

# Google Core Web Vitals





**Google Webmasters** ✓

@googlewmc



We're announcing that page experience ranking signals for Google Search will launch in May 2021. This will combine Core Web Vitals and previous UX-related signals.

Learn more:

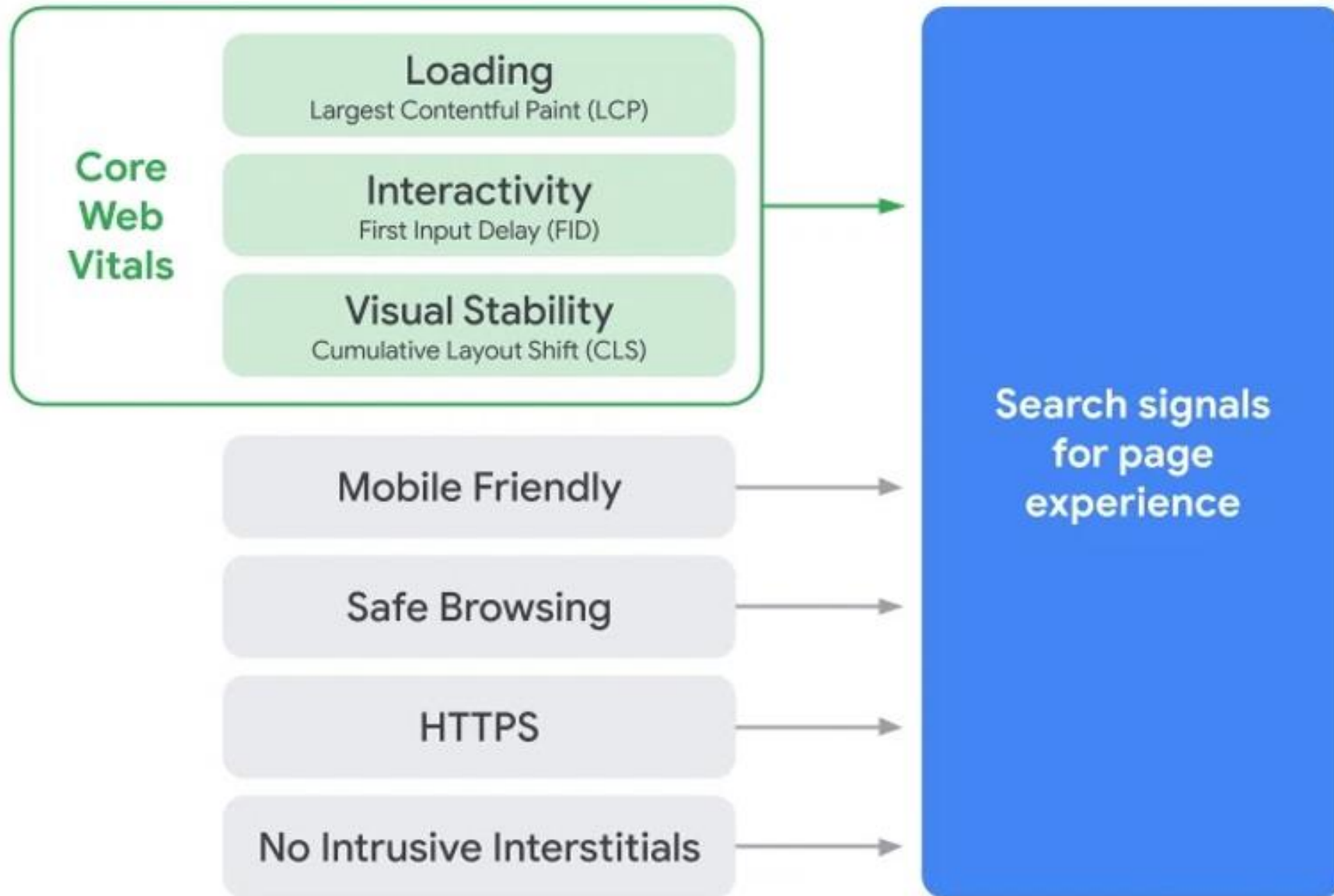
Timing for bringing page experience to Google Search  
Official news on crawling and indexing sites for the Google index  
[webmasters.googleblog.com](https://webmasters.googleblog.com)

5:00 PM · Nov 10, 2020



♥ 493

💬 280 people are Tweeting about this





*(Loading)*

# LCP

**Largest Contentful Paint**



*(Interactivity)*

# FID

**First Input Delay**



*(Visual Stability)*

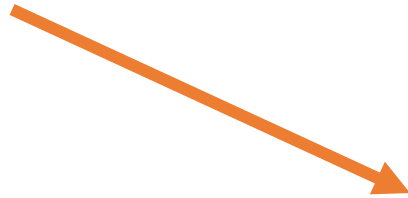
# CLS

**Cumulative Layout Shift**



# Pagespeed Insights (PSI)

Chrome-Daten echter Nutzer



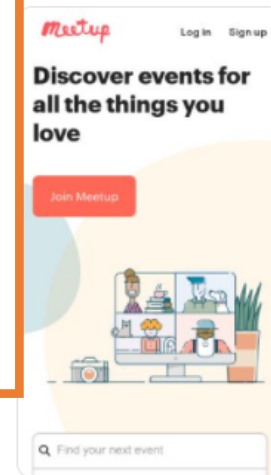
https://www.meetup.com/



**Felddaten** — Nach Auswertung der Felddaten aus den letzten 28 Tagen hat diese Seite den **Core Web Vitals-Test nicht bestanden.**

■ First Contentful Paint (FCP) 2,6 s	● First Input Delay (FID) 32 ms
32% 50% 18%	89% 7% 3%
■ Largest Contentful Paint (LCP) 2,8 s	● Cumulative Layout Shift (CLS) 0,06
68% 22% 10%	79% 10% 12%

Übersicht über Ursprung einblenden



Nicht alle (folgenden) Optimierungen spiegeln sich direkt in PSI wieder (bspw. da hier nur der Aufruf einer einzelnen Seite gewertet wird).

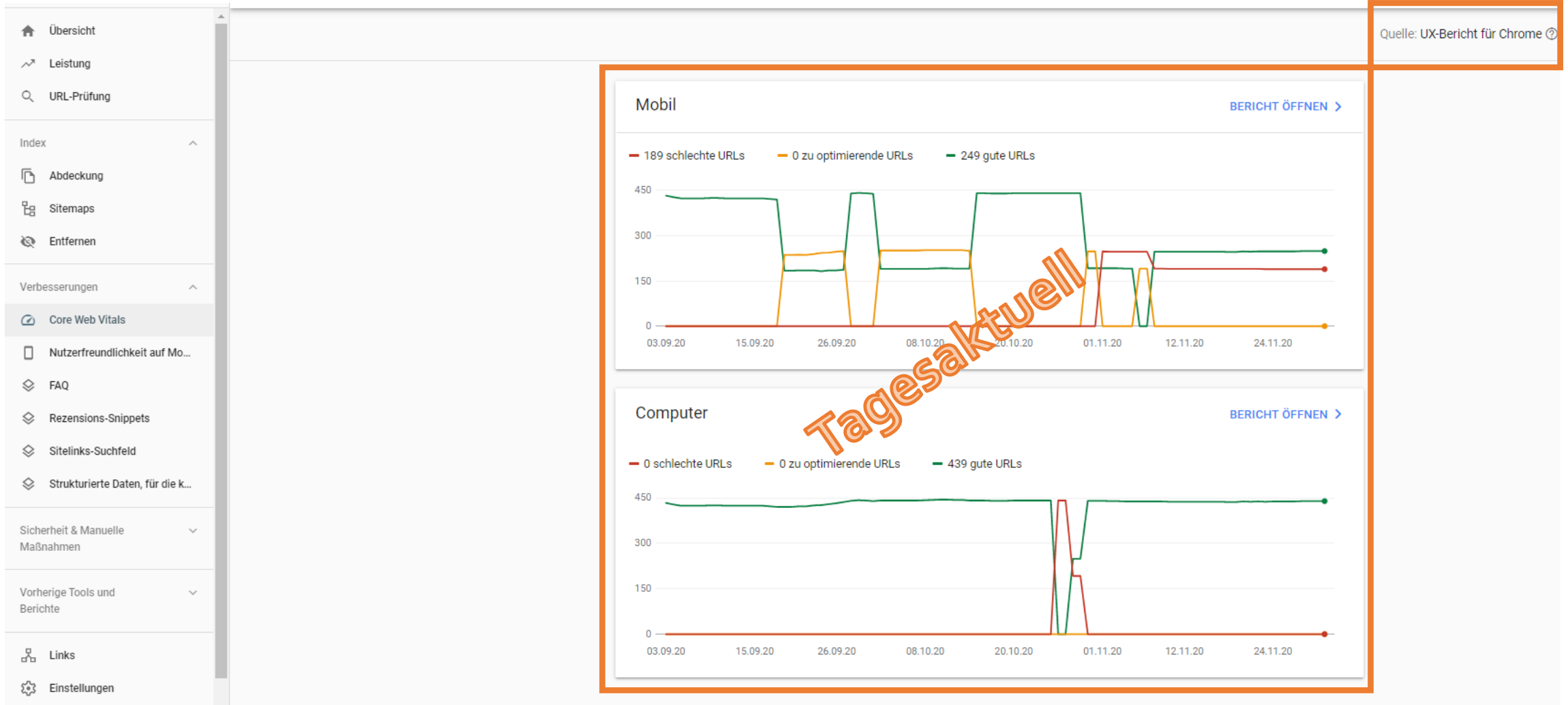
Jedoch wirken sich alle Pagespeed-Optimierungen auf die Field-Daten aus.

