {

  const [count, setCount] = React.useState(initialValue);

  const increment = () => setCount(count + 1);

  const decrement = () => setCount(count - 1);

  return { count, increment, decrement };

}

...`
```

JSX

- ****Sintaxis****

```
```jsx
```

```
const element = <h1>Hello, world!</h1>;
```

```
```
```

React Router (Rutas)

- **Instalar React Router**

```
```bash
```

```
npm install react-router-dom
```

```
```
```

- **Definir rutas**

```
```jsx
```

```
import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';
```

```
const App = () => (
```

```
 <Router>
```

```
 <Switch>
```

```
 <Route path="/home" component={Home} />
```

```
 <Route path="/about" component={About} />
```

```
 </Switch>
```

```
 </Router>
```

```
);
```

```
```
```

Formulario

- **Manejo de formulario controlado**

```
```jsx
```

```
const [value, setValue] = React.useState("");
```

```
const handleChange = (e) => setValue(e.target.value);
```

```
return <input type="text" value={value} onChange={handleChange} />;
```

```
...
```