

# GUTER TECH SEO WERDEN, OHNE PROGRAMMIEREN ZU KÖNNEN? EIN SELBSTVERSUCH



Seit 10 Jahren habe ich „irgendwie was mit SEO zu tun“, seit etwa fünf Jahren verdiene ich damit mein Geld. Die große Begeisterung für technisches SEO teile ich sicher mit vielen aus der Branche seit Tag eins. Aber wie bekommt man jetzt das reine Interesse so richtig auf die Straße? Und kommt man dabei wirklich ohne Programmier-Skills ans Ziel?

In meiner ersten Woche als SEO-Trainee und damit meiner ersten „offenen Fragerunde“, an der ich schon mal als ZuhörerIn teilnehmen durfte, diskutierte meine Kollegin das Thema „Paginierung mit Ghostblock“ aus dem Guide von Tobias Schwarz. Steiler Einstieg – ich war mir weder sicher, ob sie Deutsch spricht, noch ob ich „hier richtig“ bin. Aber angefixt war ich irgendwie trotzdem, sodass ich mir damals vorgenommen habe: „Eines Tages will ich verstehen, wovon die hier geredet haben.“

## Der Unterschied zwischen Theorie und Praxis ist in der Theorie kleiner

Um sich erst einmal die technischen SEO-Basics draufzuschaffen, gibt es unzählige wirklich gute Blogs, Bücher, Schulungen, die hilfreichen Tipps der Kollegen u. v. m. – ein großer Pool an Möglichkeiten, sich die grundsätzlichen Dinge beizubringen. Mit der praktischen Erfahrung durch den Job kommt mit der Zeit dazu, dass man die Dinge

nicht nur isoliert versteht, sondern besser in einen Gesamtkontext einordnen kann. Je mehr verschiedene Websites man betreut, desto mehr technische Problemstellungen und Set-ups sieht man und lernt kontinuierlich dazu. Man kommt langsam in eine Phase von „die meisten Sachen irgendwie irgendwo schon mal gehört“, und damit kann man sich ganz gut über Wasser halten.

Bis du das erste Mal einschätzen sollst, welche technische Umsetzung einfacher und schneller geht, oder deine mühsam analysierten Page Speed Findings in eine Priorisierungsreihenfolge nach Impact absteigend sortieren sollst – plötzlich merkst du, dass du irgendwie in der Praxis doch nicht so viel weißt, wie du in der Theorie dachtest.

Technische SEO-Issues zu identifizieren und Empfehlungen auszusprechen, welche den meisten Impact auf die SEO-Performance haben, lernt man an sich verhältnismäßig schnell. Ich hatte das Gefühl, mich auszukennen, und trotzdem habe ich mich oft unsicher gefühlt,

wenn „handfeste“ pragmatische Tipps für Entwickler oder realistisches Abwägen von „so wäre es in einer perfekten SEO-Welt“ vs. „knappe DEV-Ressourcen“ für die Umsetzung meiner Audit-Ergebnisse gefragt waren.

## Kickstart: Tech SEO Mentoring

Ich war die Erste, die „hier“ und „Tech SEO“ geschrien hat, als mir die „offizielle“ Möglichkeit gegeben wurde, mich auf ein Themengebiet im SEO zu spezialisieren – on the job. Eine Stunde die Woche darf ich alle technischen (SEO-)Fragen dieser Welt stellen und das an „The One and Only“ Matthias Hotz (unseren Head of Technical SEO).