## Labdaten

■ First Contentful Paint 2,5 s	▲ Time to Interactive 18,0 s
■ Speed Index 5,1 s	▲ Total Blocking Time 7.570 ms
▲ Largest Contentful Paint 5,5 s	● Cumulative Layout Shift 0,001

Die Werte sind geschätzt und können variieren. Die **Leistungsbewertung** wird direkt aus diesen Messwerten berechnet. [See calculator.](#)

# Google Search Console (GSC)





*(Loading)*

# LCP

**Largest Contentful Paint**



*(Interactivity)*

# FID

**First Input Delay**



*(Visual Stability)*

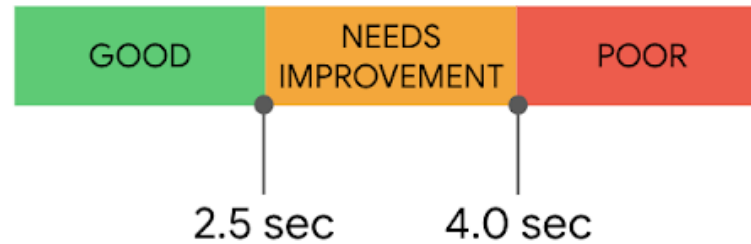
# CLS

**Cumulative Layout Shift**



# LCP

Largest Contentful Paint



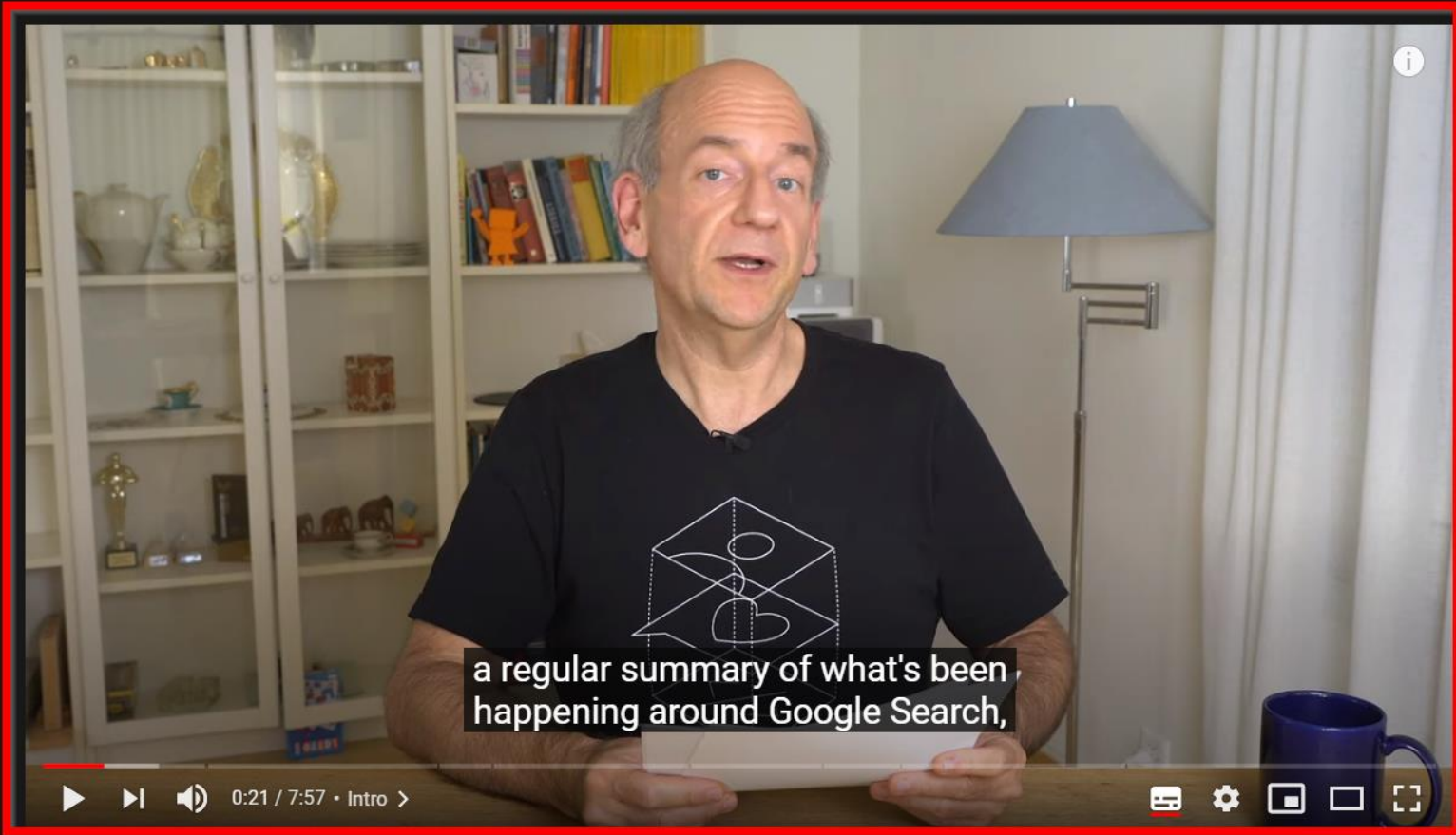
*Die LCP-Metrik (Largest Contentful Paint) beschreibt den Zeitpunkt, an welchem das größte Bild oder der größte Textblock ohne zu scrollen sichtbar ist.*

