

Schema.org Cheatsheet (Builders Edition)

How your Jekyll metadata becomes machine-readable meaning (JSON-LD)

For blog + code folks

The 10-second mental model

You write **YAML front matter** → the site emits **JSON-LD** in the page <head> → search engines understand your content.

YAML → Schema (what actually gets generated)

| YAML | JSON-LD |
|---|-------------------------------------|
| title: "Getting Started with GitHub Pages" | { |
| subtitle: "Launch a free website in minutes" | "@type": "HowTo", |
| excerpt: "A beginner-friendly guide..." | "headline": "Getting Started...", |
| tags: [github-pages, tutorial, web-development] | "alternativeHeadline": "Launch...", |
| date: 2024-07-28 | "description": "A beginner...", |
| | "datePublished": "2024-07-28...", |
| | "keywords": "github-pages..." |

Where this lives in your repo

- `_includes/schema.html` generates JSON-LD
- `_includes/head.html` includes schema on every page
- `assets/visualizations/schema-examples.html` interactive demo

Three schema types (auto-selected)

Article: Default for posts/pages

HowTo: If tags/categories include **tutorial**

SoftwareSourceCode: If content is in `_projects` collection

High-impact field mapping

- **title** → headline
- **subtitle** → alternativeHeadline
- **excerpt** → description
- **date** → datePublished (YYYY-MM-DD)
- **tags** → keywords (comma-separated)
- **header image** → image / social previews
- **collection** → drives @type selection

Tip: keep tags as concepts (not tool soup) so "aboutness" stays clean.

Verify it in under 2 minutes

- **View source:** search for "@context": "`https://schema.org`"
- **Rich Results Test:** search.google.com/test/rich-results
- **Schema Validator:** validator.schema.org
- **Interactive demo:** </assets/visualizations/schema-examples.html>