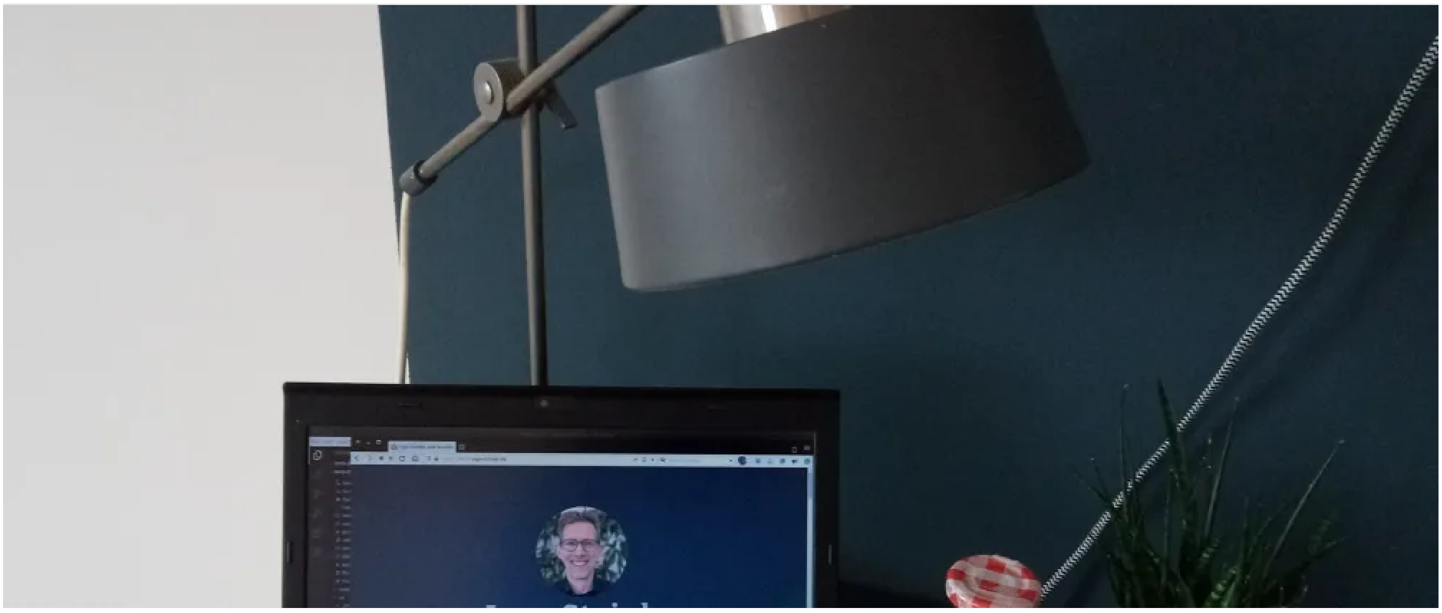


Ingo Steinke
Posted on 23 Feb 2021 • Updated on 18 Mar 2021

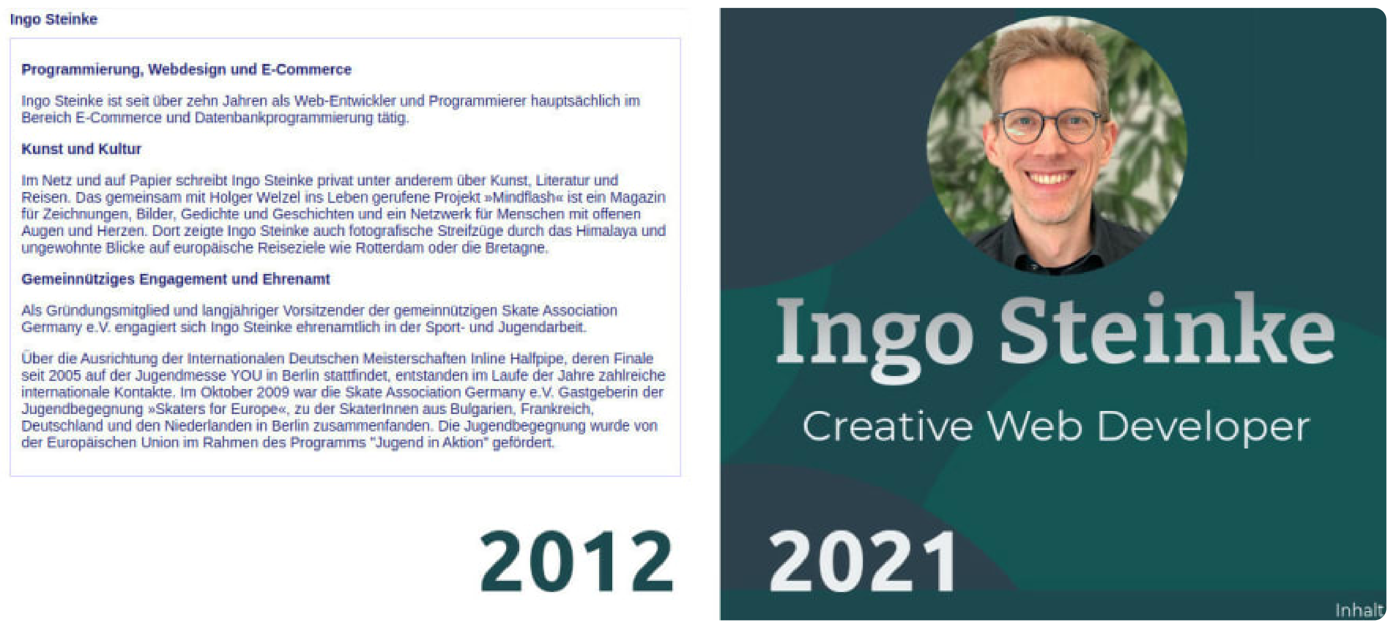
Creating a Fast ⚡ and Beautiful 🌼 Portfolio Website using HTML, CSS 🎨, Eleventy ⑪ and Netlify

#webdev #webperf #eleventy #netlify

Eleven years ago, in 2010, my personal website was minimal by every means. Relaunching in 2021 using Eleventy and modern CSS was fun and brought with it some unexpected insights.

How It Started, How It's Going ...

In 2010, I registered my personal domain and put up a minimal HTML file writing some text about me doing web development, web design and charitable work. I already forgot, but web archive remembers.



Ingo Steinke

Programmierung, Webdesign und E-Commerce

Ingo Steinke ist seit über zehn Jahren als Web-Entwickler und Programmierer hauptsächlich im Bereich E-Commerce und Datenbankprogrammierung tätig.

