

# Where to Put Schema Code in Website: Best Practices for Modern SEO and AI Discovery

Metehan Yesilyurt

*AI Search & Information Retrieval Researcher · metehan.ai*

Published: September 06, 2026 ·

Canonical: <https://metehan.ai/articles/where-to-put-schema-code-in-website-best-practices-for-modern-seo-and-ai-discovery/>

I get asked about schema placement more than almost any other technical SEO topic. "Where exactly should I put my schema code?" sounds like a simple question, but the answer has real implications for both traditional search performance and AI discoverability. After implementing schema markup on hundreds of pages across different platforms and CMS systems, I have developed strong opinions on what works best. Let me share them.

## Why Schema Placement Matters More Than You Think

Schema markup (structured data) tells search engines and AI crawlers exactly what your content is about in a machine-readable format. But the placement of that schema on your page affects how reliably it gets parsed, how quickly it loads, and whether it even gets read at all.

Google has stated that they support JSON-LD, Microdata, and RDFa, and that JSON-LD can be placed in the `<head>` or `<body>` of a page. But "supported" and "optimal" are different things. AI crawlers from companies like OpenAI, Anthropic, and Perplexity also consume structured data, and their parsing behavior may differ from Google's.

## The Three Main Placement Options

There are three common approaches to placing schema on a website, each with distinct trade-offs.

### Option 1: In the `<head>` Section

Placing JSON-LD schema in the `<head>` of your HTML document is my recommended approach for most situations. The `<head>` is parsed first by browsers and crawlers, which means your structured data is available immediately. This reduces the risk of schema being missed if a crawler has a rendering budget or timeout.

For static sites and server-side rendered pages, this is straightforward. You add a `<script type="application/ld+json">` block inside your `<head>` tags, and it loads with the initial HTML response.

### Option 2: In the `<body>` Section

Placing JSON-LD in the `<body>` is perfectly valid according to Google's documentation. Many CMS plugins default to this approach because it is easier to implement near the content the schema describes. WordPress

plugins like Yoast and Rank Math typically inject schema at the bottom of the `<body>`.

The main advantage is contextual proximity. When your Product schema sits right next to your product description, it is easier to maintain and audit. The disadvantage is that crawlers with limited rendering capability might not reach it if the page is heavy.

### Option 3: JavaScript Injection (Client-Side Rendering)

Some modern frameworks and single-page applications inject schema dynamically via JavaScript after the page loads. This includes React apps that use libraries like `react-helmet` or Next.js applications using the `Head` component.

However, it introduces risk with AI crawlers that may not execute JavaScript, or that may have shorter rendering timeouts. This approach works for Googlebot because Google renders JavaScript. I have seen cases where dynamically injected schema was completely invisible to certain AI crawlers.

## Placement Comparison

Here is a side-by-side technical comparison of schema placement methods across search engine crawler compatibility, AI bot execution fidelity, and maintenance overhead:

| Placement Method             | Google Support            | AI Crawler Support   | Page Speed Impact         | Implementation Complexity | Maintenance          | My Recommendation       |
|------------------------------|---------------------------|----------------------|---------------------------|---------------------------|----------------------|-------------------------|
| `` JSON-LD                   | Full support              | Best compatibility   | Minimal (loads with HTML) | Low to moderate           | Easy to audit        | Best for most sites     |
| `` JSON-LD                   | Full support              | Good compatibility   | Minimal                   | Low                       | Moderate (scattered) | Good alternative        |
| Footer JSON-LD               | Full support              | Good compatibility   | Minimal                   | Low                       | Easy to audit        | Acceptable              |
| Inline Microdata             | Full support              | Variable             | None (part of HTML)       | High                      | Difficult            | Avoid unless required   |
| JS-injected JSON-LD          | Full support (renders JS) | Risky (many skip JS) | Depends on framework      | Framework-dependent       | Moderate             | Avoid if possible       |
| Server-side rendered JSON-LD | Full support              | Best compatibility   | Minimal                   | Moderate                  | Easy                 | Ideal for JS frameworks |
| Tag manager injection        | Supported but discouraged | Unreliable           | Delayed loading           | Low                       | Easy                 | Last resort only        |

## My Recommended Approach

For most websites, I recommend placing JSON-LD schema in the `<head>` section, generated server-side. Here is why:

**Reliability:** Every crawler, whether Google, Bing, OpenAI, or Perplexity, parses the `<head>` first. By placing your schema there, you ensure maximum compatibility.

**Performance:** JSON-LD in the `<head>` loads with the initial HTML response. There is no waiting for JavaScript to execute or for the DOM to fully render.

