

LEISTET MEIN WEBDESIGNER GUTE ARBEIT?

Liebe Kolleg*innen, auch wenn ich mich möglicherweise bei Webdesignern unbeliebt mache, möchte ich Euch vermitteln, wie man auch als Laie überprüfen kann, wie gut der Webdesigner die eigene Homepage programmiert hat.

Ich werde ausschließlich auf die Qualität des Codes eingehen, der die Basis einer jeden Webseite darstellt, nicht aber auf inhaltliche, Textgestaltung etc., Aspekte.

Im Wesentlichen wird es um Suchmaschinenaspekte gehen. Ich möchte Euch eine einfache Einführung in die Basiselemente einer Webseite geben und welche Kriterien bestimmen, ob man bei einer Suchmaschine gut gefunden wird.

Dies alles werde ich am Beispiel der Google Suchmaschine machen und als Überprüfungsstools auch hauptsächlich den Google Browser Chrome nutzen. Dies soll keine Aufwertung der Firma Google sein, sondern dient nur der Veranschaulichung.

GRUNDLAGEN:

Woraus besteht eine Webseite und wer oder was bestimmt, wie am Ende meine Seite in einem Browser aussieht?

1. Der Browser ist das Werkzeug, das bestimmt, wie meine Webseite aussieht und funktioniert. Die bekanntesten Browser sind: Microsoft Edge (früher Internet Explorer), Mozilla Firefox, Google Chrome, Apple Safari, und noch viele mehr. Deshalb kann es passieren, dass eine programmierte Webseite in unterschiedlichen Browser auch leicht unterschiedlich aussieht. Da alle verschiedenen Browser jedoch recht ähnlich arbeiten, sind die Unterschiede meist sehr klein, oder gar nicht zu sehen. Der Browser interpretiert den Code, mit dem eine Webseite erstellt wurde und „weiss“ dadurch, wie er diese Seite optisch und funktional darstellen soll. Dieser Prozess der Interpretation des Codes nennt man „rendern“. Der Browser rendert die Seite also.

2. Der Code, aus dem eine Internetseite erstellt wird.

Es sind meist drei verschiedene Programmiersprachen, die für die Erstellung einer Internetseite benutzt werden. Es gibt zwar noch andere Programmiersprachen, mit der man Internetseiten erstellen kann, diese werden jedoch nach der Programmierung in die drei Sprachen übersetzt (kompiliert), mit denen wir uns nun beschäftigen.

HTML, ODER AUCH HTML5, WIE MAN DEN AKTUELLEN STANDARD NENNT.

Html ist eine sogenannte „Auszeichnungssprache“ und damit nicht im eigentlichen Sinn eine Programmiersprache. Der html-Code enthält im Wesentlichen die eigentlichen Inhalte einer Internetseite. Dort sind konkret die Texte und Bilder gespeichert, die dann auf der Internetseite erscheinen.

Diese Inhalte werden von Befehlen umschlossen, die dem Browser mitteilen, um was für Inhalte es sich handelt.

Eine solche Codezeile könnte so aussehen:

```
<h1>Überschrift</h1>
```

Hierdurch wird dem Browser signalisiert: „Schreibe das Wort „Überschrift“ und behandle es als eine Überschrift (h = headline) 1. Ordnung, also als die wichtigste Überschrift der gesamten Webseite. Insgesamt kann ein Browser 6 verschiedene „Wichtigkeitsstufen“ solcher Überschriften unterscheiden:

h1, h2, h3, h4, h5, h6.

H1 ist die wichtigste und darf nur ein einziges Mal im gesamten Code einer Internetseite vorkommen. Diese Überschrift wird später auch in den Suchmaschinenergebnissen angezeigt.

Die Sprache html ist so aufgebaut, dass man Texte oder Bilder immer mit solchen Befehlen umschließt. Man sagt dem Browser also, wann er mit einer solchen Überschrift anfangen soll „<h2>“ und wann er damit wieder aufhören soll „</h2>“. Deshalb spricht man von einer Auszeichnungssprache, weil man die Teile einer Internetseite mit diesen Befehlen auszeichnet. Diese Befehle (u.a. <h1>) nennt man „html-Tags“

Ihr sollt zwar keine Programmierexperten werden, dennoch muss ich Euch noch etwas über diese html-Tags schreiben, da diese großen Einfluss auf die Suchmaschinenergebnisse haben.

