

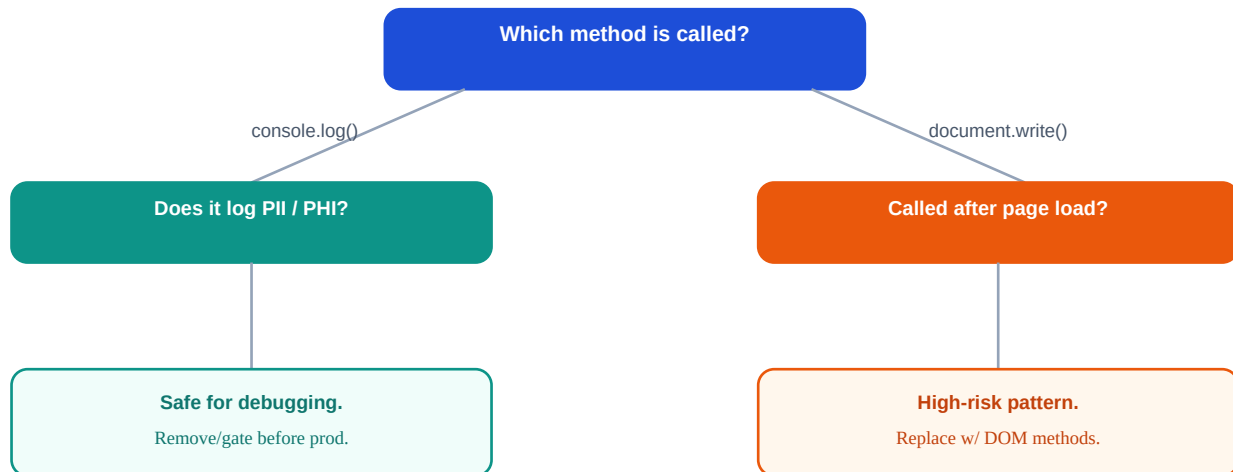
JavaScript Output Methods Audit Checklist

Flag `document.write()` usage and PII-unsafe `console.log()` calls before they reach production.

1. At a Glance

Characteristic	<code>document.write()</code>	<code>console.log()</code>
Output destination	The HTML document	Browser console only
Visible to end users?	Yes	No, unless DevTools open
Safe after page load?	No — overwrites the page	Yes
Production-appropriate?	No	Only if PII/PHI-free and gated

2. Triage Decision Tree



3. Pre-Production Checklist

- Search the codebase for `document.write()` — flag every result for removal or replacement with `createElement/appendChild`.
- Confirm the build pipeline (Webpack/Vite/esbuild) strips `console.*` calls in production builds — don't assume, verify the config.
- Review any `console.log()` touching user objects — log specific safe fields (e.g. `user.id`), never the full object.
- Test all third-party embeds (chat widgets, ads, analytics) under throttled 3G, not just office Wi-Fi — `document.write()` failures are network-dependent.

■ Check DevTools console for '[Violation] Avoid using document.write()' warnings during any blank-page defect triage.

■ Request vendor documentation confirming async DOM-based embed methods before approving any new third-party script.

Source: Chrome's official Lighthouse documentation confirms `document.write()` can delay page content by tens of seconds on slow connections, and that Chrome may block its execution outright.