# React Hooks Summary (Hindi Edition)

## 1. Basic Hooks

### 1️⃣ useState

- **काम:** Component की internal state manage करने के लिए
- **Purpose:** Manage dynamic data (counter, input, toggle)
- **Example:** Counter, Form

### 2️⃣ useEffect

- **काम:** Component render/update पर side effects चलाने के लिए
- **Purpose:** Run code after render (API calls, event listeners)
- **Example:** API call, title update

### 3️⃣ useRef

- **काम:** DOM elements या previous value को hold करने के लिए (re-render के बिना)
- **Purpose:** Access DOM directly or store mutable value
- **Example:** Focus input, Timer

### 4️⃣ useContext

- **काम:** Global data को share करने के लिए बिना props pass किए
- **Purpose:** Share data across components
- **Example:** Theme / User data

## 2. Advanced Hooks

### 5️⃣ useMemo

- **काम:** Expensive calculations को optimize करने के लिए
- **Purpose:** Memoize computed values
- **Example:** Filtered/Sorted List

### 6️⃣ useCallback

- **काम:** Functions को re-render पर दोबारा create होने से रोकने के लिए
- **Purpose:** Memoize callback functions
- **Example:** Parent-child optimization

### 7️⃣ useReducer

- **काम:** Complex state logic manage करने के लिए (Redux जैसा)
- **Purpose:** Manage multiple state transitions
- **Example:** Todo app, Counter

## 3. Special Hooks

### 8️⃣ useLayoutEffect

- **काम:** DOM update से पहले synchronous layout changes करने के लिए
- **Purpose:** Measure DOM or sync animations
- **Example:** Box size, animation

### 9️⃣ useImperativeHandle

- **काम:** Child component के methods को parent में expose करने के लिए
- **Purpose:** Customize ref behavior
- **Example:** Custom input focus

### 🔟 useDebugValue

- **काम:** Custom hooks debug info React DevTools में दिखाने के लिए
- **Purpose:** Add debug info
- **Example:** Online/offline status

## 4. Custom Hooks

- **काम:** Reusable logic create करने के लिए
- **Purpose:** Combine multiple hooks into one reusable function
- **Examples:** useFetch, useToggle, useLocalStorage, useWindowWidth, usePrevious

## 5. React 18+ Hooks

| Hook | Purpose | Example |
|---|---|---|
| useId | Unique ID generate करने के लिए | Dynamic forms |
| useDeferredValue | Slow UI को smooth करने के लिए | Search results lag fix |
| useTransition | State updates को low priority देने के लिए | Slow filters |
| useSyncExternalStore | External store (Redux-like) से subscribe करने के लिए | Zustand / Redux |
| useInsertionEffect | Styles inject करने से पहले run होता है | Styled components |

## Quick Memory Trick

| Category | Hooks | Easy Hint |
|---|---|---|
| Basic | useState, useEffect | State और Side Effect |
| Intermediate | useRef, useContext | Reference और Context |
| Advanced | useMemo, useCallback, useReducer | Optimization और State Logic |
| Special | useLayoutEffect, useImperativeHandle, useDebugValue | DOM Control और Custom Debug |

| Category | Hooks | Easy Hint |
| --- | --- | --- |
| Custom | useToggle, useFetch, useLocalStorage | Reusable Logic |

## Practical Real-Life Mapping

| Use Case | Recommended Hook(s) |
| --- | --- |
| Counter / Form | useState |
| Fetch API Data | useEffect + useState / useFetch |
| Light / Dark Theme | useContext + useState |
| Performance Optimization | useMemo + useCallback |
| Complex State Logic | useReducer |
| DOM Manipulation | useRef + useLayoutEffect |
| Custom Component Control | useImperativeHandle |
| Persistent Data | useLocalStorage (custom hook) |
| Debugging | useDebugValue |

---

End of Handbook