

Schema Markup für Produkte: Schritt-für-Schritt-Anleitung

Strukturierte Daten richtig implementieren und Rich Snippets in Google freischalten

Tobias Wenzel · 25.08.2026

Es ist Freitag, 14 Uhr, und ein Kunde sucht bei Google nach "wasserdichte Smartwatch unter 200 Euro". Die Suchergebnisse laden. Bei einem Shop sieht er sofort den Preis, die Bewertung mit Sternen und ob das Produkt verfügbar ist - direkt in der Vorschau. Bei den anderen Shops muss er erst klicken, um diese Infos zu sehen. Welchen wird er wählen? Genau: den ersten. Das ist die Kraft von Schema Markup.

Strukturierte Daten sind nicht optional für E-Commerce-Seiten - sie sind der Unterschied zwischen einer langweiligen Textzeile und einer clickfähigen Rich Snippet, die Conversions anzieht. Google bevorzugt Shops, die ihre Produktinformationen maschinenlesbar auszeichnen. Der Aufwand ist überschaubar, der Return On Investment aber erheblich.

Was ist Product Schema Markup und warum braucht es jeder Online-Shop?

Strukturierte Daten sind Informationen, die in einem standardisierten Format codiert sind - so, dass nicht nur Menschen, sondern auch Suchmaschinen und andere Systeme sie verstehen. Product Schema ist ein Vokabular von schema.org, das speziell für Produkte gedacht ist.

Wenn Sie kein Schema Markup nutzen, sieht Google nur den Rohtext Ihrer Seite. Es muss raten: Ist das eine Produktseite? Welcher Preis ist der aktuelle? Wie viele Bewertungen gibt es? Mit Schema Markup sagen Sie es Google direkt und unmissverständlich.

Die Vorteile sind messbar:

- Höhere Klickrate (CTR): Rich Snippets mit Sternen, Preis und Verfügbarkeit erhalten bis zu 30 % mehr Klicks als einfache Text-Links.
- Bessere Crawl-Effizienz: Google indexiert Ihre Produktseiten schneller und genauer.
- Voice Search Readiness: Sprachassistenten wie Google Assistant greifen auf strukturierte Daten zu.
- E-Commerce-spezifische Features: Google Shopping, Google Lens und andere Dienste nutzen Schema Markup.
- Wettbewerbsvorteil: Nicht alle Shops implementieren es - wer es tut, fällt auf.

Laut Google sind Shops, die Product Schema nutzen, bei der Rich Result Eligibility vorne dabei. Das bedeutet konkret: mehr Sichtbarkeit, mehr Traffic, mehr Umsatz.

Die Anatomie des Product Schema: Pflichtfelder und optionale Attribute

Product Schema hat eine klare Hierarchie. Es gibt Pflichtfelder, die Sie setzen müssen, und optionale, die den Unterschied zwischen einem OK-Schema und einem

großartigen machen.

Die absoluten Pflichtfelder

Ohne diese Felder wird Ihr Schema nicht vollständig erkannt:

- **name:** Der Produktname. Beispiel: "Doppelwandige Thermoflasche 500ml, Schwarz". Sei präzise, nicht "Produkt" oder "Artikel".
- **image:** Die URL eines Produktbildes. Mindestens 100x100 Pixel, idealerweise 1200x1200 oder höher. Google bevorzugt quadratische oder annähernd quadratische Bilder.
- **description:** Eine kurze Produktbeschreibung. 50-160 Zeichen sind optimal.

Hochrelevante optionale Felder (die du setzen solltest)

- **offers:** Der Preis, die Währung und der Verfügbarkeitsstatus. Das ist das Feld, das Rich Snippets in Google auslöst. Ohne offers kein Preisanzeige in der Suche.
- **aggregateRating:** Bewertungen und Sternchen. Dieses Feld ist Gold wert - es erhöht die CTR massiv.
- **brand:** Die Marke des Produkts.
- **sku:** Die interne Artikelnummer.
- **productId:** Die globale Produktkennung (UPC, EAN, ISBN).

Weitere nützliche Felder

Je nach Produkttyp:

- **color:** Farbe
- **size:** Größe
- **material:** Material
- **weight:** Gewicht
- **dimensions:** Abmessungen
- **review:** Einzelne Bewertungen mit Autor, Datum und Inhalt
- **isPartOf:** Gehört das Produkt zu einer Kollektion oder Serie?

Ein häufiger Fehler: Shops implementieren nur name und image und wundern sich, warum keine Rich Snippets erscheinen. Das liegt fast immer daran, dass das offers-Feld fehlt oder falsch formatiert ist.

Schritt 1: Wähle dein Markup-Format - JSON-LD vs. Microdata vs. RDFa

Es gibt drei Wege, strukturierte Daten einzubauen. Google bevorzugt klar eine Methode.

