



Kai Priestersbach, Carsten Hinrichs

Web Vitals rechtzeitig optimieren

So bekommen Sie die angekündigten neuen Rankingsignale LCP, FID und CLS in den Griff!

Googles frühzeitig angekündigtes Page-Experience-Update steht nun unmittelbar vor der Tür. 2021 wird die Ladezeit in Gestalt der Core Web Vitals endgültig zum Rankingfaktor und damit zum Optimierungsgegenstand für Suchmaschinenoptimierer. In Ausgabe #63 konnten Sie bereits lesen, wie sich die drei neuen Metriken LCP, FID und CLS zusammensetzen und wie diese durch Nutzer erhoben werden. Ein weiterer Artikel in derselben Ausgabe zeigte, wie diese sich für eine größere Anzahl von URLs erheben lassen.

In diesem Beitrag finden Sie nun eine Anleitung für die Analyse von Problemen mit den Core Web Vitals. Die gezeigten Techniken helfen, die Ursachen zu identifizieren, und zeigen konkrete Ansätze für Verbesserungen auf, mit denen die Geschwindigkeit moderner Webseiten effektiv gesteigert werden kann.

DER AUTOR



Kai Priestersbach ist Inhaber des Online-Magazins SEARCH ONE und freier Mitarbeiter im Research & Development bei der Online-Marketing-Agentur eology GmbH. Sein Schwerpunkt liegt in der Aus- und Weiterbildung der Mitarbeiter.

DER AUTOR



Carsten Hinrichs ist Inhaber des Online-Marketing-Unternehmens Hinrichs Medien, mit dem er verschiedene Webportale betreibt. Der Fokus seiner Tätigkeit liegt im Bereich Planung & Entwicklung von Strategien und Softwarelösungen.

Wie groß der Einfluss der Core Web Vitals auf das Ranking ausfallen wird, darüber lässt sich derzeit nur spekulieren. Vollkommen klar ist jedoch, wie wichtig eine positive Benutzererfahrung mit den rankenden Webseiten aus Sicht der Suchmaschine ist. Sowohl Googles interne Studien (<http://einfach.st/speed5>) als auch unabhängige Forscher (<http://einfach.st/webvitals3>) konnten in der Vergangenheit deutlich nachweisen, dass Benutzer generell Seiten mit einer großartigen User Experience bevorzugen und diese nicht nur häufiger aufsuchen, sondern auch intensiver nutzen, ihnen größeres Vertrauen entgegenbringen und sogar die Konversionsraten mit kürzeren Ladezeiten und einer besseren Nutzerführung steigern. Eine gute Seitenerfahrung ersetzt selbstverständlich nicht, großartigen und relevanten Inhalt anzubieten, allerdings kann sie den Unterschied machen, wenn es mehrere Seiten mit ähnlichem Inhalt gibt.

Es kann also angenommen werden, dass ein schlechtes Abschneiden im sogenannten Core-Web-Vitals-Test zukünftig negativen Einfluss auf das Ranking der betreffenden URL haben wird. Zur Erinnerung: Dabei wertet Google die Daten aus dem Bericht zur Benutzererfahrung über den Chrome-Browser aus. Diese enthalten tatsächliche Nutzungsdaten einer Website von echten Nutzern. Es ist also nicht zielführend, auf den allgemeinen PageSpeed-Score oder gar ein Google-fremdes PageSpeed-Tool zu optimieren.

Wo besteht Handlungsbedarf?

Wenn man nun eine Webseite auf mögliche Probleme mit den Web Vitals hin untersuchen möchte, benötigt man Zugriff auf deren Daten in der Search Console von Google. Zwar zeigt der PageSpeed-Insights-Bericht an, ob der Core-Web-Vitals-Test bestanden wurde, jedoch erhält man hier lediglich

eine zusammengefasste Auswertung der Daten aus den letzten 28 Tagen von allen Seiten, die von der betreffenden Domain bereitgestellt werden. Man kann zwar auch direkt auf die Rohdaten des Chrome User Experience Reports zugreifen, die in einem öffentlichen BigQuery-Projekt bei Google zur Verfügung stehen, allerdings ist die Abfrage darüber nicht gerade einfach und außerdem veröffentlicht Google die Felddaten des Vormonats immer gegen Monatsmitte des nächsten Monats. Das bedeutet zum Beispiel, dass am 15.10. die Daten aus dem September veröffentlicht werden. Außerdem sind in dieser Datenbank ebenfalls lediglich zusammengefasste Daten zu derzeit 850.885 Domains enthalten.