Diese html-Tags sagen dem Browser, und damit den Suchmaschinen, mit welchem inhaltlichen Teil einer Webseite sie es zu tun haben. Die Überschriften habt Ihr ja bereits kennengelernt (h1-h6).

Weitere sind:

```
<head> ... </head>
```

Alles was zwischen diesen beiden html-Tags steht, wird nicht auf der Internetseite angezeigt. Hier stehen auf der einen Seite technische Aspekte drin, die der Browser braucht, um die Seite richtig anzuzeigen, aber auch inhaltliche Aspekte, wie der Titel und die Beschreibung einer Seite. Letzteres ist sehr wichtig für die Suchmaschinen.

`<body> ... </body>`

Hier steht alles drin, was der Browser später tatsächlich anzeigen soll. Hier zwischen stehen auch die kommenden html-Tags drin, die dem Browser sagen, aus welchen inhaltlichen Teilen eine Webseite besteht.

`<header> ... </header>`

Hier findet man in der Regel die Überschrift, die angezeigt werden soll, das Logo und oft auch das Hauptmenü.

`<nav> ... </nav>`

Hier findet man meist das Hauptmenü, womit man sich zu den Unterseiten klicken kann.

`<main> ... </main>`

Hier findet man die eigentlichen Inhalte der Internetseite. Die Texte, Artikel, Bilder, Formulare, Videos. Bei Blogs sind diese zusätzlich von `<article> ... </article>` umschlossen, damit der Browser weiss, um was für eine Art von Inhalt es sich handelt. Das gibt es auch für Bilder ``, für Links `<a>`, für Videos `<video>` und so weiter.

`<aside> ... </aside>`

Hier findet man Zusatzinformationen, die zwar zum Haupttext gehören, aber nicht direkt in diesem drin stehen sollen. Diese werden oft in einer Seitenleiste dargestellt (Sidebar).

`<footer> ... </footer>`

Hier befindet sich die Fusszeile einer Internetseite, in der oft die rechtlichen Inhalte dargestellt werden: Link zu Startseite, Impressum und Datenschutz; Hinweise zum Autor und oder Eigentümer einer Internetseite.

Ein verkürzter kompletter Quellcode einer solchen Internetseite könnte dann so, oder so ähnlich aussehen:

```
<!doctype html>
```

```
<html lang="de">
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
  <title>Titel der Webseitetitle</title>
```

```
  <link rel="stylesheet" type="text/css" href="main.css" />
```

```
  <meta name="description" content="">
```

```
  <link rel="icon" href="/favicon.ico" sizes="any">
```

```

</head>

<body>
  <header class="kopfzeile" aria-label="Hauptkopfzeile">
    <h1>Überschrift</h1>
    <nav class="hauptnavigation" aria-label="Hauptnavigation">
      <a>home</a>
      <a>unterseite</a>
    </nav>
  </header>

  <main id="content" class="hauptbereich" aria-label="Hauptinhalt">
    <h2>Überschrift eines Artikels</h2>
    <p>Hier kommt der eigentliche Text rein</p>
    <img>hier könnte ein Bild sein
  </main>

  <aside id="sidebar">
    <p>Hier kann ergänzender Text für den Haupttext stehen</p>
  </aside>

  <footer id="footer" class="webseitenfuss" aria-label="Fußzeile">
    <p>© 2024 Sozialdienstleister-Webseite. Alle Rechte vorbehalten.</p>
  </footer>

</body>
</html>

```

Ihr seht in diesem „Quellcode“ zusätzlich zu den bekannten html-Tags, dass darin noch weitere Anweisungen enthalten sind. Diese Anweisungen, oder Befehle, vermitteln dem Browser weitere Informationen über den jeweiligen Inhalt. Sie haben aber zusätzlich die Aufgabe, für die Barrierefreiheit der Seite zu sorgen. Manche dieser Befehle können von Vorlesegeräten interpretiert werden und diese Geräte „wissen“ dann, dass sie nun ein Menü, eine Überschrift, oder einen Text vorlesen sollen.