Kunst und Kultur

Im Netz und auf Papier schreibt Ingo Steinke privat unter anderem über Kunst, Literatur und Reisen. Das gemeinsam mit Holger Welzel ins Leben gerufene Projekt »Mindflash« ist ein Magazin für Zeichnungen, Bilder, Gedichte und Geschichten und ein Netzwerk für Menschen mit offenen Augen und Herzen. Dort zeigte Ingo Steinke auch fotografische Streifzüge durch das Himalaya und ungewohnte Blicke auf europäische Reiseziele wie Rotterdam oder die Bretagne.

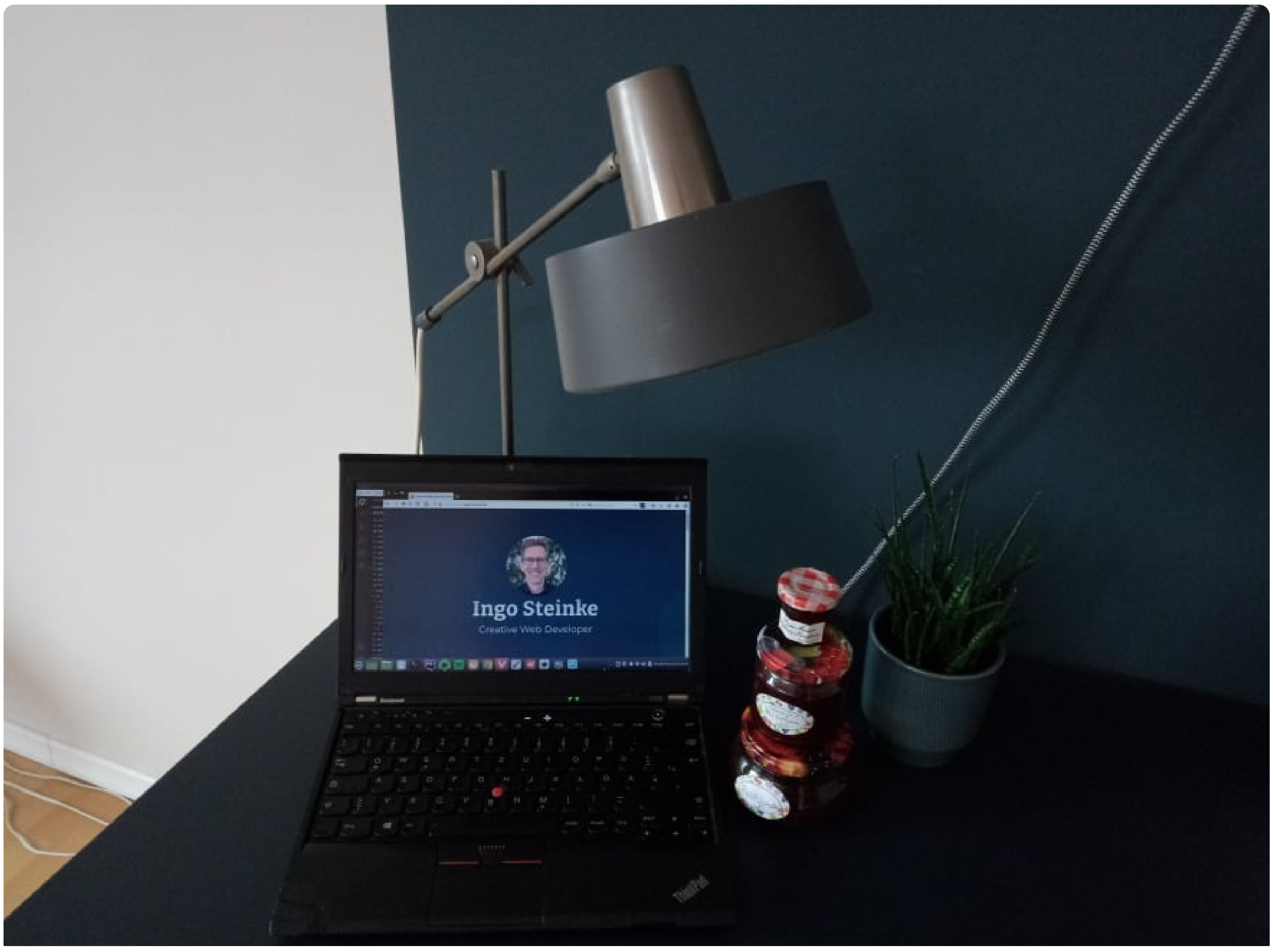
Gemeinnütziges Engagement und Ehrenamt

Als Gründungsmitglied und langjähriger Vorsitzender der gemeinnützigen Skate Association Germany e.V. engagiert sich Ingo Steinke ehrenamtlich in der Sport- und Jugendarbeit.

Über die Ausrichtung der Internationalen Deutschen Meisterschaften Inline Halfpipe, deren Finale seit 2005 auf der Jugendmesse YOU in Berlin stattfindet, entstanden im Laufe der Jahre zahlreiche internationale Kontakte. Im Oktober 2009 war die Skate Association Germany e.V. Gastgeberin der Jugendbegegnung »Skaters for Europe«, zu der SkaterInnen aus Bulgarien, Frankreich, Deutschland und den Niederlanden in Berlin zusammenfanden. Die Jugendbegegnung wurde von der Europäischen Union im Rahmen des Programms "Jugend in Aktion" gefördert.

In 2021, I wanted to keep things light-weight with a focus on page speed, usability and accessibility. All information should fit on one page, so it can be printed out as a resume with a CV, project list, and skill matrix.

Choosing a "Stack"



Lamp stack or jam stack? 💡 🍯

A static site generator fitted my needs, and I liked the effortless deployment on Netlify, once I pushed code to GitHub.

Eleventy + Netlify === JAMStack ?