Die erste Session haben wir quasi bei null begonnen: Wie setze ich einen Server auf? Indem wir es live gemacht haben! Seitdem haben wir sämtliche Grundlagen der Webentwicklung und wie, wo, wann sie einen Einfluss auf SEO haben, ausinandergenommen, wieder zusammengesetzt und mit Absicht kaputt

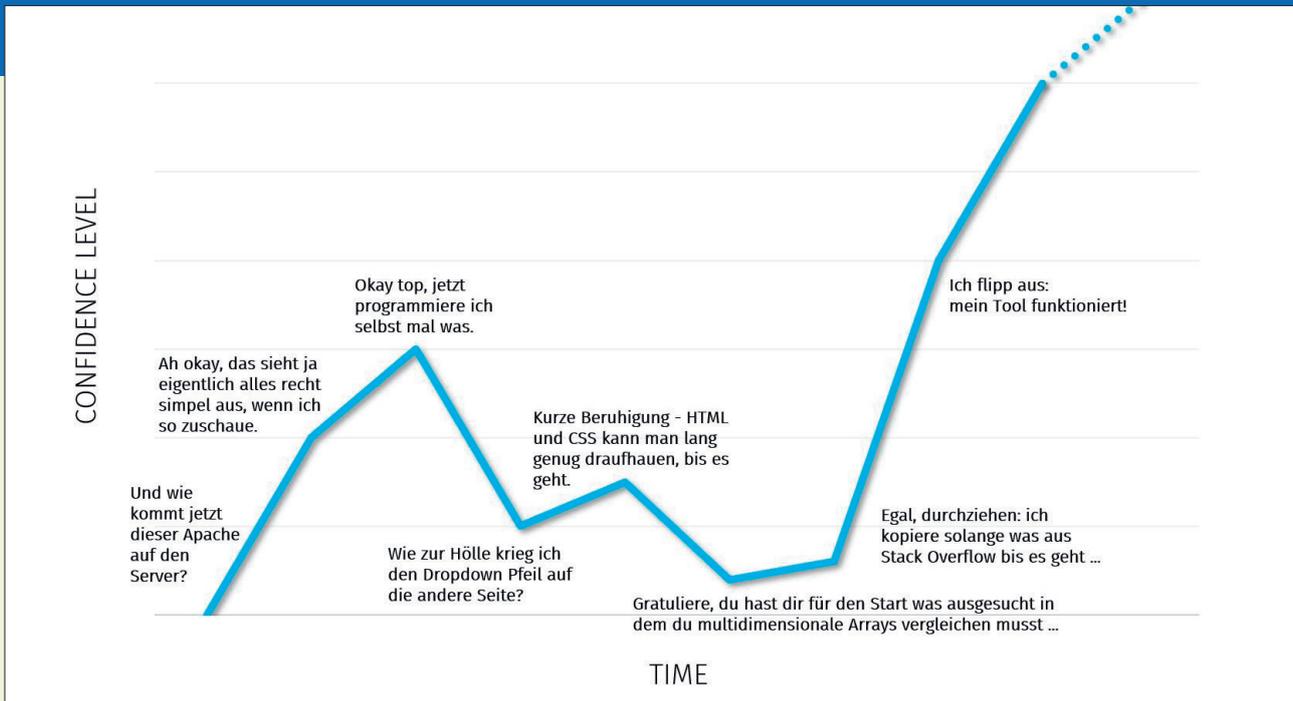


Abb.1: Meine persönliche Lernkurve

gemacht – bis ich nach ein paar Monaten dachte, ich lerne viel dazu, ich verstehe, was wir hier besprechen, aber mir dämmert langsam, dass es ohne „es richtig selbst zu probieren“ vielleicht nicht wirklich haften bleibt.

Also habe ich spontan beschlossen, mein erstes SEO-Tool zu programmieren. Matthias hatte mir bereits gezeigt, wie man Formulare baut, Google-Ergebnisse scraped und mit AJAX seine Ergebnisse wieder ins HTML zurückbekommt. ‚Damit müsste man ja grundsätzlich das SERP Overlap Tool von SEO Revolution zusammenkriegen‘, dachte ich mir. Mir war klar, ich muss an etwas arbeiten, was ich danach auch wirklich verwenden kann. Eine Übungsaufgabe zu machen, um dann eine Übungsaufgabe gemacht zu haben, weckt bei mir überschaubaren Ehrgeiz.