Abschliessend zum Thema html, und ich verspreche Euch, dass die nächsten beiden Programmiersprachen deutlich kürzer vorgestellt werden, könnt Ihr einmal folgendes probieren:

Öffnet in irgendeinem Browser irgendeine Internetseite; klickt irgendwo in einen Text mit der rechten Maustaste und klickt dann im Kontextmenü auf „Seitenquelltext anzeigen“. Dann solltet Ihr in einem neuen Tab den kompletten Quellcode der Internetseite sehen, die Ihr gerade geöffnet habt.

Dieser wird viel länger und komplexer sein, als mein Beispiel oben, aber sucht doch einmal, ob Ihr die html-Tags wieder findet, die ich Euch erklärt habe. Mit „Strg + f“ bekommt Ihr auch eine Suchmaske und könnt nach diese Wörtern (Tags) suchen.

CSS CODE

Die zweite Programmiersprache die benötigt wird, ist der CSS Code; man nennt diese Code auch Stylesheet. Dieser Code bestimmt im Wesentlichen, wie die Inhalte, die bekanntlich mit html erzeugt werden, aussehen sollen. Über Befehle kann ich bestimmen, welche Farben benutzt werden sollen, wie groß die Schriftart, oder eine Überschrift ist, ob es Hintergrundfarben geben soll, wie groß die Bilder sein sollen und schlussendlich auch das gesamte Layout einer Internetseite: Soll einfach alles untereinander sein? Soll es eine Seitenleiste geben?

Nur in der Kürze zeige ich Euch an wenigen Beispielen, wie das funktioniert, damit Ihr eine Vorstellung habt. Das CSS ist nur ganz wenig relevant für die Suchmaschinen.

Eine Codezeile könnte hier folgendermaßen aussehen:

```
h1 {  
    font-size: 30px;  
    color: #e2001A;  
}
```

Hiermit sage ich dem Browser, dass alles, was zwischen <h1> ... </h1> im html-Code steht eine Schriftgröße (font-size) von 30 Pixeln (px) haben soll und in der Schriftfarbe (color) AWO-Rot erscheinen soll. #e2001A ist der Hexadezimalcode für unser AWO Rot.

Der Code im CSS ist immer gleich aufgeteilt. Zuerst bestimme ich, auf welches Element ich Einfluss nehmen möchte, hier die h1 Überschrift, und zwischen die geschweiften Klammern schreibe ich, welche Stileigenschaften wie aussehen sollen. Ich kann so viele Eigenschaften zwischen die geschweiften Klammern schreiben, wie ich will. Zur Plastizität ein letztes Beispiel und ich zeige Euch zur Verdeutlichung einen kurzen html-Code und den dazugehörigen CSS Code:

Html-Code

```
<header>  
    <h1>Digitales Netzwerk AWO</h1>  
    <p>Willkommen auf der Internetseite des digitalen Netzwerkes des AWO  
Bundesverbandes</p>  
</header>
```

Css-Code

```
header {  
  display: block;  
  max-width: 100%;  
  background: #e2001A;  
  padding: 20px;  
  margin: 20px;  
}
```

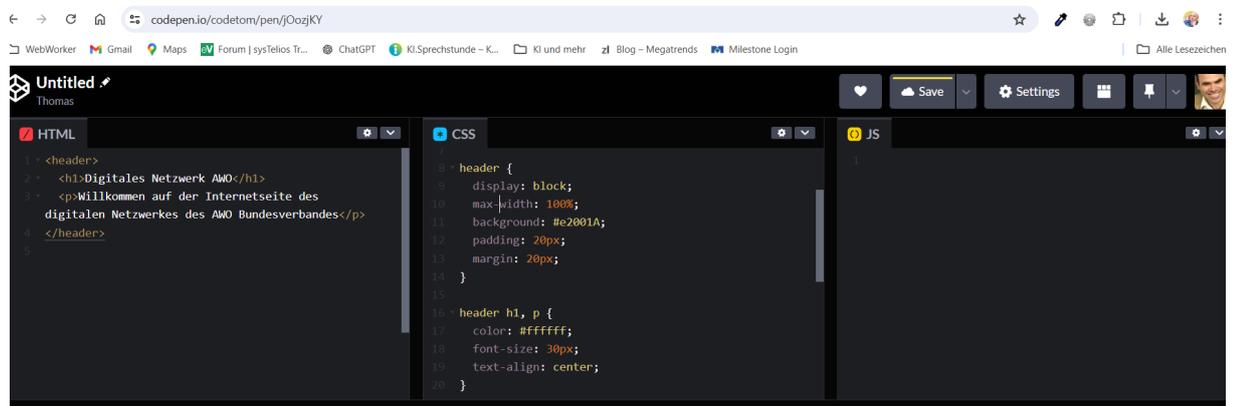
```
header h1, p {  
  color: #ffffff;  
  font-size: 30px;  
  text-align: center;  
}
```