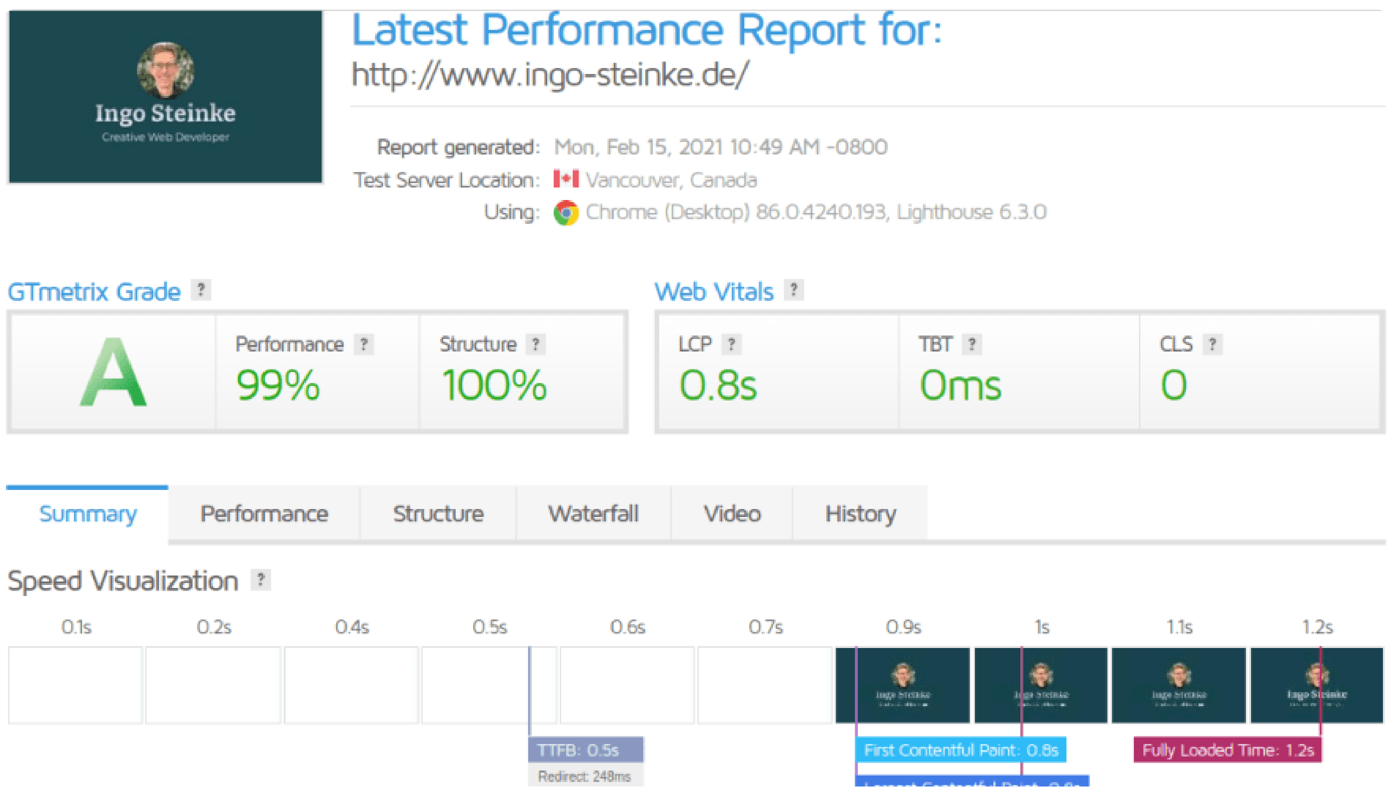
In my opinion, my setup is not necessarily a JAMStack just because I used Eleventy and Netlify. My portfolio does not use any API. Front-end JavaScript is optional, only there to improve the mobile menu experience. My development tools, like StyleLint, PostCSS, Babel, and Cypress depend on node.js, so there is more JavaScript involved at least under the hood.

Despite its complexity at build time, the result is a static website based on a single HTML page, enhanced with some CSS and very little JavaScript.

Core Web Vitals and Progressive Enhancement

Google's Core Web Vitals emphasize usability and web performance for search engine optimization. I knew that progressive enhancement can benefit both usability and performance, but now I know how much it still matters in 2021.

As my project started from scratch, it started with top performance scores on PageSpeed Insights. While adding content and decoration, I made sure to re-test regularly. My goal: maintain 90% - 100% PageSpeed scores, and "A" or "B" ratings using WebPageTest or GTMetrix:



Latest Performance Report for:
http://www.ingo-steinke.de/

Report generated:	Mon, Feb 15, 2021 10:49 AM -0800
Test Server Location:	Vancouver, Canada
Using:	Chrome (Desktop) 86.0.4240.193, Lighthouse 6.3.0

GTmetrix Grade

	Performance	Structure
A	99%	100%

Web Vitals

LCP	TBT	CLS
0.8s	0ms	0

Google's emphasis on mobile and usability optimization made me go for a progressive enhancement strategy starting with a plain semantic HTML template.