## Beispiele

<https://tutorial.seo-atelier.com/beispiele/lcp/>

<https://tutorial.seo-atelier.com/beispiele/lcp-text/>



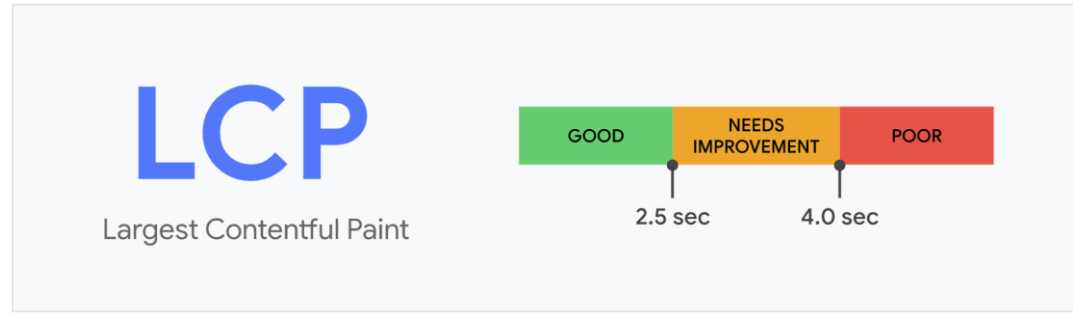


Nächste Titel

AUTOPLAY

- English Google SEO office-hours from November 27, 2020**  
Google Search Central  
3475 Aufrufe · vor 6 Tagen  
Neu
- English Google SEO office-hours from October 27, 2020**  
Google Search Central  
1742 Aufrufe · vor 1 Monat
- Japanese Webmaster Office Hours (ウェブマスター オフ...**  
Google Search Central  
2204 Aufrufe · vor 4 Wochen
- JavaScript SEO office hours November 25th, 2020**  
Google Search Central  
1139 Aufrufe · vor 1 Woche
- Site Speed: What SEOs Need to Know #AskGoogleWebmasters**  
Google Search Central  
31.690 Aufrufe · vor 1 Jahr
- The Complete Guide to SEO in 2020 (Full webinar)**  
Income School  
164.613 Aufrufe · vor 9 Monaten

Google Search News (November '20) - from Google Webmasters to Google Search Central and more  
7.872 Aufrufe · 25.11.2020  
411 14 TEILEN SPEICHERN ...



## Welche Elemente?

Text

`<img>`

`<image>` innerhalb von `<svg>`

Background-Images (via `url()`, kein Gradient, o.Ä.)

Block-Elemente mit Text

(Stand 12.2020)

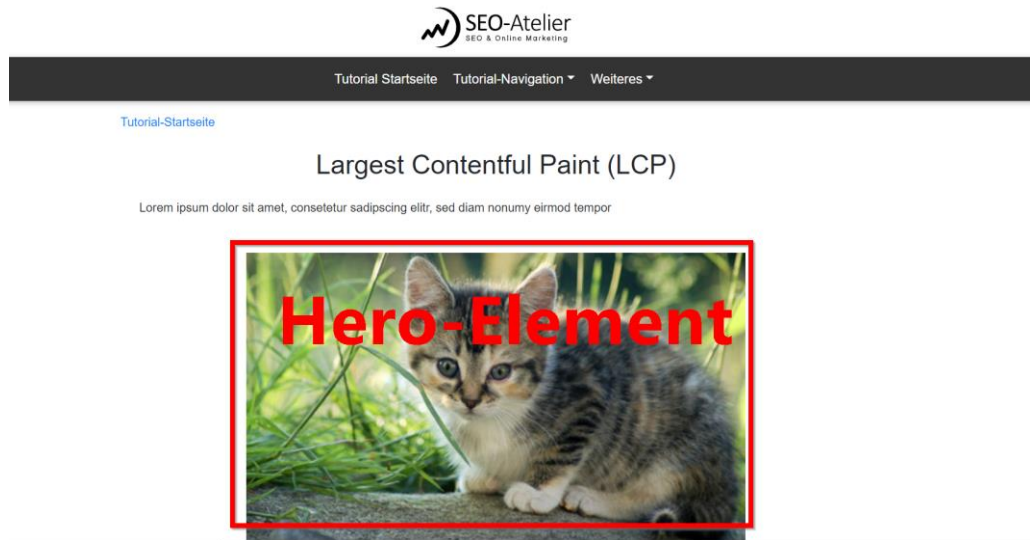


## Besonderheiten!

Bei `<video>` wird lediglich das poster-Image gewertet (!)

Weitere Elemente können folgen

# PSI & Lighthouse - Hero-Element ermitteln zur Optimierung der LCP



### ● Largest Contentful Paint-Element — 1 element found

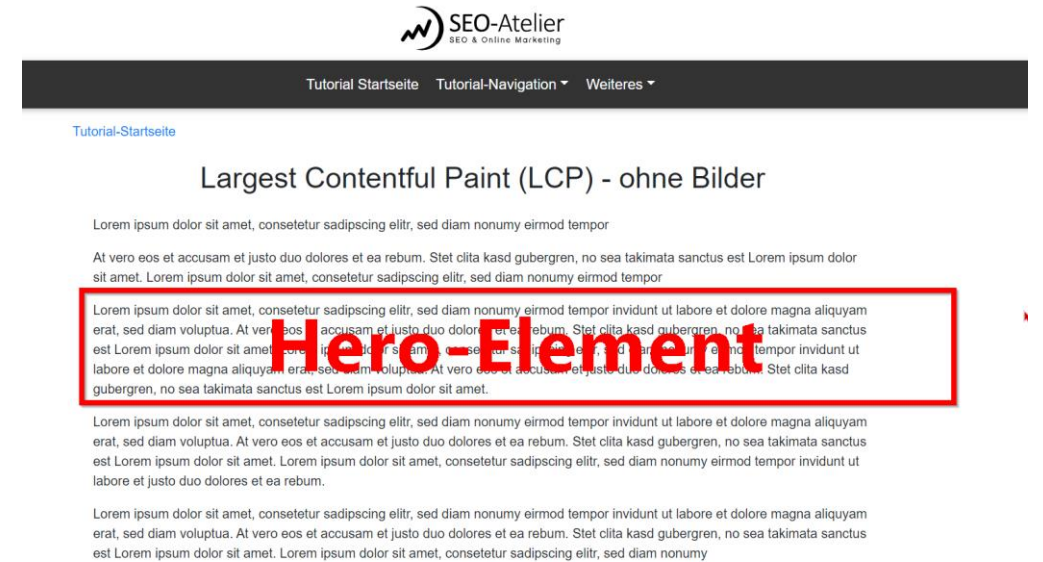


Dies ist das größte Inhaltselement, das im Darstellungsbereich angezeigt wird. [Weitere Informationen](#)

Element

```
img  

```



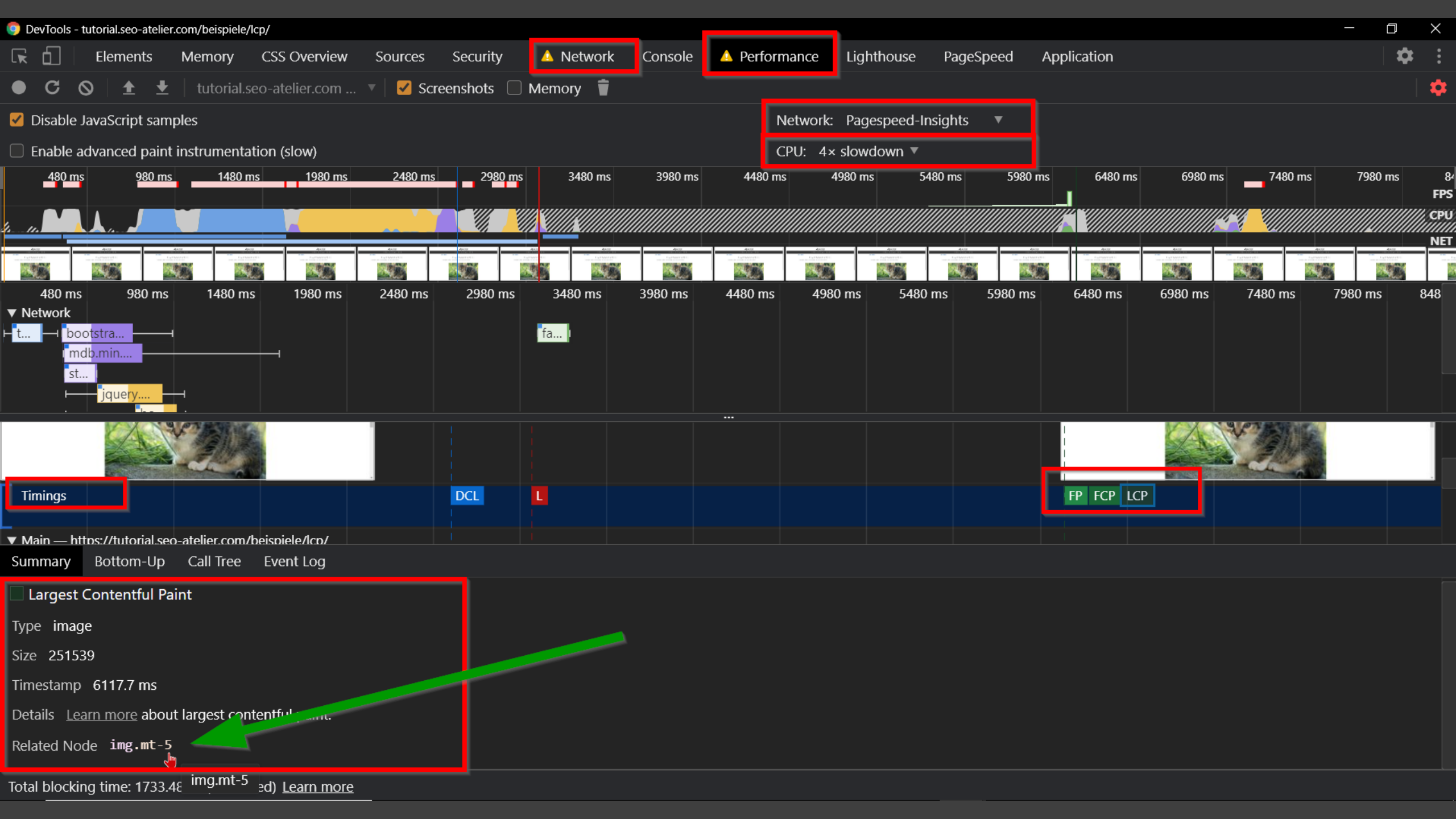
### ● Largest Contentful Paint-Element — 1 element found