JSON-LD (JSON for Linking Data)

Das ist der Standard, den Google empfiehlt. Es wird als `<script>`-Block im `<head>` oder `<body>` eingefügt und beeinflusst nicht das HTML-Rendering.

Vorteile: Sauber, wartbar, kein Durcheinander mit HTML-Attributen, relativ einfach zu debuggen. Nachteil: Es ist ein Extra-Block im Code.

Microdata

Microdata nutzt HTML-Attribute wie `itemscope`, `itemtype` und `itemprop`. Es wird direkt in

die HTML-Tags eingebaut.

Beispiel: `<div itemscope itemtype="https://schema.org/Product">`. Vorteil: Alles ist inline, keine zusätzlichen Script-Tags. Nachteil: Unübersichtlich, schwerer zu warten, leicht zu durcheinander zu bringen.

RDFa

RDFa ist älter, weniger verbreitet. Google unterstützt es, aber viele Shops nutzen es nicht mehr. Wir ignorieren es für diese Anleitung.

Empfehlung: Nutze JSON-LD. Es ist wartbar, Google-konform und wird von den meisten E-Commerce-Plattformen (Shopify, WooCommerce, Magento) nativ unterstützt.

Schritt 2: Implementiere das Product Schema - Praktisches Code-Beispiel

Hier ist ein realistisches Beispiel für eine Produktseite - ein kabelloser Kopfhörer für 129,99 Euro:

```
<script type="application/ld+json"> { "@context": "https://schema.org/", "@type":  
"Product", "name": "SoundMax Pro Wireless Kopfhörer, Bluetooth 5.3, 30h Akku",  
"image": [r> "https://example-shop.de/images/soundmax-pro-1.jpg",  
"https://example-shop.de/images/soundmax-pro-2.jpg",  
"https://example-shop.de/images/soundmax-pro-3.jpg"r> ], "description":  
"Premium-Kopfhörer mit aktiver Geräuschunterdrückung, 30 Stunden Akkulaufzeit und  
schneller Kabellosverbindung.", "brand": {r> "@type": "Brand", "name": "SoundMax"r> },  
"offers": {r> "@type": "Offer", "url": "https://example-shop.de/produkte/soundmax-pro",  
"priceCurrency": "EUR", "price": "129.99", "priceValidUntil": "2026-12-31",  
"availability": "https://schema.org/InStock", "seller": {r> "@type": "Organization",  
"name": "ElectroShop GmbH"r> }r> }, "aggregateRating": {r> "@type": "AggregateRating",  
"ratingValue": "4.7", "ratingCount": "342", "bestRating": "5", "worstRating": "1"r> },  
"sku": "SM-PRO-BLK-2026", "gtin": "4260123456789", "color": "Schwarz", "weight": {r>  
"@type": "QuantitativeValue", "unitCode": "GRM", "value": "250"r> }r> } </script>
```

Lass mich das aufschlüsseln:

- @context und @type: Das sagt Google: "Das ist ein Produkt nach schema.org-Standard."
- name: Der Produktname. Sprechend, mit Varianten wenn relevant.
- image: Array von Bild-URLs. Google nimmt das erste für Rich Snippets, aber mehrere sind besser für Bildsuche und Social Media.
- description: Kurz und aussagekräftig. Nicht die 5000-Wörter-Kategorie-Beschreibung, sondern eine prägnante Zusammenfassung.
- brand: Nested Object mit Name. Wichtig für Markensuchen.
- offers: Das Herzstück. Preis, Währung, Verfügbarkeit. Das availability-Feld sollte "InStock", "OutOfStock" oder "PreOrder" sein - nicht "Verfügbar" oder "Lagernd".
- aggregateRating: Die durchschnittliche Bewertung mit Anzahl der Bewertungen. Das ist der Hingucker in Google-Ergebnissen.
- sku und gtin: Interne und globale Nummern für Eindeutigkeit.

- color und weight: Zusätzliche Attribute, die für bestimmte Produkttypen hilfreich sind.

Ein wichtiger Punkt: Das Markup muss mit den tatsächlichen Daten auf der Seite übereinstimmen. Wenn auf der Seite 119,99 Euro angezeigt wird, aber das Schema 129,99 Euro sagt, wird Google misstrauisch und kann das Markup ignorieren.

Schritt 3: Mehrere Varianten mit ProductCollection oder VariationProduct

Viele Produkte gibt es in verschiedenen Farben, Größen oder Konfigurationen. Wie markierst du das?

Variante 1: Separate Product-Objekte für jede Variante

Wenn dein Shop jede Variante eine eigene URL hat (z.B. /soundmax-pro-schwarz und /soundmax-pro-silber), dann erstelle für jede ein separates Product-Schema mit ihren eigenen Daten.