Spezifische Auswertungen für einzelne URLs sind aktuell ausschließlich über den Core-Web-Vitals-Bericht in der Google Search Console (<http://einfach.st/gscvitals>) zu extrahieren.

Dort lassen sich, mit nur einem Tag Verzögerung, die Werte aller relevanten URLs samt Empfehlungen seitens Google abrufen und die Entwicklung der letzten 90 Tage einsehen (Abbildung 2).

Wie Mario Fischer in seinem Artikel in Ausgabe #63 bereits angemerkt hat, liegen hier zwar nicht für sämtliche URLs einzeln entsprechende Werte vor, aber die Praxis der letzten Wochen und Monate hat gezeigt, dass die zusammengefassten Reports für die gezielte Optimierung hinreichend detailliert sind. Denn die Suchmaschine fasst offenbar URLs mit denselben Problemen auf Template-Basis zusammen und

INFO

Das Page-Experience-Update im Überblick

Aufmerksame Beobachter verwundert es nicht, dass der Suchmaschinen Gigant aus Kalifornien seine Bemühungen, die Benutzererfahrung für seine Nutzer zu verbessern, über das Page-Experience-Update auch auf Bereiche außerhalb der eigenen Plattform ausdehnt. In den letzten Jahren wurden bereits vier Kriterien im Sinne einer positiven Benutzererfahrung als Ranking-Faktoren eingeführt. Zu den bekannten Faktoren „Mobile Friendly“, „Safe Browsing“, „HTTPS“ und der Vermeidung störender Overlays und Pop-ups aka „No Intrusive Interstitials“ gesellen sich nun die neu eingeführten Core Web Vitals (vgl. Abbildung 1). Diese Metriken sollen Website-Besitzern helfen, Möglichkeiten zur Verbesserung der Benutzererfahrung zu identifizieren. Die drei Werte LCP, FID und CLS beziehen sich dabei auf die Aspekte Geschwindigkeit, Reaktionsfähigkeit und visuelle Stabilität, wie im Artikel „Das Page-Experience-Update“ in Ausgabe #63 detailliert dargelegt.

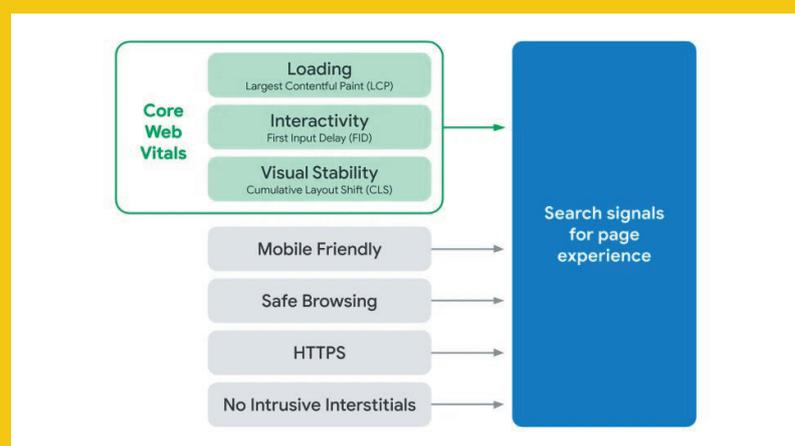


Abb.1: : Übersicht aller Signale des kommenden Page-Experience-Updates
(Quelle: <https://webmasters.googleblog.com/2020/05/evaluating-page-experience.html>)

liefert für eine Gruppe betroffener URLs maximal 20 Beispiel-URLs. So steht in Abbildung 3 der Artikel „Die besten WordPress Themes“ stellvertretend für eine Gruppe von 14 URLs, die aufgrund desselben Fehlers einen CLS-Wert auf Mobilgeräten von 0,19 erhalten und damit die Grenze von 0,1 überschreiten. Mit einem Klick bekommt man eine Liste von bis zu 20 weiteren URLs, die unter demselben Problem zusammengefasst wurden (rechte Spalte in Abbildung 3). Wenn man die betroffenen URLs genauer untersucht, stellt man fest, dass der erhöhte CLS-Wert tatsächlich durch ein templatebasiertes Phänomen verursacht wird und somit zu einem Task zusammengefasst werden kann.