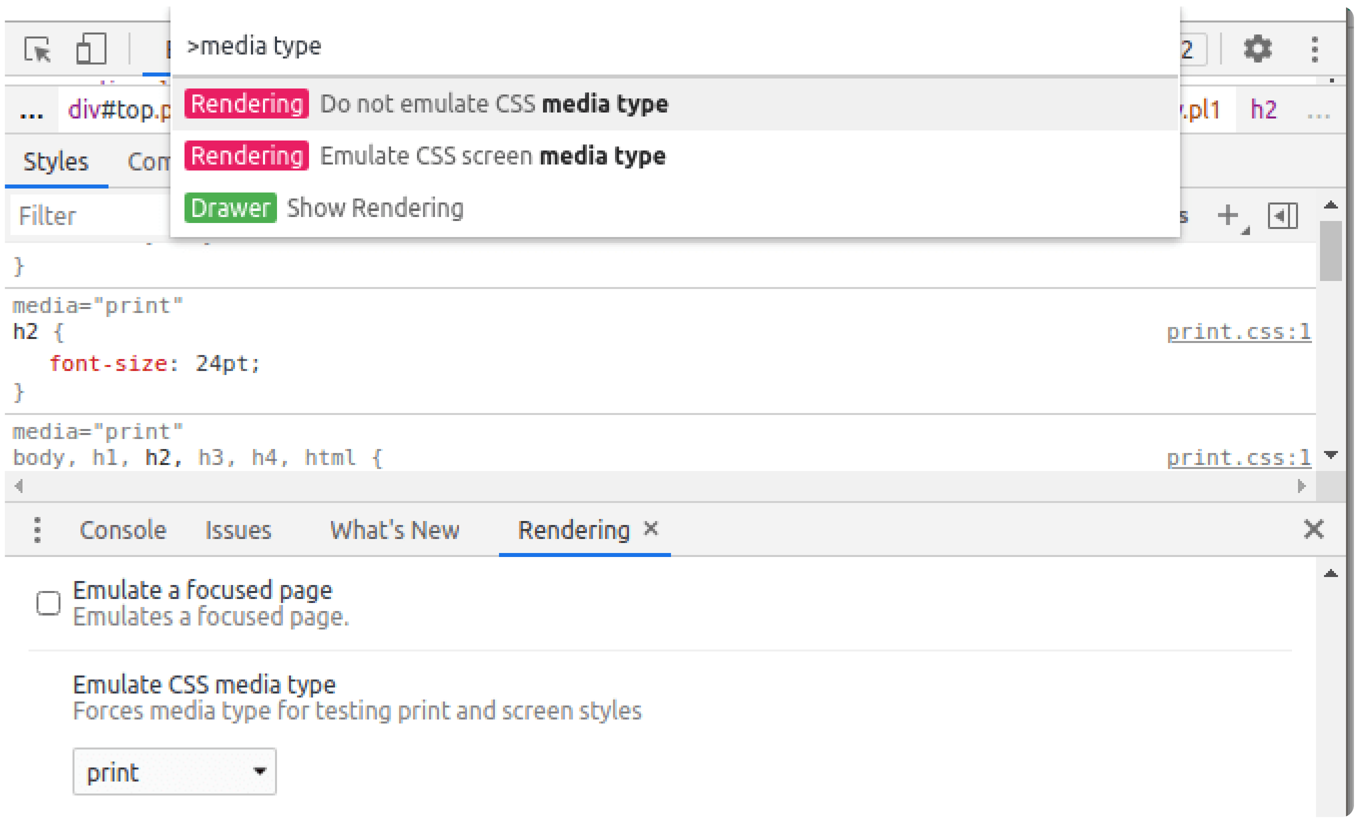
Building Upon the Essentials

Using semantic HTML 5 markup, I put my existing page content into a new template and set up a basic build and deployment toolchain. When adding Style Sheets, I made the occasional check without CSS or rather by looking at its print preview.

Print First, Images Second?

Printing your website is an effective way to inspect your website without interactivity. If you already use native lazy loading in Chrome, you can even print a page without images below the fold (which, in this case, is a bug, not a feature). In the end, my designer advised me to replace my project screenshots with decorative vector icons, due to optical reasons, so all images below the fold are now optional anyway.

While print preview shows a more realistic output, emulating CSS media type print in the browser's developer tools will also allow you to inspect and edit the effective style directives. Media type emulation is a little bit hidden in the console drawer's Rendering tab, so I prefer the quicker "Run Command" (`Ctrl+Shift+P`) and search for "print".


>media type

Rendering Do not emulate CSS **media type**

Rendering Emulate CSS screen **media type**

Drawer Show Rendering

```
}
media="print"
h2 {                                                    print.css:1
   font-size: 24pt;
}
media="print"
body, h1, h2, h3, h4, html {                            print.css:1
```

Console | Issues | What's New | Rendering ×

☐ Emulate a focused page
Emulates a focused page.

Emulate CSS media type
Forces media type for testing print and screen styles

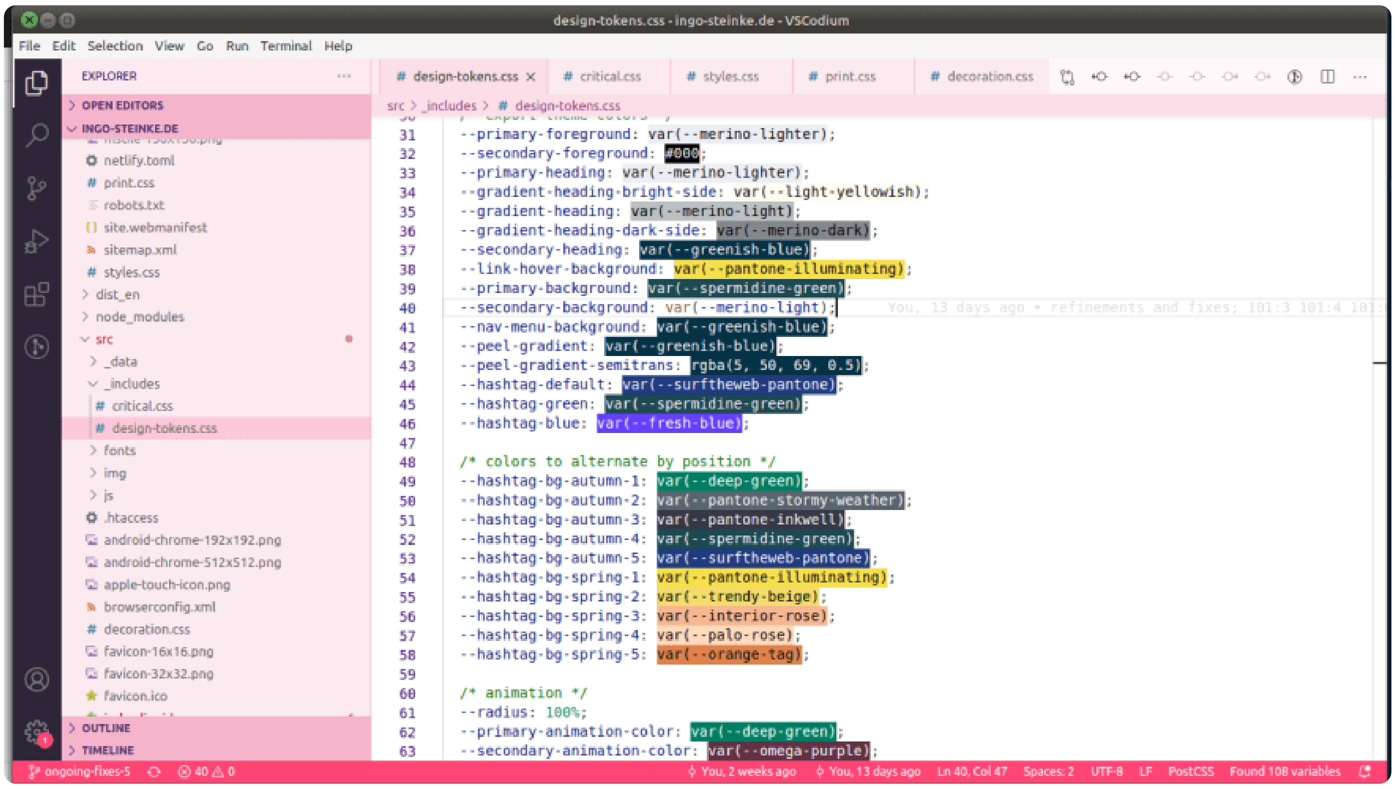
print ▾

Critical Style Sheets

I split my styles into several files. Critical styles, that are necessary for layout, colours and typography, will be included inline. Most other styles will be loaded depending on the media type, screen or print, and the optional animation is deferred until page load.

Splitting files, defining media types, and respecting user preferences like [prefers-reduced-motion]() will spare the browser loading and parsing style definitions that might not be needed at all.