Dies ist das größte Inhaltselement, das im Darstellungsbereich angezeigt wird. [Weitere Informationen](#)

Element

```
At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd guberg...  
<div class="row" id="block-3">
```



Network

Performance

Network: Pagespeed-Insights

CPU: 4x slowdown

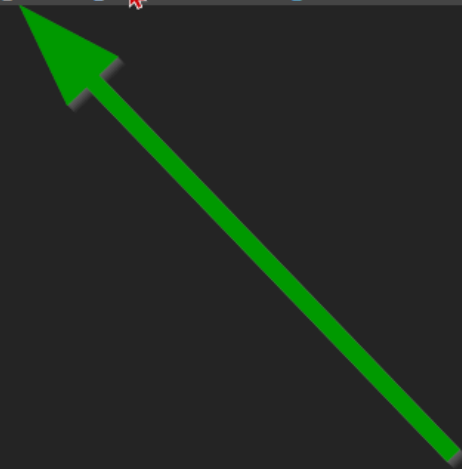
Timings

**Largest Contentful Paint**  
Type image  
Size 251539  
Timestamp 6117.7 ms  
Details [Learn more about largest contentful paint](#)  
Related Node **img.mt-5**

Total blocking time: 1733.48 ms (img.mt-5) [Learn more](#)

```
!DOCTYPE html>
html lang="de" dir="ltr">
<head>...</head>
<body>
  <div id="wrapper">
    <header>...</header>
    <div class="container-fluid">...</div>
    <main>
      <div class="container my-4">...</div>
      <div class="container textblock">...</div>
      <div class="container">
        <div class="row justify-content-center">
           == $0
        </div>
      </div>
      <div class="container textblock mt-5">...</div>
      <div class="container textblock mt-5">...</div>
    </main>
  </div>
</body>
/html>
```

Styles	Computed
Filter	
element.style {	
}	
.mt-5, .my-5 {	margin-top: 3rem
}	
img {	vertical-align: top
	border-style: solid
}	
*, ::after,	
::before {	box-sizing: border-box
}	
img[Attributes Styl	height: 400px;
}	
Inherited from body	
body {	font-family: "Roboto", sans-serif
	font-weight: 300
}	
body {	margin: 0;
	font-family: -apple-system, BlinkMacSystemFont, "Helvetica Neue", Arial, sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol";

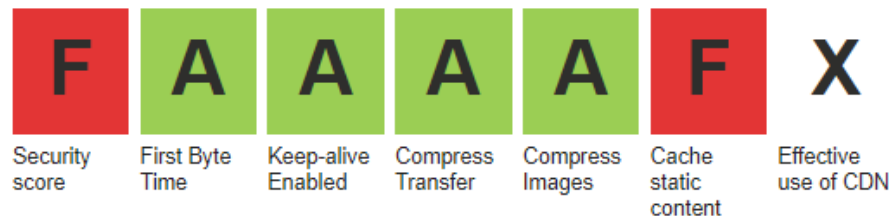




# Web Page Performance Test for

<https://tutorial.seo-atelier.com/beispiele/lcp/>

From: Frankfurt, Germany - EC2 - Chrome - 3GFast  
30.11.2020, 22:14:04

[Summary](#)[Details](#)[Performance Review](#)[Content Breakdown](#)[Domains](#)[Processing Breakdown](#)[Screenshot](#)[Image Analysis](#)[Request Map](#)

Tester: i-037a812833af9ccc5

Test runs: 3

[Re-run the test](#)

[View JSON result](#)

[Raw page data](#) - [Raw object data](#)

[Export HTTP Archive \(.har\)](#)

[View Test Log](#)

## Performance Results (Median Run - SpeedIndex)

					Web Vitals			Document Complete			Fully Loaded			
	First Byte	Start Render	First Contentful Paint	Speed Index	Largest Contentful Paint	Cumulative Layout Shift	Total Blocking Time	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View ( <a href="#">Run 1</a> )	0.729s	1.700s	1.698s	2.443s	2.831s	0	≥ 0.000s	2.797s	9	160 KB	2.971s	10	160 KB	\$----
Repeat View ( <a href="#">Run 2</a> )	0.708s	0.900s	0.882s	1.302s	1.499s	0	≥ 0.000s	1.479s	2	1 KB	1.479s	2	1 KB	