Web-Vitals-Optimierung: Den richtigen Anfang finden

Zuallererst einmal ist es wichtig, den problematischen Wert aus der Search Console nachvollziehen zu können. Je nach betroffenem Wert sind unterschiedliche Herangehensweisen ratsam. Für den LCP- und den CLS-Wert sollte man mit einem Lighthouse-Test im eigenen Browser beginnen. Dieses Tool ist in den Entwicklertools des Google Chrome-Browsers integriert und ermöglicht es, aus dem Test direkt in die tiefere Analyse zu gelangen. Da man während des Ladevorgangs, ähnlich dem PageSpeed-Insights-Test, im Lighthouse-Test nicht mit der Webseite interagieren kann, wird kein FID-Wert erhoben. Empfehlungen zur Analyse und Optimierungsansätze für den FID erhalten Sie im letzten Teil dieses Artikels.

LCP und CLS mit Lighthouse analysieren:

- So wird die Analyse im Google Chrome-Browser gestartet:
1. Die entsprechende URL in einem Inkognito-Fenster öffnen.

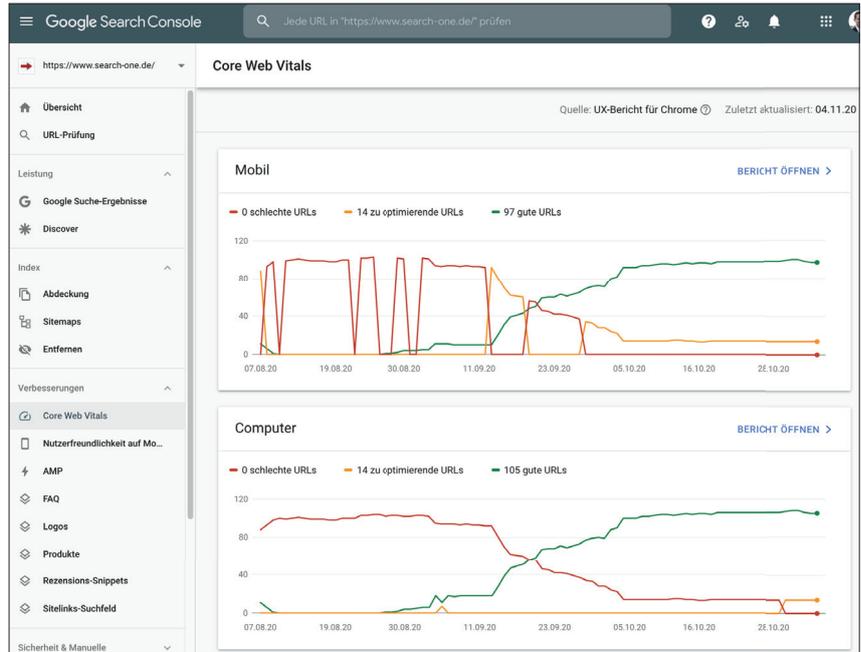


Abb. 2: Core-Web-Vitals-Bericht in Googles Search Console

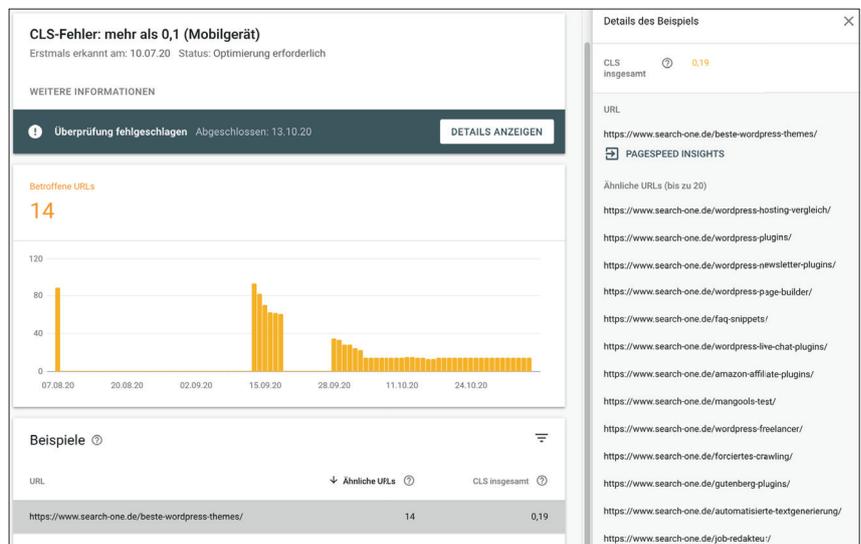


Abb. 3: Eine URL als Beispiel liefert stellvertretend Hinweise für 14 URLs mit demselben Problem

2. Die Entwicklertools öffnen (Umschalt + F12 unter Windows bzw. CMD + Option + I unter macOS) oder per Rechtsklick auf der Seite und im Menü „Untersuchen“ wählen.
 3. Wechsel in den Reiter „Lighthouse“ in den nun sichtbaren Developer-Tools.
 4. Auswahl, ob Desktop oder Mobil getestet werden soll. (Bei den Kategorien reicht es aus, „Performance“ auszuwählen.)
 5. Klick auf „Generate Report“.
- Die Seite wird nun neu geladen und verschiedenen Tests unterzogen.

Der Test dauert je nach Webseite in etwa zwischen 15 und 90 Sekunden. Anschließend lassen sich die erhobenen Werte einsehen und mit den gemeldeten Werten aus der Search Console vergleichen. Wenn die Werte übereinstimmen, kann man sich an die Optimierung machen.

Wichtig bei der Optimierung ist es, stets an Desktop- und Mobile-Ansicht zu denken. Durch die veränderte Darstellung kommen, insbesondere bei responsiven Templates, häufig abweichende Werte zustande.

INFO

Was tun bei abweichenden Werten?

Weichen die Werte in der Search Console deutlich von den gemessenen im Lighthouse-Test ab, sollte man folgende drei Punkte berücksichtigen:

1. Felddaten mit simulierter Drosselung nachvollziehen

Wie bereits erwähnt, werden die Web Vitals von echten Besuchern erhoben, daher kann deren Netzwerkverbindung und System-Performance erhebliche Auswirkungen auf den LCP und insbesondere den FID haben. Um die eigene Webseite unter erschwerten Bedingungen zu testen, lassen sich sowohl weniger zur Verfügung stehende Hardware-Ressourcen als auch eine langsame Internetgeschwindigkeit simulieren. Dazu kann im Einstellungs-Menü des Performance-Tabs das sogenannte CPU-Throttling aktiviert und die Geschwindigkeit der Netzwerk-Übertragung reduziert werden.

2. Weitere Viewport-Varianten testen

Die Auflösung der echten Besucher kann von den simulierten Viewports des Lighthouse-Tools abweichen. Daher kann es bei ungewöhnlichen Größen- oder Seitenverhältnissen zu einem anderen Verhalten responsiver Layouts kommen. Dies kann sowohl zu einer Verschiebung von Elementen führen und damit den CLS-Wert beeinflussen als auch den LCP verändern. Der LCP hängt vom größten ohne Scrollen sichtbaren Element ab und dieses kann durch eine geänderte Auflösung ebenfalls variieren. Um dies zu testen, sollte man den Ladevorgang in diversen Auflösungen über das Performance-Tab in den Chrome-Entwicklertools analysieren, bis man das betroffene Element identifizieren konnte.

3. Chrome-Erweiterungen und Ad-Blocker deaktivieren

Sämtliche Erweiterungen, die in die Seiteninhalte eingreifen, wie Ad-Blocker oder Script-Blocker, müssen unbedingt deaktiviert werden, um eine möglichst neutrale Testumgebung zu verwenden. Ebenso sollten Sie sich aus dem eigenen Backend ausloggen oder noch besser alle Domain-Cookies löschen oder ein Inkognito-Fenster verwenden, bevor Sie den Lighthouse-Test durchführen. Als eingeloggter Nutzer könnten Caching-Ausnahmen greifen oder zusätzliche Elemente wie Admin-Leisten oder Bearbeitungswerkzeuge geladen werden, die ein normaler Besucher nie zu Gesicht bekäme. *Tipp: In den Chrome-Einstellungen kann man für jede Erweiterung separat festlegen, ob diese in einem Inkognito-Fenster aktiviert werden soll oder nicht.*

Wie ist das Vorgehen bei einem schlechten CLS-Wert?