**Auditability:** Having all your schema in one predictable location makes it easier to audit and debug. When I run schema validation, I know exactly where to look.

**AI readiness:** This is increasingly important. AI crawlers are newer and less mature than Googlebot. They may not render JavaScript, may have shorter timeouts, and may prioritize content in the `<head>`. Placing schema there gives you the best chance of being properly understood by AI systems.

## Platform-Specific Guidance

---

### WordPress

If you use Yoast or Rank Math, the schema is typically injected in the `<head>` automatically. Verify this by viewing your page source and searching for `application/ld+json`. If you are adding custom schema beyond what your plugin generates, add it to your theme's `header.php` or use a custom plugin that outputs in the `<head>`.

### Shopify

Shopify themes often include product schema in the `<body>` near the product template. For better AI compatibility, I recommend adding critical schema (Organization, WebSite) to your `theme.liquid` file within the `<head>`. Product-specific schema in the body is acceptable since it changes per page.

### Next.js and React

Use server-side rendering (SSR) or static site generation (SSG) to ensure schema is present in the initial HTML. Avoid client-side-only schema injection. Next.js makes this easy with the `Head` component in `getServerSideProps` or `getStaticProps`. For Next.js App Router, use the `metadata` API or a `Script` component with `strategy="beforeInteractive"`.

### Astro

Astro is excellent for schema placement because it is server-rendered by default. Add your JSON-LD directly in the `<head>` of your layout components. Since Astro outputs static HTML, your schema is always present in the initial response with zero JavaScript dependency.

## Common Mistakes I See

---

**Duplicate schema:** Multiple plugins or components injecting the same schema type. This confuses crawlers and can lead to validation errors. Audit your pages regularly to catch duplicates.

**Orphaned schema:** Schema that references entities not present on the page. If your FAQ schema lists questions that do not appear in the visible content, Google may flag this as a mismatch.

**Tag Manager schema:** Injecting schema via Google Tag Manager. While technically functional for Google, the delayed loading and JavaScript dependency make it unreliable for AI crawlers. I strongly advise against this.

approach.

**Missing updates:** Schema that was accurate when created but has not been updated as products, prices, or content changed. Stale schema is worse than no schema because it provides incorrect information to AI models.

## Validating Your Schema Placement

---

After implementing schema, validate it using Google's Rich Results Test and Schema.org's validator. But do not stop there. For AI visibility specifically, it is worth testing your pages with tools like Profound, Peec AI, AirOps, or AEO Vision to verify that AI crawlers are actually reading and using your structured data. The gap between "valid schema" and "schema that AI models actually use" is real, and these tools help you identify it.

## Frequently Asked Questions

---

Here are direct, authoritative answers to frequently asked questions about JSON-LD placement, multi-schema scripts, and structured data validation for search engines and AI models:

### Can I have multiple JSON-LD blocks on a single page?

Yes, absolutely. Google explicitly supports multiple JSON-LD script blocks on a single page. I commonly use separate blocks for Organization, BreadcrumbList, Product, and FAQ schema on the same page. Keeping them separate actually makes maintenance easier than combining everything into a single block. Just make sure there are no conflicts or duplicates between blocks.

### Does schema placement affect page speed?

The impact is negligible for JSON-LD because it is just a small script tag that does not render anything visible. A typical schema block adds 1 to 3 KB to your page, which is trivial. The exception is if you are dynamically generating schema with heavy JavaScript, in which case the script execution could add to your page's processing time. Server-side rendered JSON-LD has virtually zero performance impact.

### Should I use JSON-LD or Microdata for AI visibility?

JSON-LD, without question. Google recommends it, it is easier to implement and maintain, and it does not require modifying your HTML structure. Most AI crawlers are optimized for JSON-LD parsing. Microdata requires embedding attributes directly in your HTML elements, which makes it harder to maintain and more error-prone. The only scenario where Microdata might be preferable is if you are working with a legacy system that cannot easily add script tags.

### How do I know if AI crawlers are actually reading my schema?

There is no direct equivalent of Google Search Console for AI crawlers yet. The best approach is to monitor your AI visibility using tracking platforms and correlate changes with schema updates. If you add comprehensive Product schema and your product starts appearing in AI recommendations within a few weeks, that is a strong signal. Tools like Profound and Peec AI can help you track these correlations over time.

Schema placement is a quiet AEO multiplier. The pages that earn AI citations in ChatGPT, Perplexity and Gemini disproportionately use clean structured data. Where you put schema in your website decides whether answer

engine optimization work has anything to ride on.