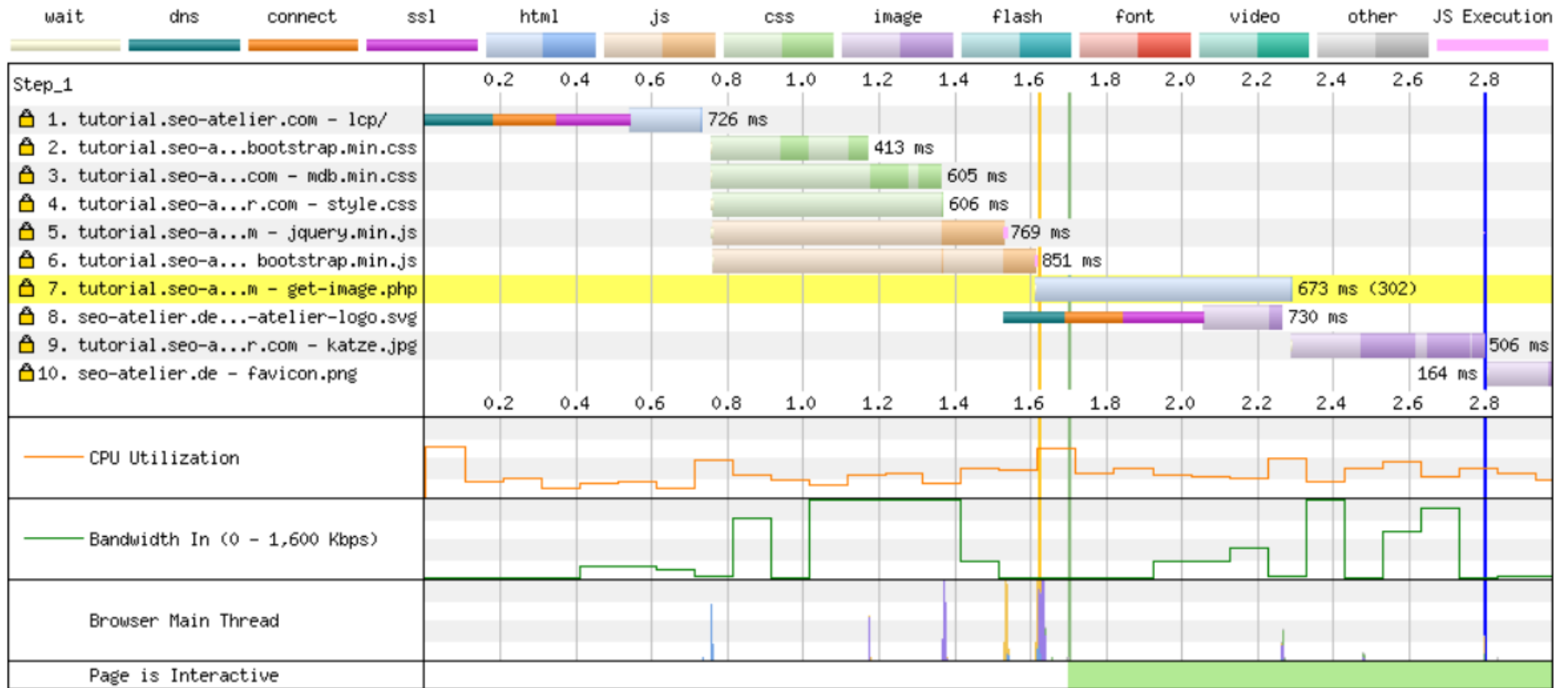
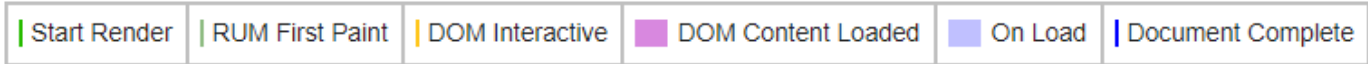
[Plot Full Results](#)

## Test Results

Run 1:

	Waterfall	Screenshot	Video

# Waterfall View





*(Loading)*

# LCP

**Largest Contentful Paint**



*(Interactivity)*

# FID

**First Input Delay**



*(Visual Stability)*

# CLS

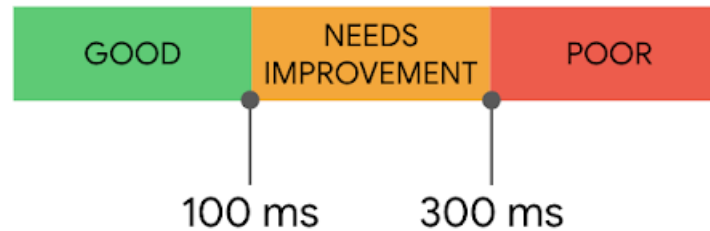
**Cumulative Layout Shift**





# FID

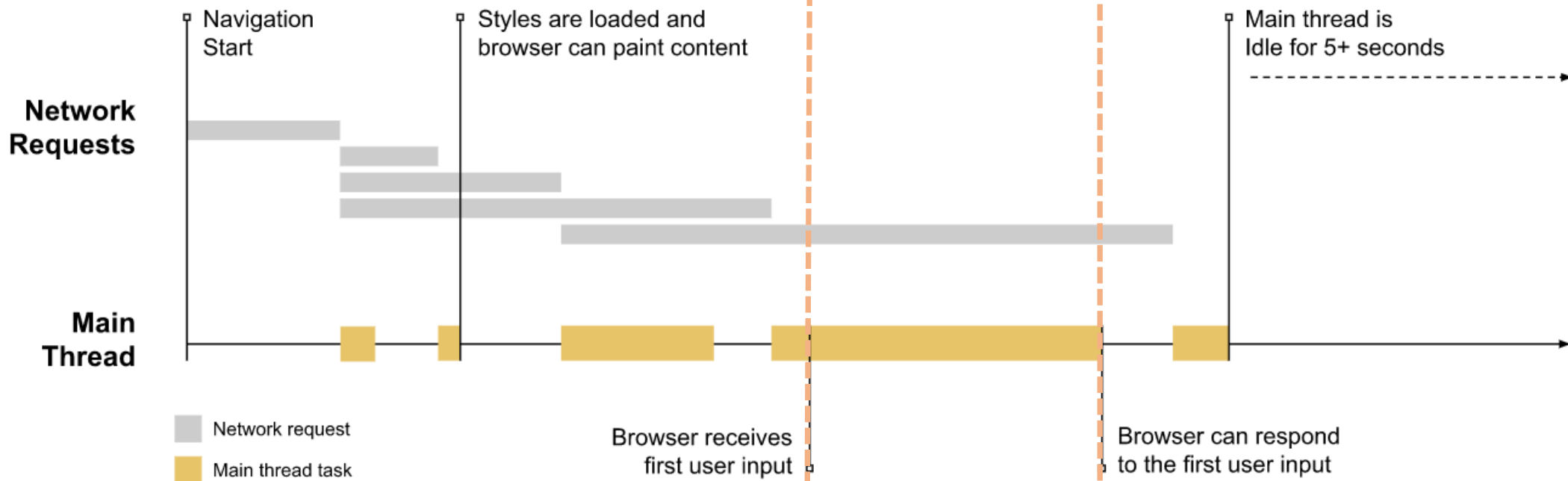
First Input Delay



*FID beschreibt die Zeit von der ersten Interaktion eines Benutzers mit einer Seite bis zu dem Zeitpunkt, an welchem der Browser auf diese Interaktion reagieren kann.*

TTI

FCP



Seite (teilweise) sichtbar

FID



Klick Klick Klick Klick

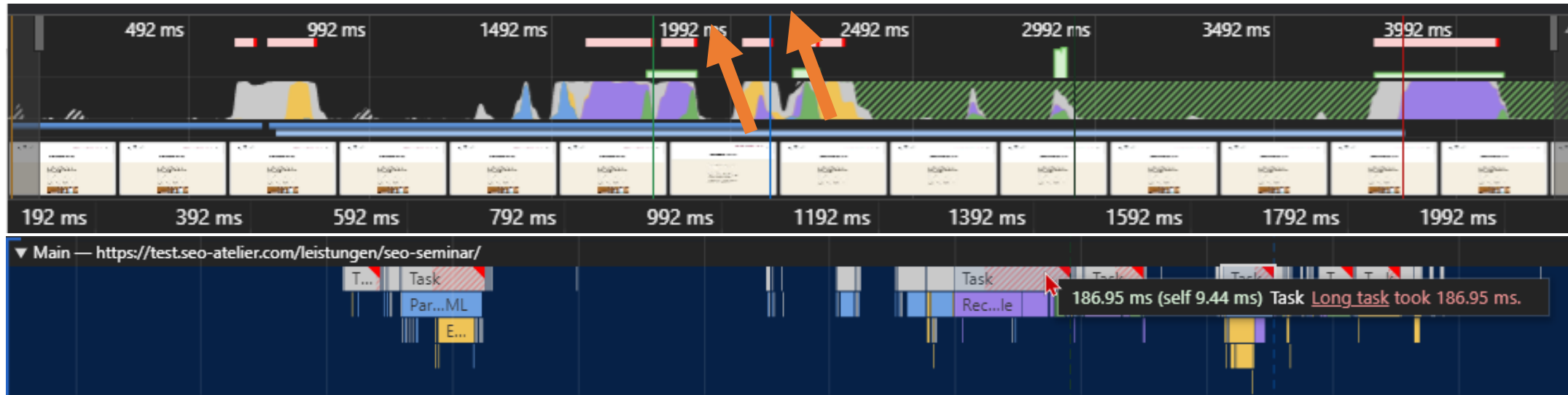
Jetzt reagiert die Seite

# First Input Delay – der Main-Thread als Flaschenhals

Bei einer Optimierung sollte das Ziel sein, „Long Tasks“ zu vermeiden.