So was passiert hier? Ich mache den kompletten header zu einem Blockelement, d.h. er nimmt die gesamte Breite des Bildschirms im Browser ein; um dies zusätzlich sicherzustellen gebe ich ihm noch die Breite 100%. Dann mache ich dessen Hintergrundfarbe in AWO-Rot. Mit „padding“ mache ich einen Innenabstand und mit „margin“ mache ich einen Außenabstand.

Danach bestimme ich, dass meine Überschrift h1 in weiss dargestellt werden soll (der Hintergrund ist ja rot), eine Größe von 30 Pixeln haben soll, und zentriert (text-align) dargestellt wird.

Ich könnte mir denken, dass diese Befehle, wenn man sie einmal weiss und ein wenig Englischkenntnisse hat, gut zu verstehen sind.

Hier noch ein Bild, wie mein obiger Code in einem Browser aussehen würde. Das Tool, dass ich dazu verwendet habe, werde ich am Ende erklären und verlinken:



Digitales Netzwerk AWO
Willkommen auf der Internetseite des digitalen Netzwerkes des AWO Bundesverbandes

JAVASCRIPT CODE

Hierzu schreibe ich nun nochmals deutlich weniger. Javascriptcode wird genutzt, um interaktive Elemente auf einer Internetseite möglich zu machen. Während html- und css-Code lediglich Inhalte ausgeben können, ist Javascript in der Lage zu erkennen, was der Benutzer, der diese Internetseite gerade liest, tut. Also Javascript kann Euch selbstverständlich nicht beobachten, aber es kann erkennen, auf welche Höhe der Webseite Ihr aktuelle gescrollt habe, ob Ihr mit dem Mauszeiger gerade über einen Link „schwebt“, oder diesen anklickt. Javascript wird auch genutzt, um bestimmte Animationen zu erzeugen.

Oft wird es genutzt, um die Effekte des bekannten „Hamburger Menü“ zu erzeugen; also diese drei horizontalen Striche, die, wenn man auf sie klickt, das Hauptmenü öffnen und sich selbst in ein „X“ verwandeln, damit man das Menü wieder schließen kann.

Abschließend ist noch erwähnenswert, dass CSS-CODE im Laufe der letzten Jahren immer mehr weiter entwickelt wurde, so dass man damit nun auch solche Effekte erzeugen kann, wie es früher nur mit Javascript möglich war. Das ist deshalb so wichtig zu wissen, weil man damit oft weitestgehend auf Javascript verzichten kann. Javascript verlängert die Ladezeit einer Internetseite und kann in wenigen Fällen auch ein Sicherheitsrisiko darstellen. Es gibt Menschen, die aus diesem Grund die Javascript-Funktion ihres Browsers grundsätzlich abschalten. Bei solchen leuten würde dann unser schöner „Hamburger Schalter“ überhaupt nicht mehr funktionieren, und das wäre doof.

Damit bin ich mit meiner Einführung in die drei wesentlichen Programmiersprachen einer Webseite fertig.

WAS BESTIMMT AM CODE DIE SOGENANNT E SUCHMASCHINENFREUNDLICHKEIT MEINER WEBSEITE? UND WIE KANN ICH ÜBER DEN CODE DIESE SUCHMASCHINENFREUNDLICHKEIT VERBESSERN?

DIE GOOGLE SUCHMASCHINE

Leider müssen wir uns nun mit den Prinzipien der Google Suchmaschine, und damit analog zu allen anderen Suchmaschinen, auseinandersetzen.

Warum nehme ich die Google Suchmaschine als Beispiel?