Variante 2: Ein Product-Objekt mit mehreren offers

Wenn alle Varianten auf einer Seite sind, nutze mehrere offers-Objekte:

```
"offers": [ {r> "@type": "Offer", "url":  
"https://example-shop.de/produkte/soundmax-pro?color=schwarz", "priceCurrency": "EUR",  
"price": "129.99", "availability": "https://schema.org/InStock"r> }, {r> "@type":  
"Offer", "url": "https://example-shop.de/produkte/soundmax-pro?color=silber",  
"priceCurrency": "EUR", "price": "139.99", "availability":  
"https://schema.org/OutOfStock"r> }r>]
```

Google bevorzugt hier Clartät: Wenn möglich, separate URLs und separate Schemas. Das macht es einfacher zu verstehen, welche Variante verfügbar ist.

Schritt 4: Validierung und Fehlersuche mit Google Tools

Dein Schema ist implementiert - aber funktioniert es wirklich? Google bietet zwei Hauptwerkzeuge.

Rich Results Test

Gehe zu <https://search.google.com/test/rich-results> (oder suche nach "Google Rich Results Test"). Gib die URL deiner Produktseite ein und klick auf "Test URL". Google crawlt die Seite und teilt dir mit:

- Welche strukturierten Daten erkannt wurden
- Welche Rich Results eligible sind
- Welche Fehler oder Warnungen es gibt

Typische Fehler, die hier auftauchen:

- Missing field "offers": Das ist der häufigste. Ohne offers keine Rich Snippets.
- Invalid price format: Der Preis muss eine Zahl sein (129.99), nicht "129,99 Euro".
- Invalid currency: EUR, USD, GBP - nicht "Euro" oder "€".

- Missing image: Jedes Product braucht mindestens ein Bild.
- Invalid rating: ratingValue muss zwischen 1 und 5 liegen (oder zwischen bestRating und worstRating).

Schema.org Validator

Alternativ nutze den Schema.org-Validator (<https://validator.schema.org>). Dieser ist etwas technischer, zeigt aber exakt, wo in deinem JSON-LD die Fehler sind.

Mein Tipp: Validiere nicht nur eine Seite, sondern ein paar verschiedene Produkte. Unterschiedliche Produkttypen können unterschiedliche Felder brauchen.

Schritt 5: Häufige Fehler und wie du sie vermeidest

Basierend auf Audits von hunderten von Shops - hier sind die Fehler, die immer wieder auftauchen:

Fehler 1: Veraltete Preise im Markup

Der Preis im Schema ist 99,99 Euro, auf der Seite aber 79,99. Das ist nicht nur irreführend, Google kann das auch als Spam interpretieren. Nutze dynamische Schemas, die sich automatisch mit dem aktuellen Preis aktualisieren - dein CMS oder dein Shop-System sollte das können.

Fehler 2: Falsche availability-Werte

"Verfügbar", "In Lager", "Lagernd" - das sind keine gültigen Werte. Google versteht nur: InStock, OutOfStock, PreOrder, Discontinued. Wenn du einen ungültigen Wert setzt, ignoriert Google das ganze availability-Feld.

Fehler 3: Keine Seller-Information bei Marketplace-Produkten

Wenn dein Shop auf Amazon, eBay oder einem Marktplatz verkauft: Nutze das seller-Feld im offers-Objekt, um klar zu machen, wer das Produkt anbietet. Das hilft Google, dich in lokalen oder Marketplace-Suchen richtig zu ranken.

Fehler 4: Übertriebene oder gefälschte Bewertungen

Google nutzt Machine Learning, um Spam-Bewertungen zu erkennen. Wenn dein neues Produkt plötzlich 4,8 Sterne mit 500 Bewertungen nach einer Woche hat, wird Google skeptisch. Nutze nur echte, verifizierte Bewertungen im Schema.

Fehler 5: Fehlende oder minderwertige Bilder

Bilder unter 100x100 Pixel, verpixelt oder nicht repräsentativ - das schadet deinem Rich Snippet. Google wird das Bild möglicherweise nicht anzeigen, und der Snippet sieht schwach aus.

Schritt 6: Best Practices für maximale Sichtbarkeit

Okay, dein Schema ist valide. Wie machst du es noch besser?

Nutze aggregateRating konsequent

Bewertungen sind die CTR-Multiplikatoren. Ein Produkt mit 4,7 Sternen erhält bis zu 35 % mehr Klicks als eins ohne Bewertung. Wenn dein Shop ein Bewertungssystem hat, implementiere aggregateRating auf jeder Produktseite. Und sei ehrlich - nur echte Bewertungen.

Setze priceValidUntil

Das Feld priceValidUntil teilt Google mit, bis wann der Preis gültig ist. Das ist wichtig für Angebote und Aktionen. Beispiel: Ein Black-Friday-Angebot hat einen priceValidUntil von "2026-11-30". Nach diesem Datum kann Google das Angebot ignorieren oder als Fehler flaggen.

