

React Functions and Features Reference

Table 1: Predefined vs Custom Functions

Functionality / Code	Description	Predefined	You Need to Write
useState()	Manage component state	Yes	Only names & value
useEffect()	Run side-effects	Yes	Only callback & logic
axios.get()	Fetch data from API	Yes	URL & logic
.then()	Handle promise success	Yes	Response logic
.catch()	Handle promise failure	Yes	Error logic
console.log()	Print to console	Yes	What to print
e.preventDefault()	Stop default action	Yes	Where to call
onChange	Input change handler	Yes	Function & binding
onSubmit	Form submit handler	Yes	Handler logic
map()	Loop through arrays	Yes	JSX in loop
return()	Render JSX	Yes	Custom JSX
props	Receive data from parent	Yes	Define structure
state	Reactive value	Yes	Define/update logic
component	Reusable block	Yes	You create it

React Functions and Features Reference

Table 2: Function Limitations and Usage

Function / Feature	Where to Use	Parameters	Returns / Effect	Limitations / Notes
useState()	Functional components	Initial value	Array [state, setter]	Not in loops/conditions
useEffect()	Functional components	Callback + deps	Runs after render	Don't return promise directly
axios.get()	Anywhere (e.g. useEffect)	URL	Promise	Handle .then/.catch or use async
.then()	After promise	Callback	On success	Only works with promises
.catch()	After promise	Callback	On error	Use with then or await
console.log()	Anywhere	Any value	Print to console	Dev use only
e.preventDefault()	Event handlers	event	Prevents reload	Use in forms/links
onChange	JSX input elements	Handler function	Calls on input change	Use with controlled inputs
onSubmit	JSX form	Handler function	Calls on submit	Use e.preventDefault()
map()	In render/JSX	Callback	New array/JSX	Must return element
return()	Component body	JSX	Rendered output	One root element only
props	Inside component	Parent data	Access value	Read-only
state	With useState()	Internal	Holds dynamic value	Update via setter
Custom Functions	Anywhere	Any	Your logic	Keep pure inside component