The included file for colour definitions will only be processed by PostCSS, so I can use custom properties (sometimes also known as CSS variables) in my code while still exporting backwards-compatible code without dynamic properties and unnecessary `calc()` calculations.



```css
  --primary-foreground: var(--merino-lighter);
  --secondary-foreground: #000;
  --primary-heading: var(--merino-lighter);
  --gradient-heading-bright-side: var(--light-yellowish);
  --gradient-heading: var(--merino-light);
  --gradient-heading-dark-side: var(--merino-dark);
  --secondary-heading: var(--greenish-blue);
  --link-hover-background: var(--pantone-illuminating);
  --primary-background: var(--spermidine-green);
  --secondary-background: var(--merino-light);
  --nav-menu-background: var(--greenish-blue);
  --peel-gradient: var(--greenish-blue);
  --peel-gradient-semitrans: rgba(5, 50, 69, 0.5);
  --hashtag-default: var(--surftheweb-pantone);
  --hashtag-green: var(--spermidine-green);
  --hashtag-blue: var(--fresh-blue);

  /* colors to alternate by position */
  --hashtag-bg-autumn-1: var(--deep-green);
  --hashtag-bg-autumn-2: var(--pantone-stormy-weather);
  --hashtag-bg-autumn-3: var(--pantone-inkwell);
  --hashtag-bg-autumn-4: var(--spermidine-green);
  --hashtag-bg-autumn-5: var(--surftheweb-pantone);
  --hashtag-bg-spring-1: var(--pantone-illuminating);
  --hashtag-bg-spring-2: var(--trendy-beige);
  --hashtag-bg-spring-3: var(--interior-rose);
  --hashtag-bg-spring-4: var(--palo-rose);
  --hashtag-bg-spring-5: var(--orange-tag);

  /* animation */
  --radius: 100%;
  --primary-animation-color: var(--deep-green);
  --secondary-animation-color: var(--omega-purple);
```

CSS Can Do That?

Although I knew I did miss some web development progress in the past eleven years, I thought I knew CSS quite well, which is why I wanted to make heavy use of it in the first place. I found out, that you can really build a beautiful, responsive and, to some intent, interactive website with just HTML and CSS and no JavaScript at all.

CSS Grid with Flexbox Fallback

This CSS grid definition includes a flexbox fallback. By design, grid layout overrides flex layout, so modern browsers will ignore any flex behaviour of the grid and its child elements. This works good enough even in Internet Explorer 10 and 11, so we do not need to specify any `-ms-grid-` directives, except for a one-column mobile view.

Code

```
.grid-container {
  display: flex;
  display: grid;
  clear: both;
  grid-template-columns: repeat(auto-fit, minmax(440px, 1fr));
  grid-column-gap: 3em;
  grid-row-gap: 2em;
  flex-wrap: wrap;
  justify-content: space-between;
}

.grid-container > * {
  flex-basis: 49%;
  margin-bottom: 2em;

  @supports (display: grid) {
    margin-bottom: 0;
  }
}
```

As a fallback for browsers that neither support `grid` nor `flex`, you can define `float`, but then you might have to `clear` the floats, like I had to do in the above example).

Once an item becomes a flex or grid child, its float will be ingored by modern browsers.

No Intrinsic Web Design

The idea of intrinsic web design, defining responsive layouts by context rather than viewport width, is impossible to achieve without container queries. At least I managed to use all but one breakpoint thus avoiding all the fine-tuning pixel-pushing that used to go along with it.

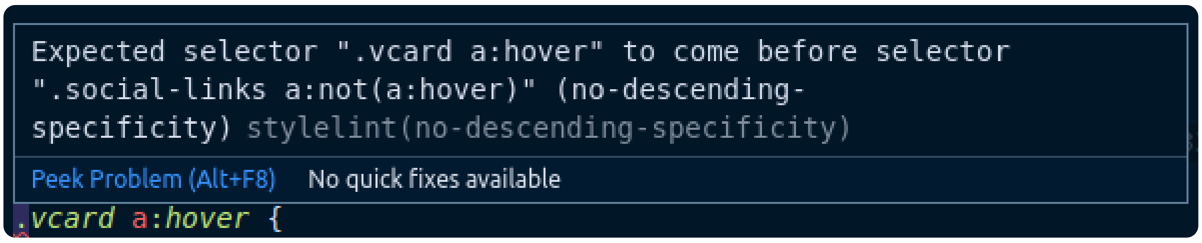
Reset gutters on small viewports:

```
@media only screen and (max-width: 440px) {
  .grid-container {
    grid-template-columns: 1fr;
    -ms-grid-columns: 1fr;
    grid-column-gap: 0;
    grid-row-gap: 1em;
  }
}
```

Specificity Issues

Specificity Issues

I was also impressed by StyleLint showing me warnings about unintended specificity issues.



```
Expected selector ".vcard a:hover" to come before selector
".social-links a:not(a:hover)" (no-descending-
specificity) stylelint(no-descending-specificity)
Peek Problem (Alt+F8)    No quick fixes available
.vcard a:hover {
```

I have to admit that even those few lines of code can soon grow to a complexity that is hard to handle, especially with arbitrary class names that do not follow any naming concept like BEM or CUBE CSS.

Step by Step

Small stories and trunk-based development helped me not to sink into chaos. Every major component, like the grid layouts for projects and skills, the sticky header, and the mobile menu, has a small preview file, that can serve as a minimal verifiable example to copy to Codepen or StackOverflow when external help is needed.

Keeping it Simple

I have seen so many overly complex code that I stopped blaming myself when I fail to understand something easily. Simplicity and robustness have made HTML and CSS outlive many other technologies.