Haben die Analysen ergeben, dass der CLS-Wert zu hoch ist, müssen die Elemente der Webseite identifiziert werden, die für die Verschiebung während des Ladevorgangs verantwortlich sind.

Zur Erinnerung: Der CLS ist ein prozentualer Wert, der sich aus der Multiplikation zweier Prozentangaben ergibt: Der Anteil des sich verändernden Bereichs multipliziert mit dem Anteil des sichtbaren Bereichs, um den sich das betroffene Element während des Ladens verschiebt.

Wie in Abbildung 4 ersichtlich ist, findet man unter den Metriken auch eine Bilderreihe mit Miniatur-Screenshots der Webseite während des Ladens und einem Button mit der Beschriftung „View Original Trace“. Die Miniatur-Screenshots sind manchmal bereits ausreichend, um die verantwortlichen Elemente der Layout-Verschiebung zu identifizieren. Falls dies nicht der Fall sein sollte, gelangt man mit einem Klick auf den Button „View Original Trace“ zum Reiter „Performance“.

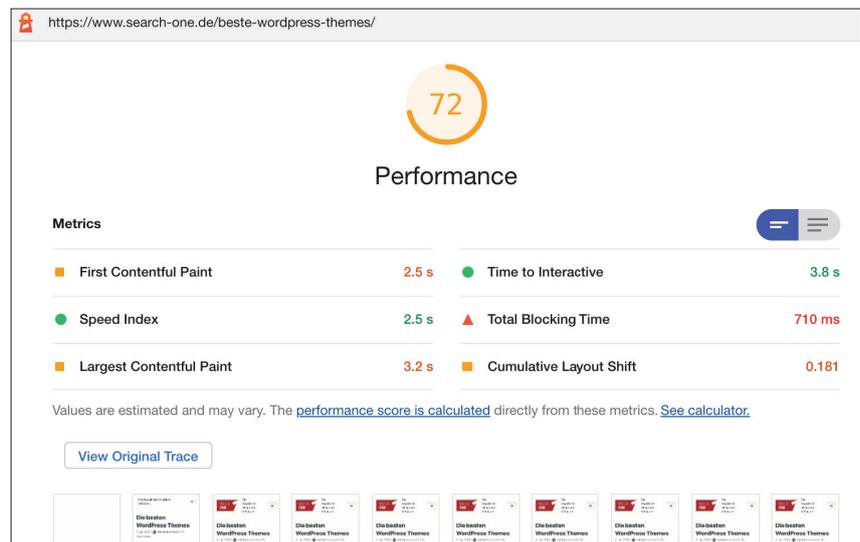


Abb. 4: Ergebnis des Lighthouse-Tests für <https://www.search-one.de/beste-wordpress-themes/>

Dieser Tab (Abbildung 5) kann auf den ersten Blick ein wenig einschüchternd wirken, denn man wird nahezu von Informationen erschlagen. Für die konkrete Optimierung findet man hier jedoch alle relevanten Hinweise und Analysen. Während der Lighthouse-Test nur darüber Auskunft gibt, ob der CLS gut oder schlecht ist, verstecken sich hinter dem Performance-Tab die Hinweise, welche Elemente für den schlechten CLS verantwortlich sind.

In der Zeile „Timings“ findet man neben dem bekannten LCP außerdem die Events FP (First Paint), FCP (First Contentful Paint), DCL (Dom Content Loaded) und mit L das finale Onload Event gekennzeichnet. In der Zeile „Experience“ darunter findet man sämtliche Layout Shifts, allerdings erst, wenn man die Aufzeichnung erneut startet und die Seite neu lädt. Dazu kann wie folgt vorgegangen werden:

1. Die angezeigte Aufzeichnung muss zunächst mit einem Klick auf das „Clear“-Symbol gelöscht werden.
2. Anschließend kann man einen Ladevorgang neu aufzeichnen, indem man das „Reload“-Symbol klickt.