- JavaScript ist eine Single-Threaded-Umgebung, es können also nicht gleichzeitig mehrere Skripts ausgeführt werden.
- Long Task = Jeder Code, welcher den Main-Thread für mehr als 50 ms (oder länger) blockiert
- Mögliche Optimierung: Aufteilen in mehrere asynchrone geladen Scripts.

Chrome Dev-Tools, Performance-Tab:



# WHEN JAVASCRIPT BLOCKS THE MAIN THREAD, NOTHING ELSE CAN HAPPEN ON IT

Block the main thread

Element	Example
<code>&lt;img&gt;</code>	
<code>&lt;a&gt;</code>	<a href="https://example.com">https://example.com</a>
<code>&lt;input&gt;</code>	<input type="button" value="Check me"/>

<https://philipwalton.com/articles/why-web-developers-need-to-care-about-interactivity/>



# JS ist eine sehr teure Ressource – weniger ist mehr

JavaScript bytes     $\neq$     JPEG bytes  
~170KB    Network Transmission    ~170KB

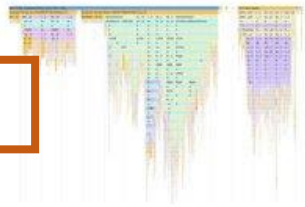


## Resource Processing

Flame Chart				Bottom-Up				Call Tree				Event Log			
Main				Filter				Group by URL							
Self Time	Total Time	Activity		Self Time	Total Time	Activity		Self Time	Total Time	Activity		Self Time	Total Time	Activity	
1997.0 ms	44.6 %	1997.0 ms	44.6 %	1997.0 ms	44.6 %	native V8Runtime									
608.9 ms	13.6 %	937.6 ms	20.9 %	608.9 ms	13.6 %	Compile									
303.6 ms	6.8 %	310.3 ms	6.9 %	303.6 ms	6.8 %	Parse									

~2s in Parse/Compile

~1.5s in Execution



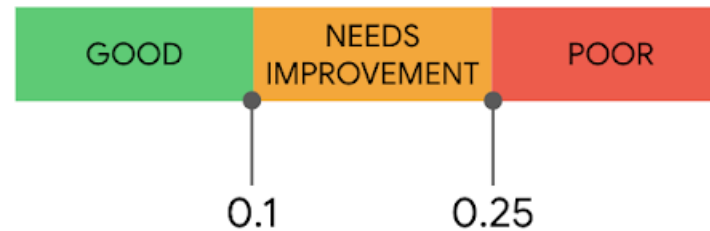
0.064s in Image Decode

0.028s in Rasterize Paint

@addyosmani - 170KB of (compressed) JS vs. JPEG bytes over a slow 3G network on a Moto G4. JS needing parsed is even larger once decompressed.

# CLS

Cumulative Layout Shift



*Messwert für die visuelle Stabilität der Seite.*

*Der CLS stellt die Summe aller unerwarteten Layoutverschiebung dar, welche während der gesamten Lebensdauer der Seite auftritt.*

Berechnung erfolgt über Größe und Maß an Verschiebung in Verhältnis zum Screen:

<https://web.dev/cls/>

[Klick](#)

# Cumulative Layout Shift

- Bilder ohne Höhenangabe
  - Bei Bildern sollten immer Höhe und Breite angegeben werden
  - Ggf. mit CSS [aspect ratio boxes](#) arbeiten
- Nachgeladene Seitenbereiche
  - Besonders Werbeeinbindungen sind hiervon gerne betroffen
- Iframes
  - [FitWids](#) „Fluid Width Video embeds“
    - “A lightweight, easy-to-use jQuery plugin for fluid width video embeds.”
- Unkritisch eingebundene Schriften?

```
img {  
  aspect-ratio: attr(width) / attr(height);  
}
```

# Cumulative Layout Shifts finden

- Pagespeed Insights
- Lighthouse
- Chrome Dev-Tools Performance-Tab
- Chrome Dev-Tools
- Webpagetest.org
- Ein ganz besonderes Tool, welches wir euch später vorstellen möchten!

**Diagnose** — Weitere Informationen zur Leistung Ihrer App finden Sie hier. Diese Angaben haben keinen **direkten Einfluss** auf die Leistungsbewertung.

▲ Statische Inhalte mit einer effizienten Cache-Richtlinie bereitstellen — 7 Ressourcen gefunden

● Verkettung kritischer Anfragen vermeiden — 5 Ketten gefunden

● Halten Sie die Anfrageanzahl niedrig und die Übertragungsgröße gering — 9 Anfragen • 163 KiB

● Largest Contentful Paint-Element — 1 element found

● Umfangreiche Layoutverschiebungen vermeiden — 2 elements found

Diese DOM-Elemente tragen am stärksten zur CLS der Seite bei.

Element	CLS-Beitrag
Lorem ipsum dolor sit amet, consetetur "verrutschender" Link sadipscing elit, ... <p>	0,063
Lorem ipsum dolor sit amet, consetetur "verrutschender" Link sadipscing elit, ... <p>	0,055

● Lange Hauptthread-Aufgaben vermeiden — 2 lange Aufgaben gefunden

**Bestandene Prüfungen (25)**

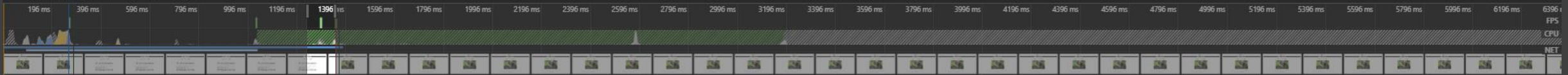


DevTools - tutorial.seo-atelier.com/beispiele/ds/

Elements Memory CSS Overview Sources Security Network Console Performance Lighthouse PageSpeed Application

tutorial.seo-atelier.com ... Screenshots Memory

Disable JavaScript samples Network: Online  
Enable advanced paint instrumentation (slow) CPU: No throttling



ms 196 ms 396 ms 596 ms 796 ms 996 ms 1196 ms 1396 ms 1596 ms 1796 ms 1996 ms 2196 ms 2396 ms 2596 ms 2796 ms 2996 ms 3196 ms 3396 ms 3596 ms 3796 ms 3996 ms 4196 ms 4396 ms 4596 ms 4796 ms 4996 ms 5196 ms 5396 ms 5596 ms 5796 ms 5996 ms 6196 ms 6396 ms

Network

get-image.php (tutorial.seo-atelier.com)

Frames

258.2 ms 5.5 ms 56.0 ms

Timings

Experience

Main — https://tutorial.seo-atelier.com/beispiele/ds/

Layout Shift

Summary Bottom-Up Call Tree Event Log

**Layout Shift**

Warning [Cumulative Layout Shifts](#) can result in poor user experiences.