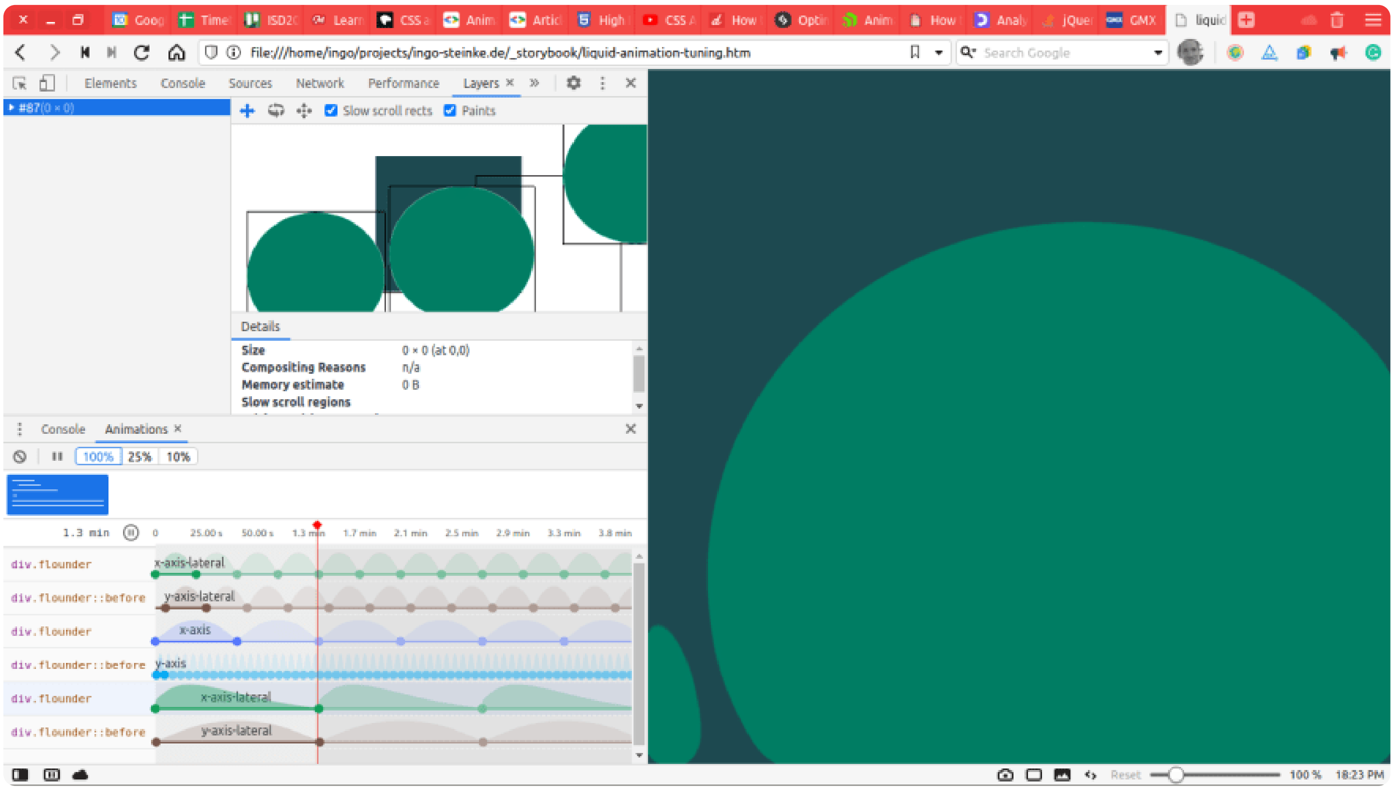
Building a small project on your own lacks the challenge of constant code review. But you do not have to rely on your being disciplined to write maintainable code. We have got tools!

Inspect and Audit

Use automated tools like Lighthouse, axe, and your browser's developer tools to find issues and errors early to ensure your site does not completely fail when it comes to speed, usability and accessibility.

Chrome and Firefox offer detailed inspections of a website's grid and flexbox layouts, web fonts and animations. Some are hard to find, unless you search for them in "Run Command" (Ctrl+Shift+P), like layers, animations, and media type emulation.

Layers and Animations in Developer Tools



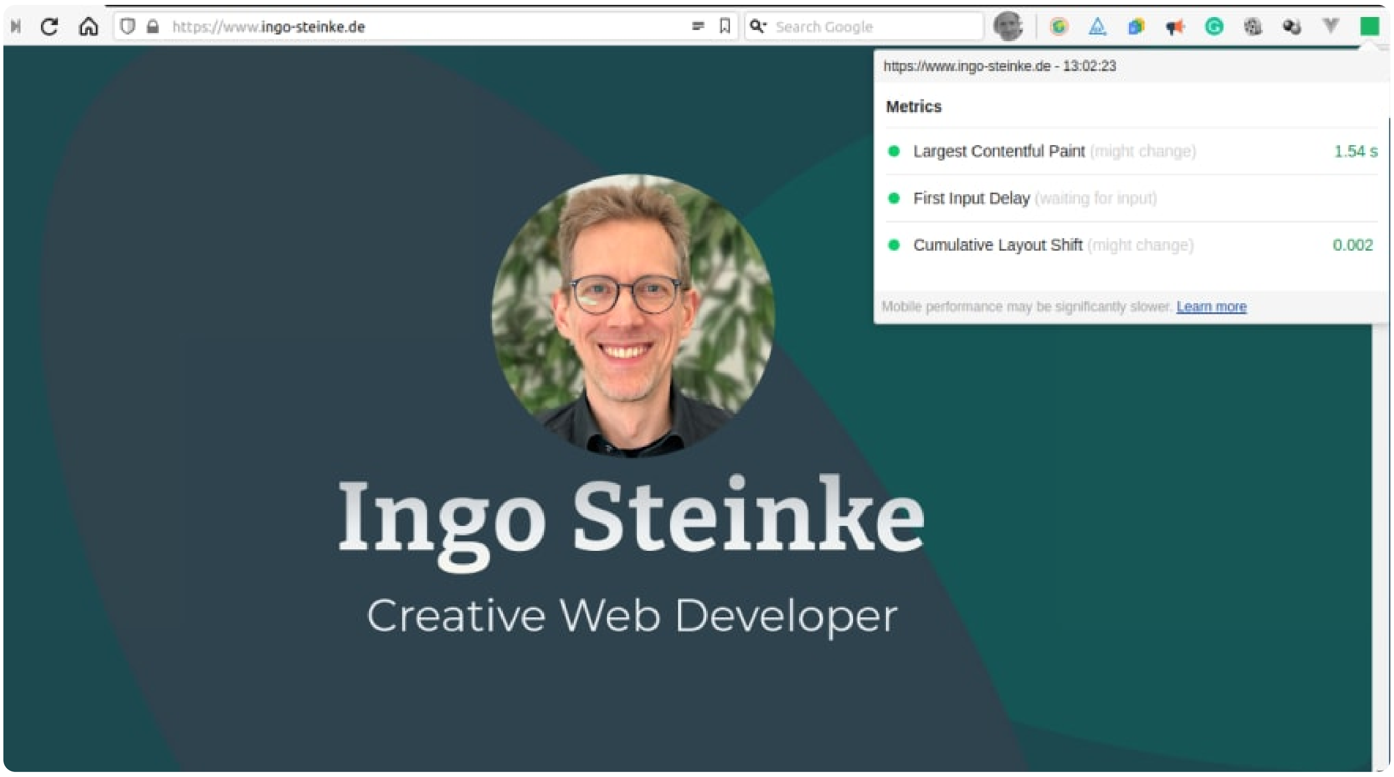
Page speed measurement can vary erratically, so do not forget to click "re-analyze" on Google's PageSpeed Insights results. At webpagetest.org you can specify the number of test runs and many other browser settings in advance. They also offer a wide range of browsers and actual devices in many places around the globe. If you prefer a more polished design, GTmetrix might be for you. Another alternative, Yellow Lab Tools, does not highlight Core Web Vitals (yet), but it does show some unique metrics like DOM complexity, JavaScript errors, or CSS redundancy.