Nun kann in der Zeile „Experience“ mit einem Klick auf den rot eingefärbten „Layout Shift“ der Übeltäter und seine Verschiebung angezeigt werden. In Abbildung 6 ist dies das Element `div.nv-title-tagline-wrap`.

Zusätzlich kann man mit einem Mouse-Over auf dem Zeitstrahl ganz oben im „Performance“-Tab Screenshots jedes einzelnen Rendering-Frames einsehen und dabei die Verschiebung im Layout visuell nachvollziehen. In den Abbildungen 7 und 8 erkennt man die Verschiebung der Tagline „Das Magazin für erfolgreiche Webseiten“ durch das geladene Logo, das vorher keinen Platz für sich reserviert hatte.

Häufiger CLS-Fehler 1: Wenn Bilder erst nach dem Laden die korrekte Größe annehmen

Häufig wird für das Logo, ein Beitragsbild oder ein anderes Visual kein Platz reserviert, da das Element weder im HTML noch über das Stylesheet eine definierte Breite und Höhe zugewiesen bekommt. Liegt diese Information nach dem Laden der Bild-Datei dem Browser vor, verschiebt sich das Layout, da das Bild nun seinen Platz beansprucht.

Die Lösung für dieses häufige Problem liegt in der Definition der korrekten und endgültigen Größe und Position. Dies kann am einfachsten mittels CSS umgesetzt werden. Wird das Beitragsbild von einem Container umschlossen, kann auch eine definierte Größe des Elternelements beim Aufruf bereits den Platz für das Bild reservieren. Das Bild wird dann wenige Sekunden später in dem reservierten Platz angezeigt und das Layout verschiebt sich nicht mehr.

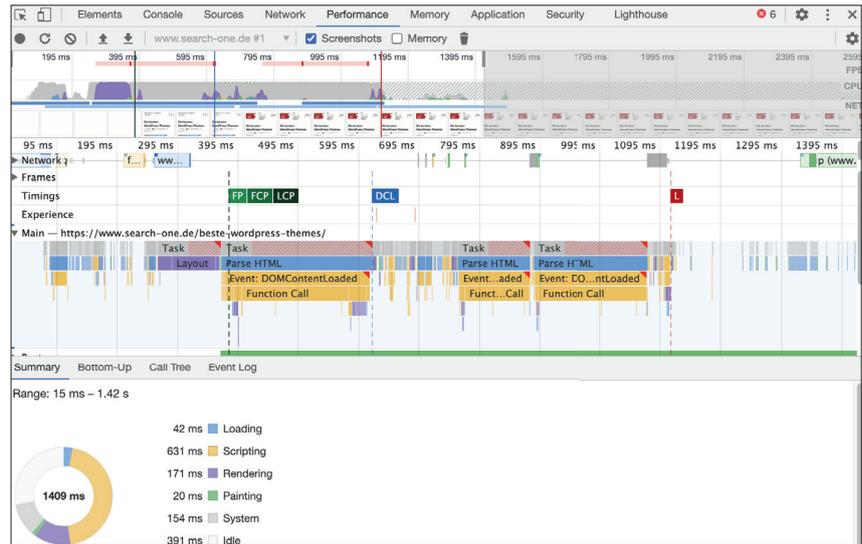


Abb. 5: Performance-Analyse in den Google Chrome Developer-Tools für <https://www.search-one.de/beste-wordpress-themes/>

