**JavaScript - Insane Level**

1. **(c)** [1, 2, 3, 4] [1, 2, 3, 4] [1, 2, 3]
   * b = a; → b and a reference the same array. So, b.push(4); modifies a as well.
   * c = [...a]; → c is a shallow copy of a, so c.pop(); only affects c.
2. **(b)** "object"
   * arguments is an object inside functions, so typeof arguments returns "object".
3. **(c)** function closure() { let secret = 42; return () => secret; }
   * Closures allow private variables in JavaScript.
4. **(a)** true
   * ![] is false, so [] == false is true due to type coercion.
5. **(a)** Uses IEEE 754 double-precision floating-point
   * JavaScript represents numbers using IEEE 754 (64-bit floating point).

**HTML - Impossible Level**

1. **(c)** The browser will treat it as invalid but still render
   * Multiple <main> elements are invalid, but browsers still render the page.
2. **(a)** Allows styling elements without affecting global CSS
   * Shadow DOM isolates styles from the global CSS.
3. **(d)** It cannot be styled with CSS
   * <canvas> is just a drawing surface; its contents cannot be styled with CSS.
4. **(a)** Defines a search field (obsolete)
   * <isindex> was used for search input but is now obsolete.
5. **(a)** Prevents cross-origin requests
   * The sandbox attribute restricts iframe content, including cross-origin requests.

**CSS - Insane Level**

1. **(a)** 100px
   * box-sizing: border-box; includes padding and border inside height.
2. **(a)** Creates a color inversion effect
   * mix-blend-mode: difference; inverts colors based on the background.
3. **(a)** Prevents rendering of offscreen elements
   * content-visibility: auto; skips rendering for elements outside the viewport.
4. **(a)** Restricts layout calculations
   * contain: layout; prevents layout recalculations from affecting other elements.
5. **(d)** outside-flex
   * There is no display: outside-flex; in CSS.

**React.js - Nightmare Level**

1. **(b)** 1
   * useEffect(() => { setCount((prev) => prev + 1); }, []); runs once on mount.
2. **(a)** To optimize the diffing algorithm
   * Keys help React identify and efficiently update list items.
3. **(a)** Prevents unnecessary re-renders
   * React.memo() memoizes a component to avoid re-renders.
4. **(a)** Delays state updates for smoother UI
   * useDeferredValue() defers state updates to prioritize responsiveness.
5. **(a)** Allows subscribing to external stores efficiently
   * useSyncExternalStore() efficiently tracks external state changes.

4o