Das ist eine absolut zentrale Frage, vor allem auch, wenn man sich mit ethischen Aspekten dieses Unternehmens auseinandersetzt. Hierbei muss bedacht werden, dass Google seinen Suchalgorhythmus geheim hält. Man kann also nicht überprüfen, ob Google bestimmte Internetseiten bevorzugt.

Dennoch muss gesagt werden, dass Google aktuell immer noch die beste Suchmaschine auf dem Markt ist. Dies ist meines Wissens nach wie vor Konsens in der Internetgemeinde und spiegelt auch meine eigenen Erfahrungen wieder. Ich gehe davon aus, dass sich das jedoch in den nächsten beiden Jahren verändern wird.

Durch die Entwicklung der KI Chatbots (ChatGPT und Co) wird sich bald die komplette Suche im Internet verändern und dann werden wir neue Strategien für die Programmierung unserer Internetseiten etablieren müssen.

Die Prinzipien, mit denen Google unsere Internetseite bewertet (Ranking), zumindest, was den Code unserer Seite angeht, sind tatsächlich ziemlich cool. Die Prinzipien sind nämlich auf den Komfort unserer Kund*innen, also den Leser*innen unserer Internetseite abgestimmt. Google möchte:

- Schnelle Ladezeiten
- Mobile Freundlichkeit
- Sicherheit
- Strukturierte Inhalte über die html-Tags und Einhaltung der Standards
- Barrierefreiheit
- Gute Navigation

Auf diese Kriterien hat man guten Einfluss durch die Qualität seines Webseitencodes:

- Je weniger Code, desto besser
- Je weniger Bilder, desto besser (Ladezeiten)
- Je besser der Code für alle Arten von Endgeräten ist (PC, Tablet, Smartphone, Vorlesegeräte), desto besser
- Barrierefreiheit

Die Prinzipien, an die man sich beim Programmieren halten sollte lauten entsprechend:

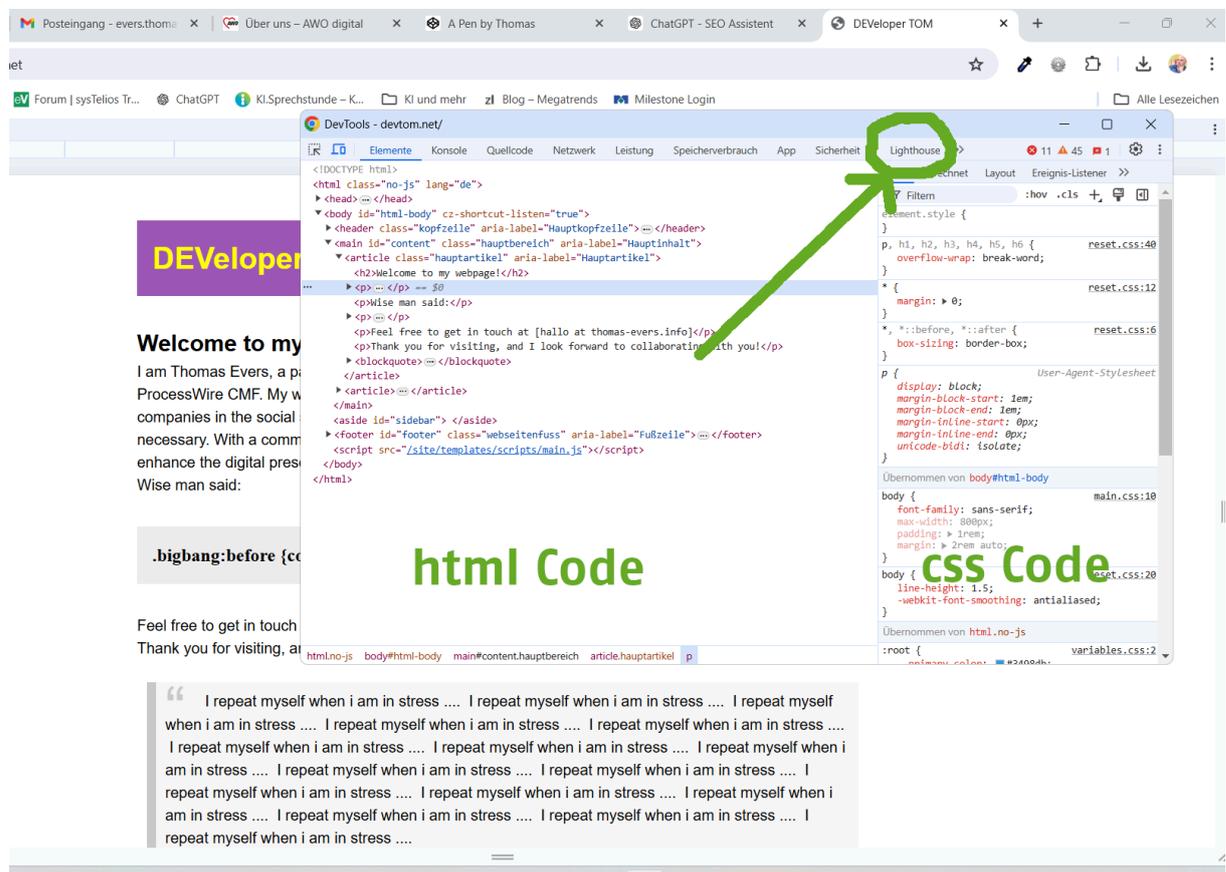
- Programmieren barrierefrei
- Programmieren mobile first
- Schreibe effektiven Code. Also nur genau so viel, wie für das erwünschte Ergebnis absolut notwendig ist
- Dimensioniere die Qualität von Bildern genauso, dass sie auf jedem Gerät nur genauso groß ist, wie erforderlich

