React useEffect Hook – Beginner to Pro (Teaching Guide)

0) Mental Model

```
useEffect = "render ke baad ka kaam" (side-effects).
UI draw → phir effect chalega. Render ko block nahi karta.
- Render = JSX banana
- Effect = data fetch, event listener, timers, subscriptions, logging, DOM focus, etc.
Syntax:
useEffect(() => {
  // side effect code
}, [dependencies]);
```

1) Three Dependency Modes

```
1. Mount only:
useEffect(() => { console.log("mounted once"); }, []);

2. On change:
useEffect(() => { console.log("query changed:", query); }, [query]);

3. Every render (avoid for beginners):
useEffect(() => { console.log("runs every render"); });
```

2) Cleanup (Important)

```
Cleanup listeners/timers/subscriptions in return function:
useEffect(() => {
  const onResize = () => setW(window.innerWidth);
  window.addEventListener("resize", onResize);
  return () => window.removeEventListener("resize", onResize);
}, []);
```

3) Async Fetch – Safe Pattern

```
useEffect(() => {
  const ac = new AbortController();
  (async () => {
    try {
      const res = await fetch(`/api/users?q=${query}`, { signal: ac.signal });
      const data = await res.json();
      setUsers(data);
    } catch (e) {
      if (e.name !== "AbortError") console.error(e);
    }
  })();
  return () => ac.abort();
}, [query]);
```

4) Common Pitfalls

```
- Missing dependencies
- Using useEffect for derived state
- No cleanup → memory leaks
- Dev Strict Mode double-run
```

5) useEffect vs Other Hooks

```
- useEffect: render ke baad side-effects
- useLayoutEffect: DOM paint se pehle
- useMemo: computation cache
- useCallback: function identity stable
```

6) Progressive Examples

```
A) Auto-focus input:
const ref = useRef(null);
useEffect(() => { ref.current?.focus(); }, []);

B) Debounce search:
useEffect(() => {
  const t = setTimeout(() => setDebounced(q), 400);
  return () => clearTimeout(t);
}, [q]);

C) WebSocket ticker:
useEffect(() => {
  const ws = new WebSocket("wss://example.com/tick");
  ws.onmessage = e => setPrice(JSON.parse(e.data).p);
  return () => ws.close();
}, []);
```

7) Teaching Plan (90m)

```
Part 1: Mental model + dependency modes (15m)
Part 2: Async fetch (25m)
Part 3: Cleanup patterns (20m)
Part 4: Pitfalls + strict mode demo (15m)
Part 5: Exercises (15m)
```

8) Exercises

```
1. Mount logger
2. Timer button
3. Query fetch with debounce
4. Online status listener
5. Scroll spy menu highlight
```

9) Mini-project: Real-time Cart

```
Products list → add to cart → total auto update.
Effects:
- Mount pe /products fetch
- Cart change pe total recompute
- Checkout pe POST + WS cleanup
```

10) Cheat Sheet

```
- render → effect → cleanup → render
- [] = once; [x] = on x change; no deps = every render
- Always cleanup
- AbortController for fetch
- Don't compute derived state in effect
- Keep effects small; split concerns
```