**I am God vs. I have no idea what I'm doing**

Die Lernkurve beim Programmieren hat ja schon eine gewisse

Komik. Man fängt an, das Frontend zu bauen, und freut sich wie ein Schnitzel, wenn das mit ein bisschen HTML und CSS sehr schnell klappt und nach was aussieht. Da es mir als kleiner Monk schwerfällt, Matthias' Rat, „schau erst mal, dass es funktioniert, dann kannst du's immer noch schön machen“, zu folgen (so sinnvoll er ist), musste ich mich durch ein paar kleinere CSS-Krisen à la „ich flippe aus mit dieser Positionierung“ und Schönheitsfehlern, wie auf welcher Seite des Dropdowns der Pfeil angezeigt wird, kämpfen. Aber wenn man das dann hinbekommt, hat man kurz schon das Gefühl, man ist ein krasser Programmierer. Bis du dich an die wirkliche Funktionalität deines Tools machst und merkst, wie viele Fehler beim Basteln eines PHP-Skripts fabriziert werden können.

Ich würde sagen, ab diesem Zeitpunkt war es ein Wechselbad der Gefühle und ein Auf und Ab der Motivation. Folgende Zustände in beliebiger Wiederholung und Reihenfolge:

- » Wieso geht das nicht, ich habe hier keine Fehler?!
- » Na gut, ich sehe es ein, Code ist 100 % logisch, natürlich war da ein Fehler.
- » Wie beschreiben andere Leute mein Problem wohl auf Stack Overflow?
- » Nice! Ich habe Code aus Stack Overflow kopiert und es funktioniert, ich weiß nur nicht warum.
- » Okay, vielleicht muss ich einfach einsehen, dass ich zu doof fürs Programmieren bin.
- » Übertrendend, es tut genau, was ich will!

Insgesamt muss man sagen, dass man schon eine gewisse Frustrationstoleranz braucht. Gleichzeitig ist es natürlich ein großer Segen, wenn man an Punkten, wo man völlig festhängt und irgendwann den Wald vor lauter Bäumen nicht mehr sieht, einen erfahrenen Entwickler fragen kann. Jemand, der Tipps gibt, warnt, wenn man in die völlig falsche Richtung rennt, und einem als erstes Debuggen beibringt, um besser selbst Fehler zu finden.

Und so habe ich es nach ein paar Wochen tatsächlich geschafft: eine funktionierende Rekonstruktion des SERP Overlap Tools. 😊

### Lohnt sich der Aufwand wirklich?

Man kann viel über Tech SEO wissen und konstruktiv einbringen, so habe ich es lange gemacht. Es eröffnet sich aber tatsächlich eine neue Perspektive und eine andere Ebene des Verstehens, wenn man den Versuch wagt, selbst programmieren zu lernen. Der Bezug zu „Wo passiert das im Code?“, „Wo muss man das auf dem Server festlegen?“, „Welche Elemente kann ich per ID ansprechen für eine Automatisierung?“, „Wie müssen alle SEO-Tools Google-Daten verarbeiten?“ etc. fällt deutlich leichter, weil man gesehen oder sogar schon ausprobiert hat, was im Hintergrund passiert. Als Nächstes steht deshalb für mich an, eine Seite in zahlreichen detaillierten Einzelschritten zu verändern, um zu verstehen, welche Dinge sich maßgeblich auf welche Page-Speed-Metriken auswirken



Abb. 2: Selbst gebautes SERP Overlap Tool

und was einfach nur aufwendig ist, aber herzlich wenig bringt.

Ich kann also nur bestätigen, dass es sich lohnt. Man bekommt einfach noch einmal einen deutlichen Schub für das Thema Technical SEO! Definitiv fünf von fünf



Abb. 3: Selbst gebautes SERP Overlap Tool – Ergebnisseite

Sternen – klare Empfehlung!

PS: Auf den Tag, an dem ich die Nummer mit dem Ghostblock wirklich verstehe, arbeite ich noch hin 😊