- Beschreibe Dein Bild, so dass diese Beschreibung einem blinden Menschen vorgelesen werden könnte
- Mache Deine Webseite sicher (aktivierte Verschlüsselung = https://)

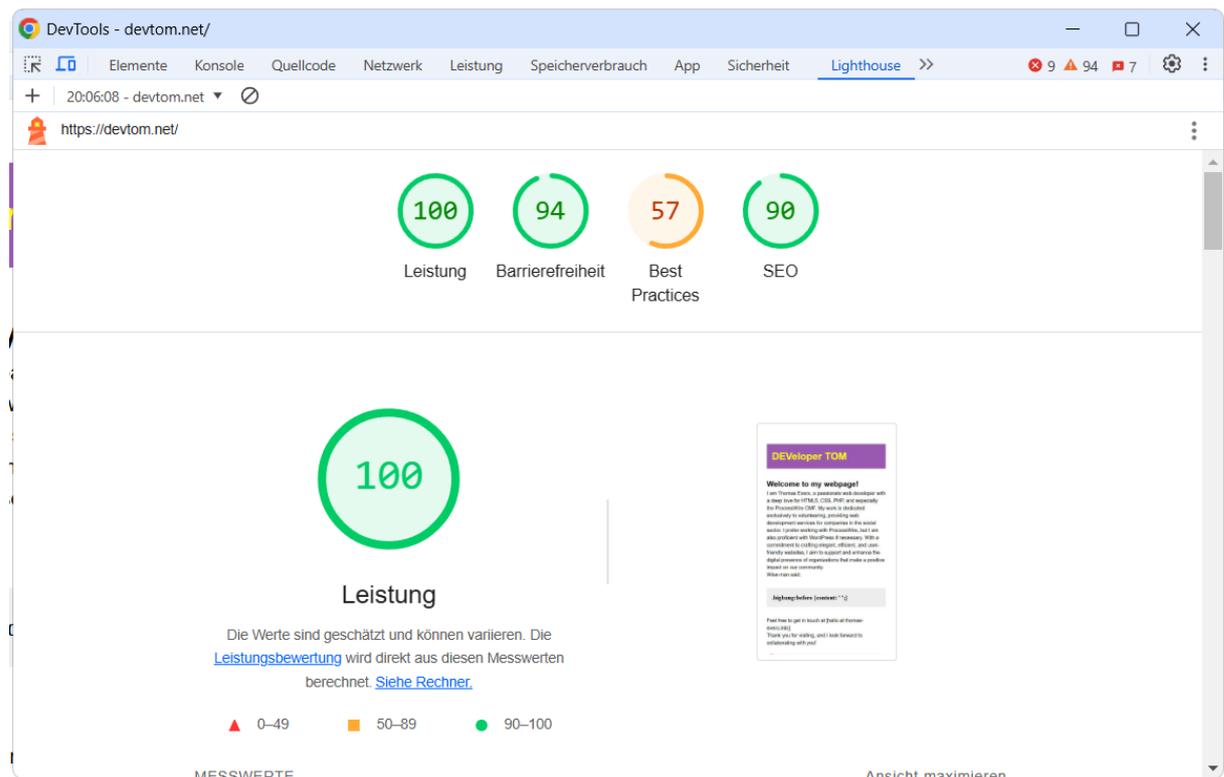
WIE KANN ICH DIESE QUALITÄT, DIESE KRITERIEN ÜBERPRÜFEN?