Score 0.1022

Cumulative Score 0.1022

Had recent input No

Moved from Location: [390,0], Size: [1140x78]

Moved to Location: [382,0], Size: [1140x78]

Moved from Location: [485,362], Size: [950x136]

Moved to Location: [477,759], Size: [950x136]

Moved from Location: [484,546], Size: [951x136]

Moved to Location: [0,0], Size: [0x0]

Moved from Location: [485,730], Size: [950x136]

Moved to Location: [0,0], Size: [0x0]

Moved from Location: [390,266], Size: [1140x96]

Moved to Location: [382,266], Size: [1140x493]

Related Node `div.container`

Total blocking time: 0.00ms [Learn more](#)

Elements Memory CSS Overview Sources Security Network Console Performance Lighthouse PageSpeed Application

```

<!DOCTYPE html>
<html lang="de" dir="ltr">
<head></head>
<body>
  <div id="wrapper">
    <header></header>
    <div class="container-fluid"></div>
    <main>
      <div class="container my-4"></div>
      <div class="container">
        <div class="row justify-content-center"> == $0
          
        </div>
        <div class="container textblock"></div>
        <div class="container textblock mt-5"></div>
        <div class="container textblock mt-5"></div>
      </div>
    </main>
  </div>
</body>
</html>

```

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

```

element.style {
}
.justify-content-center {
  -ms-flex-pack: center!important;
  justify-content: center!important;
}
.row {
  display: -ms-flexbox;
  display: flex;
  -ms-flex-wrap: wrap;
  flex-wrap: wrap;
  margin-right: -15px;
  margin-left: -15px;
}
*, ::after, ::before {
  box-sizing: border-box;
}
div {
  display: block;
}
Inherited from body
body {
  font-family: "Roboto",sans-serif;
  font-weight: 300;
}
body {
  margin: > 0;
  font-family: -apple-system,BlinkMacSystemFont,"Segoe UI",Roboto,"Helvetica Neue",Arial,"Noto Sans",sans-serif,"Apple Color Emoji","Segoe UI Emoji","Segoe UI Symbol","Noto Color Emoji";
  font-size: 1rem;
  font-weight: 400;
  line-height: 1.5;
  color: #212529;
  text-align: left;
}

```

html body div#wrapper main div.container div.row.justify-content-center

Console What's New Rendering Coverage Issues

- Paint flashing  
Highlights areas of the page (green) that need to be repainted. May not be suitable for people prone to photosensitive epilepsy.
- Layout Shift Regions**  
Highlights areas of the page (blue) that were shifted. May not be suitable for people prone to photosensitive epilepsy.
- Layer borders  
Shows layer borders (orange/olive) and tiles (cyan).
- Frame Rendering Stats  
Plots frame throughput, dropped frames distribution, and GPU memory.
- Scrolling performance issues  
Highlights elements (teal) that can slow down scrolling, including touch & wheel event handlers and other main-thread scrolling situations.
- Highlight ad frames  
Highlights frames (red) detected to be ads.
- Hit-test borders  
Shows borders around hit-test regions.
- Disable local fonts  
Disables local() sources in @font-face rules. Requires a page reload to apply.

ESC



Tutorial-Startseite

## Cumulative Layout Shift (CLS)



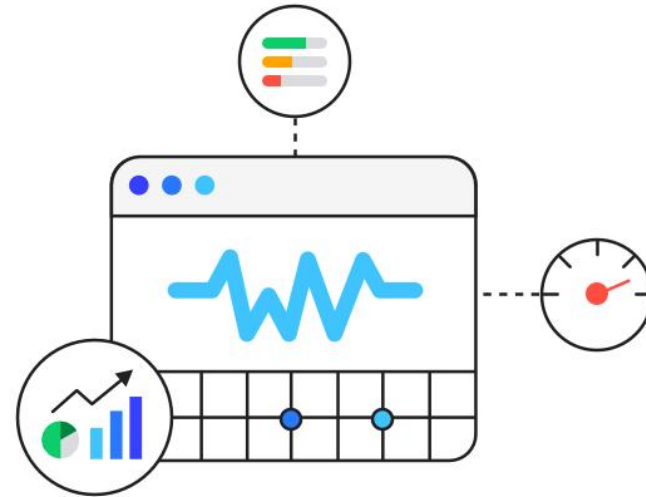
Lorem ipsum dolor sit amet, consetetur ["verrutschender" Link](#) sadipscing elit, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elit, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum [Wichtiger Link](#) dolor sit amet, consetetur sadipscing elit, sed diam nonumy eirmod

# Monitoring der Web Vitals?

## Web Vitals

Essential metrics  
for a healthy site

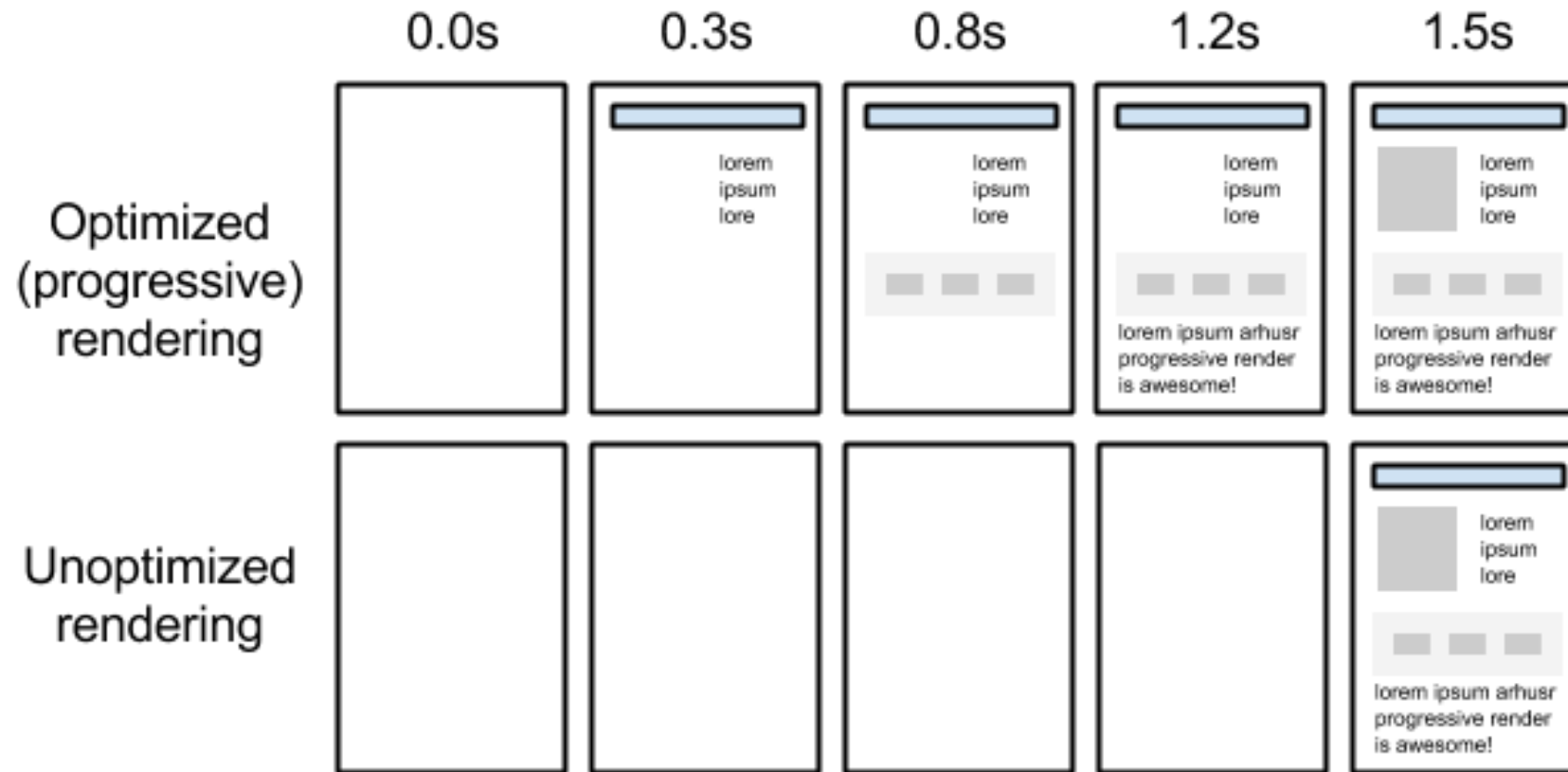


(Ende Teil I)

# Teil II



**Optimierung**



**Fragestellungen:**

- Wie entsteht der Critical-Rendering-Path
- Was blockiert das Parsing/Rendering?
- Wie können wir das optimieren?



# Grundsätzliches

## In den meisten Fällen...

- ist dein Hoster nicht schuld an deiner Performance!
- benötigst du kein CDN sofern deine Website nicht internationale Relevanz hat! Optimierung via CDN möglich, allerdings lassen sich diese Punkte meist auch lokal implementieren.