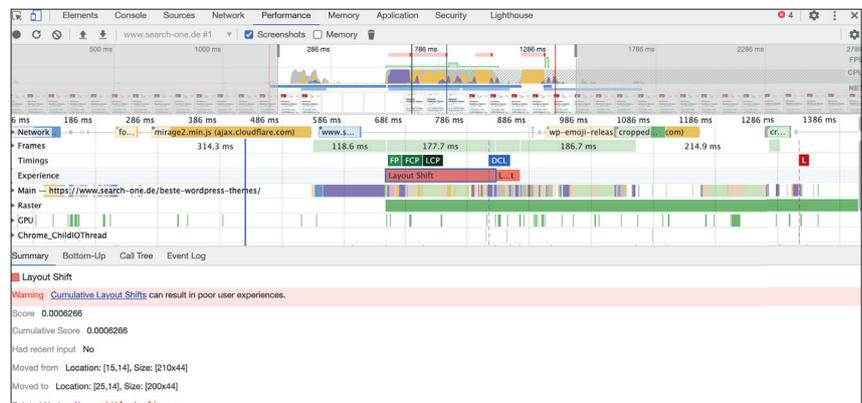


Abb. 6: Detail-Informationen zum Layout Shift beim Laden der Seite

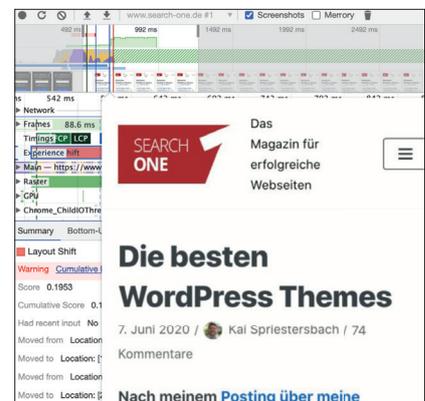
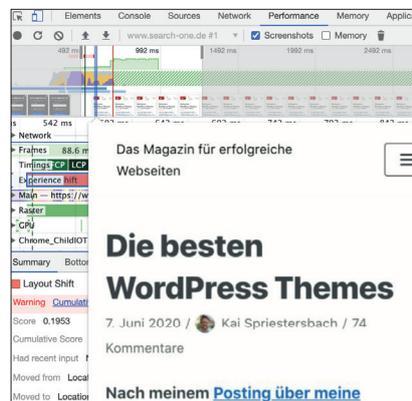


Abb. 7 und 8: Screenshot der Seite noch ohne und mit geladenem Logo

Häufiger CLS-Fehler 2: Child-Themes überschreiben Styles des Parent-Themes

Wurde eine Webseite mit einem Content-Management wie zum Beispiel WordPress erstellt, kann es passieren, dass zunächst die Styles des Haupt-Themes geladen werden und diese

anschließend mit den Modifikationen des Entwicklers in einem sogenannten Child-Theme überschrieben werden. Durch den integrierten Customizer, visuelle Page-Builder oder andere Arten von Custom CSS kann der Nutzer wiederum die Styles der Seite überschreiben, was mehrfache Änderung beim

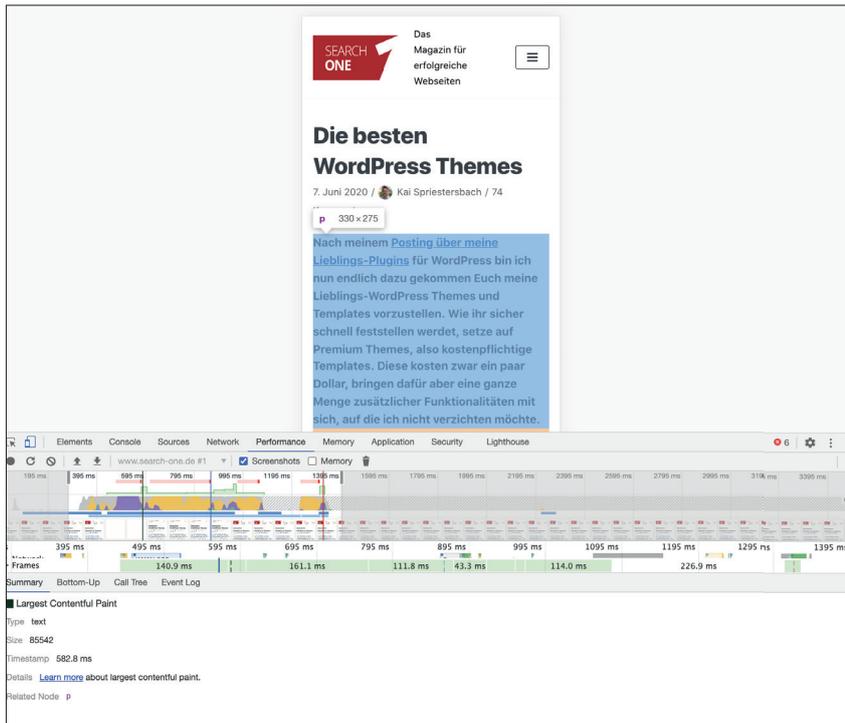


Abb. 5: Performance-Analyse in den Google Chrome Developer-Tools für <https://www.search-one.de/beste-wordpress-themes/>

Laden und entsprechende Verschiebungen im Layout zur Folge haben könnte.