Web Fonts causing Layout Shift

Ironically, some of the advice you will find in Google's Lighthouse discourages code created by other Google tools!

Using font-display swap to prevent a flash of invisible text (FOIT) will cause a flash of unstyled text (FOUT) instead which in turn will add to the cumulative layout shift. Relying on web-safe fonts only would mean missing out on a lot of appealing typographic options.

I did not find a perfect solution yet, but using a full-height header as a cover of my website helped me reduce layout shift to nearly zero.



https://www.ingo-steinke.de - 13:02:23

Metrics

●	Largest Contentful Paint (might change)	1.54 s
●	First Input Delay (waiting for input)	
●	Cumulative Layout Shift (might change)	0.002

Mobile performance may be significantly slower. Learn more

Content: the Hard Parts

Writing about myself was harder than any technological challenge. But when I want people to work with me, I should give them a reason to trust me. Also, I wanted to create some more content to feed the search engines.

From Full-Stack to Creative Web Developer

Am I a full-stack developer? I have worked with front-end technology like HTML, CSS and JavaScript, as well as back-end technology like PHP, Symfony, SQL, and I even set up local development stacks and deployments to Jenkins, Rancher, or Netlify.

Once a good friend told me "if you call yourself a full-stack developer, you will only get back-end jobs", I changed my claim to be a front-end developer. As a result, job offers changed to React and Angular developer.

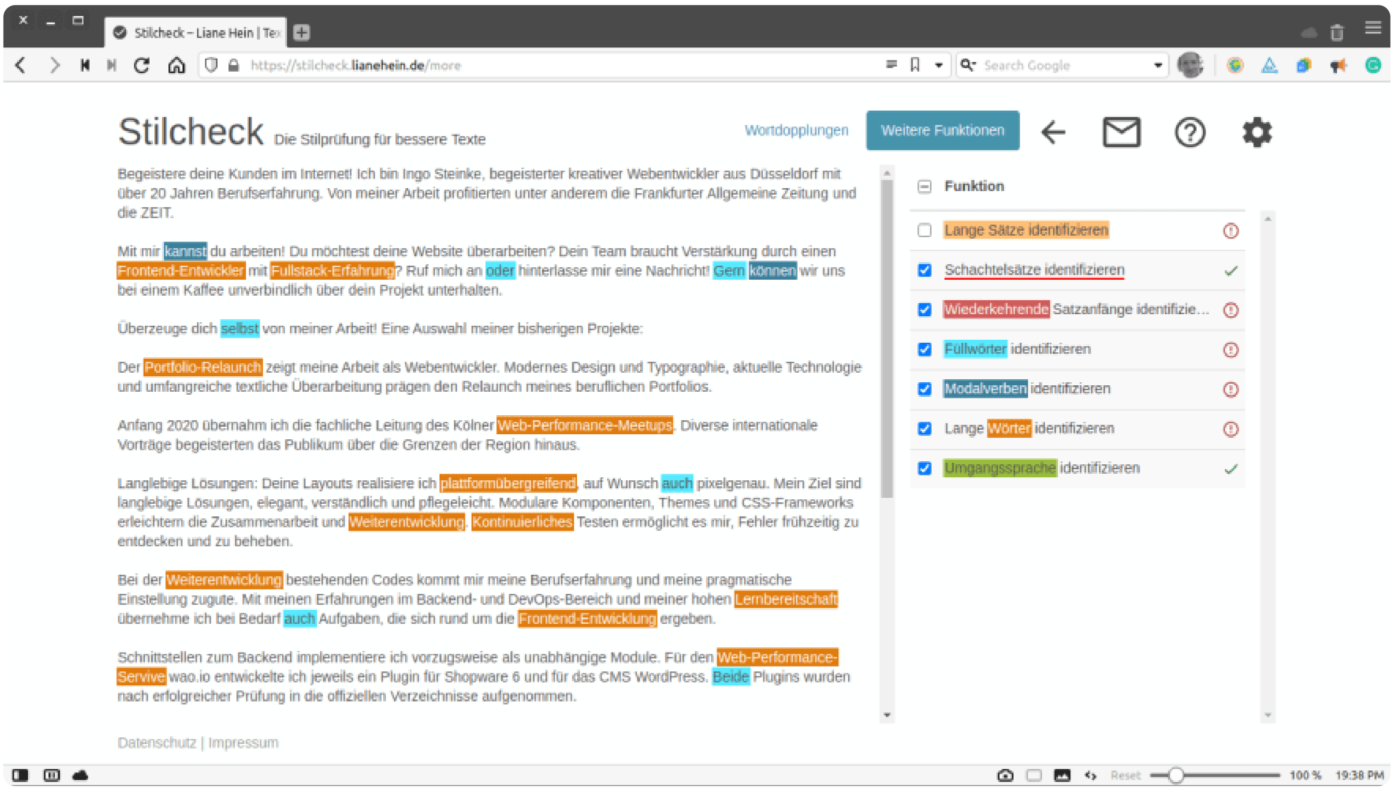
So I started to examine what exactly made me feel grumpy about those React projects of the past and what kind of projects would possibly make me happy and motivated in the future, like building beautiful websites on a minimalistic technology stack.

While my portfolio's web design and technology stack should speak for itself, I still had to find the right words to express my stance and came up with "creative web developer"

Style Checking

Style Checking

Stilcheck, an online tool to improve German language style, shamed my first drafts by putting every imaginable criticism in nearly each of my sentences. One of the few things they do not check is spelling, so I ended up publishing a nice text full of misspellings before I noticed.



Stilcheck Die Stilprüfung für bessere Texte

Begeistere deine Kunden im Internet! Ich bin Ingo Steinke, begeisterter kreativer Webentwickler aus Düsseldorf mit über 20 Jahren Berufserfahrung. Von meiner Arbeit profitierten unter anderem die Frankfurter Allgemeine Zeitung und die ZEIT.