Nutze mehrere Bilder

Das image-Feld kann ein Array sein. Setze 3-5 hochwertige Bilder: Frontalansicht, Seitenansicht, Detail, Einsatzbild. Google nutzt diese für Google Images und für die Bildsuche.

Implementiere Review-Schema für Einzelbewertungen

Zusätzlich zum aggregateRating kannst du einzelne Review-Objekte hinzufügen:

```
"review": [ {r> "@type": "Review", "author": {"@type": "Person", "name": "Max M."},  
"datePublished": "2026-03-15", "reviewRating": {"@type": "Rating", "ratingValue": "5"},  
"reviewBody": "Exzellenter Klang, sehr bequem, lange Akkulaufzeit."r> }r>]
```

Das erhöht die Glaubwürdigkeit und kann in erweiterten Rich Snippets angezeigt werden.

Teste regelmäßig und monitore die Implementierung

Implementierung ist nicht "einmal machen und vergessen". Wenn dein Shop sich ändert (neuer Shop-Plattform, neues Theme, Preis-Updates), können Fehler entstehen. Nutze Google Search Console, um zu sehen, ob Rich Results noch angezeigt werden. Wenn die Anzahl plötzlich sinkt, stimmt etwas nicht mit deinem Markup.

Schritt 7: Integration mit beliebten Shop-Plattformen

Je nach deiner Plattform ist die Implementierung unterschiedlich:

Shopify

Shopify generiert automatisch Product Schema für deine Produkte. Du brauchst normalerweise nichts manuell zu tun. Wenn du aber angepasste Felder brauchst, nutze Liquid-Templates oder eine App wie "JSON-LD Schema" oder "Schema App". Schau in dein Theme in den Produktseiten-Code - da sollte das Schema schon vorhanden sein.

WooCommerce

WooCommerce mit dem Yoast SEO Plugin generiert Product Schema automatisch. Aktiviere es unter SEO ? Search Appearance ? Product. Wenn du tiefere Kontrolle brauchst, nutze Code-Snippets in functions.php oder ein Plugin wie "Schema Pro".

Magento

Magento hat eingebaute Strukturierte-Daten-Features, die teilweise standardmäßig aktiviert sind. Überprüfe unter Stores ? Configuration ? Catalog ? Structured Data. Für erweiterte Anpassungen brauchst du einen Entwickler.

Custom-Shops

Wenn du einen eigenen Shop hast, musst du das Schema manuell implementieren. Das ist eine Aufgabe für einen Frontend-Entwickler. Die obigen Code-Beispiele sind ein guter Startpunkt.

Schritt 8: Monitoring und Optimierung nach Launch

Dein Schema läuft. Wie misst du den Erfolg?

Google Search Console

Gehe zu Search Console ? Performance und filtere nach "Rich Results". Du siehst, wie viele Impressionen deine Produkte mit Rich Snippets bekommen. Vergleiche das mit der Zeit vor dem Markup - die Zahlen sollten steigen.

Außerdem: Gehe zu Enhancements ? Rich Results. Google zeigt dir, wenn es Fehler in deinem Markup gibt. Diese Liste sollte klein sein oder leer.

Google Analytics und Konversionsmetriken

Rich Snippets erhöhen die CTR, aber letztendlich geht es um Verkäufe. Überwache in Google Analytics, wie sich der Traffic und die Konversionsrate nach der Implementierung entwickeln. Manche Shops sehen 15-25 % CTR-Verbesserung, andere weniger - es hängt vom Wettbewerb und der Produktkategorie ab.

A/B-Testing

Wenn du mehrere Shop-Instanzen hast, implementiere das Schema nur auf einer und beobachte den Unterschied. Das gibt dir valide Daten.

Abschließende Gedanken

Product Schema Markup ist kein Hexenwerk. Es ist strukturierte Arbeit: Daten sammeln, richtig formatieren, validieren, monitoren. Shops, die das ernst nehmen, haben einen echten Vorteil in der Google-Suche.

Der häufigste Fehler, den ich sehe, ist Halbherzigkeit. Ein Shop implementiert das Schema zu 80 %, vergisst aber die aggregateRating oder nutzt falsche Verfügbarkeitswerte. Das reicht nicht für Top-Rankings und Rich Snippets. Wenn du es machst, dann richtig.

Starte mit den Pflichtfeldern (name, image, offers, description) und baue von dort auf. Validiere regelmäßig, aktualisiere die Daten automatisch und monitore die Resultate in der Search Console. In 2-4 Wochen solltest du erste Verbesserungen in der Klickrate sehen.

Die Investition ist klein, der Return kann erheblich sein. Viel Erfolg beim Implementieren.