Tipp: Critical CSS in den Kopf der Seite einfügen

Idealerweise werden die Style-Regeln für sämtliche im Bereich Above-The-Fold sichtbaren Elemente im Kopfbereich der Seite (<head>) definiert und liegen somit beim Rendern des Layouts bereits vor und müssen nicht erst über eine zusätzliche CSS-Ressource geladen werden. Tools wie <https://criticalcss.com/> helfen dabei, das notwendige kritische CSS automatisch zu generieren.

Wie ist das Vorgehen bei einem schlechten LCP-Wert?

Sollte der LCP-Wert über der magischen Grenze von 2,5 Sekunden liegen, kann man das entsprechende Element mit einem Klick auf die LCP-Box in der Zeile „Timings“ anklicken und erfährt unter „Related Node“, welches HTML-Element offenbar das größte zu ladende ist. Im Beispiel in Abbildung 9 ist dies der größte Paragraph und stellt mit einer Renderzeit von nur 582,8 ms kein Problem dar.

Falls es ein Problem mit dem „Largest Contentful Paint“ gibt, obwohl das HTML schnell genug heruntergeladen wird, liegt dies in der Regel entweder an einer zu großen Bilddatei, deren Auflösung reduziert und deren Kompression verbessert werden sollte, oder eine Schriftart wurde ohne die CSS-Eigenschaft `font-display:swap;` eingebunden. Ist dies der Fall, wartet der Browser mit der Anzeige des betreffenden Textes, bis die Schriftart vollständig heruntergeladen wurde. Natürlich kann es auch generelle Probleme mit der Antwortzeit des Servers oder der Ladezeit des HTML geben, jedoch ist deren Verbesserung meist nur mit serverseitigen Optimierungen wie Caching oder Ähnlichem möglich.

Wie ist das Vorgehen bei einem schlechten FID-Wert?

Den „First Input Delay“ kann man leider in keinem Analyse-Tool selbst messen. Denn der FID misst die Zeit von der ersten Interaktion eines Benutzers mit der Seite (d. h., wenn er auf einen Link klickt, auf eine Schaltfläche tippt oder ein benutzerdefiniertes, Java-

Script-gesteuertes Steuerelement verwendet) bis zu dem Zeitpunkt, an dem der Browser tatsächlich in der Lage ist, als Reaktion auf diese Interaktion mit der Verarbeitung von Event-Handlern zu beginnen.

Der FID-Wert erfordert daher einen realen Benutzer und kann nicht in Tools gemessen werden. Die TBT-Metrik (Total Blocking Time) ist jedoch über die Web-Developer-Tools messbar und korreliert laut Google gut mit dem FID im Feld und erfasst Probleme, die die Interaktivität negativ beeinflussen. Optimierungen, die die TBT verbessern, sollten daher auch den FID-Wert echter Benutzer verbessern.

Der FID misst lediglich die „Verzögerung“ bei der Ereignisverarbeitung. Er misst weder die Ereignisverarbeitungszeit selbst noch die Zeit, die der Browser benötigt, um die Benutzeroberfläche nach der Ausführung des Event-Handlers zu aktualisieren.

Für das Blockieren der Verarbeitung ist in der Regel die Ausführung komplexer JavaScript-Codes verantwortlich. Somit lässt sich der FID häufig nur durch Reduktion verwendeter Skripte verbessern. Sämtliche Skripte, die nicht für den visuellen Aufbau der Seite benötigt werden, sollten mittels des Attributs „defer“ ausgezeichnet und damit erst gestartet werden, wenn die Seite bereits geladen und geparkt wurde. Dies ermöglicht es beispielsweise, Analytics-Tags, aber auch Ad-Server-Integrationen und andere Skripte so auszuführen, dass diese keinen negativen Einfluss auf die Interaktivität der Seite für den Nutzer haben. ¶