Mit mir kannst du arbeiten! Du möchtest deine Website überarbeiten? Dein Team braucht Verstärkung durch einen Frontend-Entwickler mit Fullstack-Erfahrung? Ruf mich an oder hinterlasse mir eine Nachricht! Gern können wir uns bei einem Kaffee unverbindlich über dein Projekt unterhalten.

Überzeuge dich selbst von meiner Arbeit! Eine Auswahl meiner bisherigen Projekte:

Der Portfolio-Relaunch zeigt meine Arbeit als Webentwickler. Modernes Design und Typographie, aktuelle Technologie und umfangreiche textliche Überarbeitung prägen den Relaunch meines beruflichen Portfolios.

Anfang 2020 übernahm ich die fachliche Leitung des Kölner Web-Performance-Meetups. Diverse internationale Vorträge begeisterten das Publikum über die Grenzen der Region hinaus.

Langlebige Lösungen: Deine Layouts realisiere ich plattformübergreifend, auf Wunsch auch pixelgenau. Mein Ziel sind langlebige Lösungen, elegant, verständlich und pflegeleicht. Modulare Komponenten, Themes und CSS-Frameworks erleichtern die Zusammenarbeit und Weiterentwicklung. Kontinuierliches Testen ermöglicht es mir, Fehler frühzeitig zu entdecken und zu beheben.

Bei der Weiterentwicklung bestehenden Codes kommt mir meine Berufserfahrung und meine pragmatische Einstellung zugute. Mit meinen Erfahrungen im Backend- und DevOps-Bereich und meiner hohen Lernbereitschaft übernehme ich bei Bedarf auch Aufgaben, die sich rund um die Frontend-Entwicklung ergeben.

Schnittstellen zum Backend implementiere ich vorzugsweise als unabhängige Module. Für den Web-Performance-Servive wao.io entwickelte ich jeweils ein Plugin für Shopware 6 und für das CMS WordPress. Beide Plugins wurden nach erfolgreicher Prüfung in die offiziellen Verzeichnisse aufgenommen.

Funktion

- ☐ Lange Sätze identifizieren ⓘ
- ☑ Schachtelsätze identifizieren ✓
- ☑ Wiederkehrende Satzanfänge identifizie… ⓘ
- ☑ Füllwörter identifizieren ⓘ
- ☑ Modalverben identifizieren ⓘ
- ☑ Lange Wörter identifizieren ⓘ
- ☑ Umgangssprache identifizieren ✓

Internationalization

To my surprise, adding a second language to my portfolio proved a major task in many aspects, besides providing the actual translations.

Another Domain for Each Language

Domains or Directories? At least, in this case, Google offers some clear advice as to the world's most popular search engine: alternative content in another language is best served on another domain, using semantic markup to indicate their relationship, like this:

Code

```
<html lang="en">
<head>
  <meta name="language" content="en">
  <link rel="alternate" hreflang="de" href="https://www.ingo-steinke.de/">
  <link rel="canonical" href="https://www.ingo-steinke.com/">
```

Hard to believe, but I had a German website for eleven years without ever thinking about registering the matching `.com` domains. Neither did the other Germans that go by the same name, so I registered my additional domains and wondered which would be the best choice of a canonical URL?

`www.ingo-steinke.com`?

`www.ingosteinke.com`?

`ingosteinke.com`?

"Worldwide Web"

Long gone the day when everyone expected worldwide web domains to start with "www", but doing so still has a practical benefit. If you want to point your name server to a content delivery network or a "serverless" server, it is much easier to add a CNAME for a subdomain than to switch the A record.

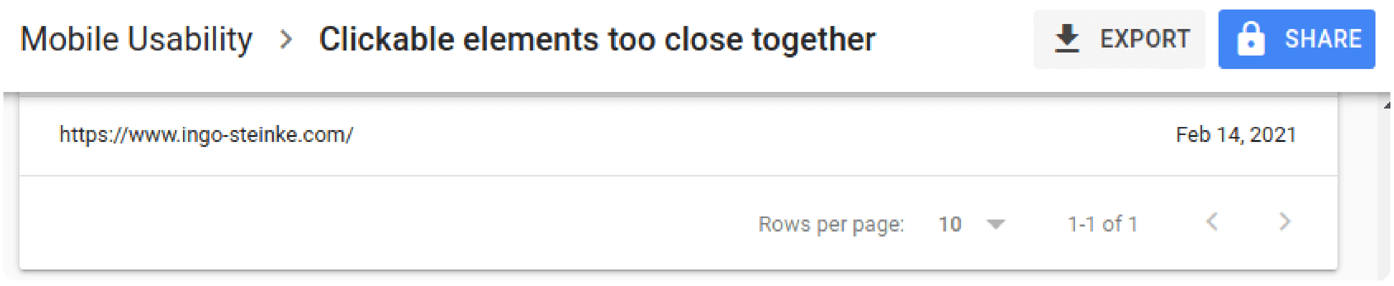
German Idiosyncrasies

German grammar dictates to create new words out of compound nouns, so front-end development becomes "Frontendentwicklung" in German - unless when registering domain names. Germans and Japanese tend to favour domains that use a hyphen wherever possible.

Apart from staying consistent with my established URL, www.ingo-steinke.de, separating given names and family names for better readability is one of the few valid reasons to use a hyphen in an international domain, so I decided to use www.ingo-steinke.com to publish my English content.

Google Search Console

Due to the increasing role of mobile usability for search engine optimization, Google's Lighthouse and Search Console try to help webmasters identify and fix issues found on their websites ... well, at least a little bit.



https://www.ingo-steinke.com/ Feb 14, 2021

Rows per page: 10 1-1 of 1

Content Wider Than Screen?

Google was helpful enough to point out what kind of issue is found on which URL, but it would not tell me, what element caused the error. Fair enough, it often needs manual it would not tell me, what element caused the error. Fair enough, it often needs manual inspection, maybe even a little bit of intuition, to fix "Content wider than screen", but if

they find some "clickable elements too close together" to be clicked conveniently on a mobile, they might just tell me the problematic elements.

Perfectly Imperfect

Automated Tests using Cypress and Percy will hopefully make it safe to refactor my setup when I find some time.

Discussion (1)

Ingo Steinke • Oct 19 '21

Another possible improvement that had already been on my to-do-list, was splitting up the application into more components. Using [fractal.build as an atomic design tool]() could be a helpful, as I only recently found out (after having given up on using Storybook for a project as small as this one).

Ingo Steinke

Web Development, Sustainability, Art and Music, Nature and Travel, Sustainability

LOCATION
Germany

WORK
Creative Web Developer at Ingo Steinke

JOINED
21 Sep 2019

CSS :has(.parent-selectors)

#css #webdev #todayilearned #javascript