Es gibt gute Hilfsmittel, um eine Webseite im Hinblick auf die Suchmaschinenfreundlichkeit zu überprüfen. Eine davon habe ich bereits benannt: Man lässt sich den Quellcode einer Webseite anzeigen und schaut sich diesen Code im Hinblick auf das bisher hier Beschriebene an. Es geht aber auch einfacher.

Viele Browser haben die sogenannten „Developer Tools“. Diese öffnet man, indem man auf einer Webseite die rechte Maustaste klickt, und im daraufhin erscheinenden Kontextmenü den Punkt „Untersuchen“ anklickt. Dann sollten sich diese Developer Tools öffnen.



Hier sieht Ihr die Developer Tools. Im linken Bereich befindet sich der html Code und im rechten Bereich den css Code. Wenn man oben auf das Wort „Lighthouse“ klickt, wird eine Analyse der Seite durchgeführt. Im Falle einer meiner eigenen Seiten sieht das Ergebnis dann folgendermaßen aus:



Leistung, betrifft alles, was mit der Ladegeschwindigkeit der Seite zu tun hat. Wieviel Code muss geladen werden? Wieviel zusätzliche Dateien müssen geladen werden (css, javascript, Bilder)?

Barrierefreiheit, erklärt sich möglicherweise von selbst.

Best Practices, betrifft alles, wo man die erforderlichen Standards eingehalten hat.

SEO, betrifft die Qualität der Textinhalte. Hier geht es vor allem um das quantitative Verhältnis (Anzahl der Wörter) von Title Element, Description Element, h-Überschriften, Textpassagen. Wie Ihr seht, sollte ich selbst vor allem im Bereich der Best Practices noch etwas verbessern.

Dieselbe Analyse kann man auch auf einer der Seiten von Google durchführen. Auf der folgenden Internetseite (<https://pagespeed.web.dev/?hl=de>) kann man oben in einem Feld eine x-beliebige Internetadresse eingeben, und wird die entsprechende Seite untersucht.

Bitte beachtet auf jeden Fall, dass man diesen Test mehrmals hintereinander durchführen sollte, da die Ergebnisse in Abhängigkeit der eigenen Internetgeschwindigkeit variieren können.

Mit den Ergebnissen eines solchen Tests weiss ich selbst, was ich am Code meiner Internetseite noch verbessern muss, oder Ihr könnt Euren Webdesigner mit den Testergebnissen konfrontieren und ihn oder sie bitten, den Code zu verbessern.

Zusammenfassend lässt sich sagen, dass eine gut gestaltete Webseite auf drei Säulen beruht: strukturiertem HTML für den Inhalt, stilvollem CSS für das Design und funktionalem

JavaScript für die Interaktivität. Die Qualität dieser Elemente beeinflusst maßgeblich, wie gut Ihre Seite von Suchmaschinen gefunden und von Besuchern genutzt wird.

Mit den vorgestellten Tools wie den Developer Tools und dem Google PageSpeed Test können Sie die Leistung Ihrer Webseite selbst überprüfen. Nutzen Sie dieses Wissen, um konstruktiv mit Ihrem Webdesigner zusammenzuarbeiten und eine optimale Online-Präsenz zu schaffen.

Denken Sie daran: Eine gute Webseite ist nie "fertig". Sie entwickelt sich ständig weiter, passt sich neuen Technologien und Nutzergewohnheiten an. Bleiben Sie neugierig und informiert über die neuesten Trends im Webdesign.

Für alle, die tiefer in die Materie einsteigen möchten, empfehle ich Ressourcen wie die Mozilla Developer Network (MDN) Dokumentation oder Online-Kurse auf Plattformen wie Codecademy oder freeCodeCamp.