## Optimierung von Wordpress

- Vermeide Plugins, wo sie sich vermeiden lassen!
  - Oftmals lassen sich Plugins nach Verwendung wieder deinstallieren.
- Lösche inaktive Plugins & Themes!
- Nehme Änderungen möglichst ausschließlich im Child-Theme vor!
  - [Child Theme Generator](#)
- (besser 2x) Backup machen!

# Standards – leider können wir sie nicht alle behandeln

## Kleinere Dateien - schnellere Dateien

- Komprimierung für Text-Dateien
  - GZIP/Brotli
- Minimierung von Text-Dateien
- Komprimierung von Bildern

Wenig  
erklärungsbedürftig

## Browser-Caching

- Großes Thema

## Protokoll

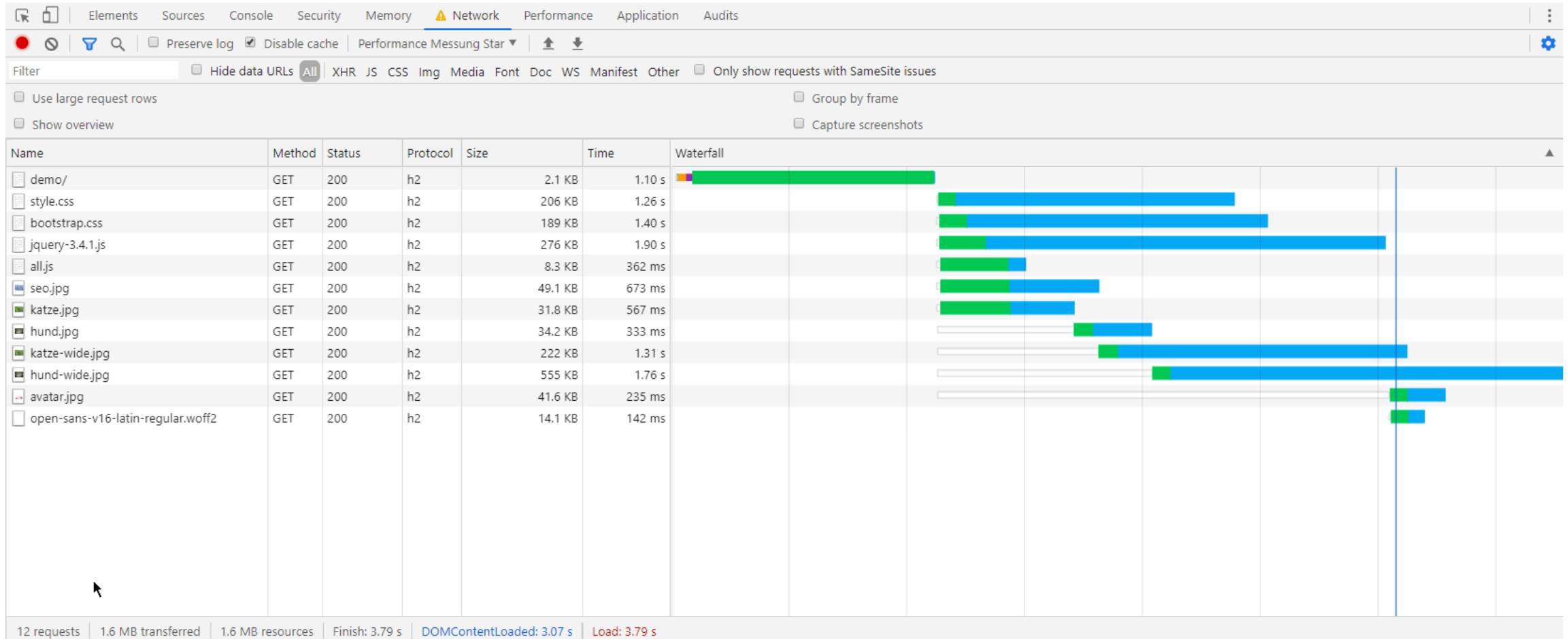
- Wird deine Website via HTTP2 ausgeliefert?
- <https://http2.pro/>
  - Hoster



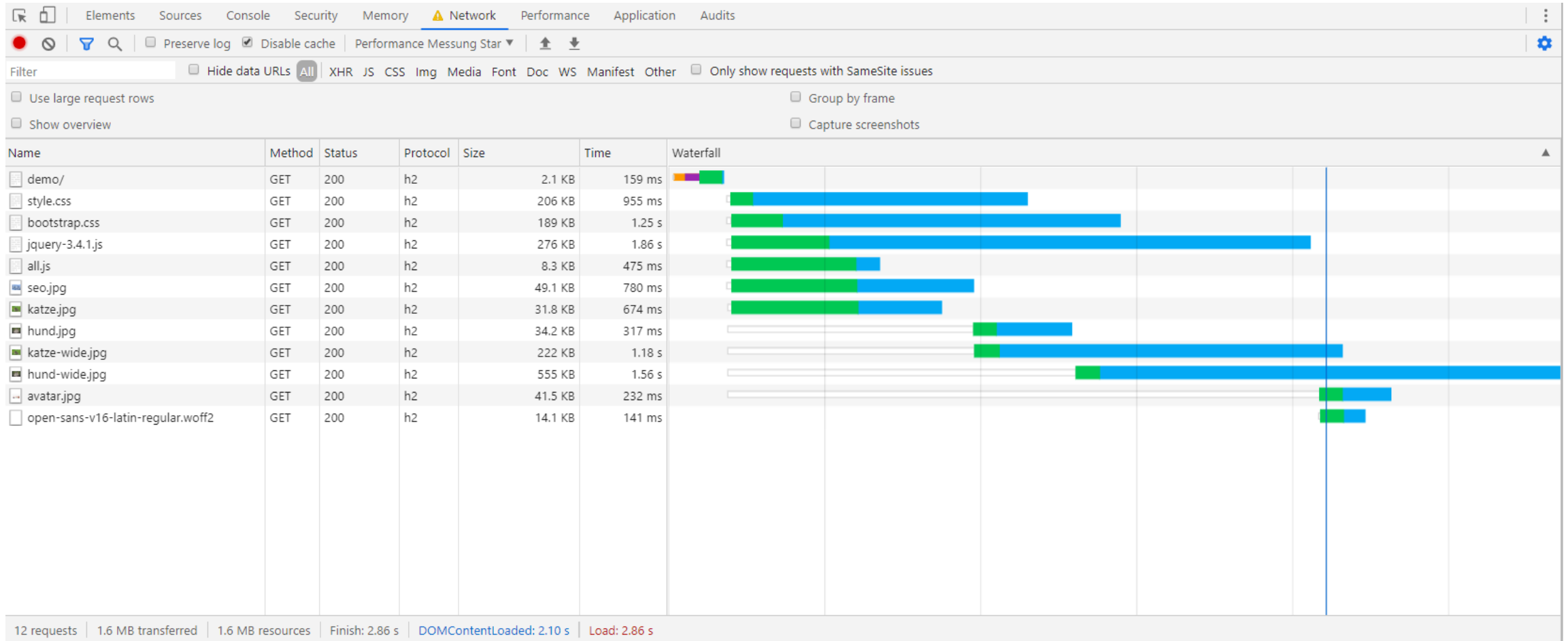


# **HTML Optimierung der TTFB**

# Die TTFB verschiebt ALLES nach hinten



# Die TTFB verschiebt ALLES nach hinten



# Optimierung der TTFB

## Caching auf seitenebene (bei „statischen“ Seiten)

- Seite muss bei Anforderung nicht mehr dynamisch erstellt werden.
  - DB und Code-Ausführung wird dadurch vermieden
  - Varnish, NGINX als Caching-Server
  - CDN
  - Die meisten WP-Caching-Plugins funktionieren auf diese Weise
  - Cache-Warming ratsam

# WP-Plugin Tipps

## Tipp: WP-Plugin

### Swift-Performance

- Cached Seiten als statische Dateien
- Legt als GZIP ab
- Text-Compression
- Image-Optimization (inkl. webp)

### WP Fastest Cache

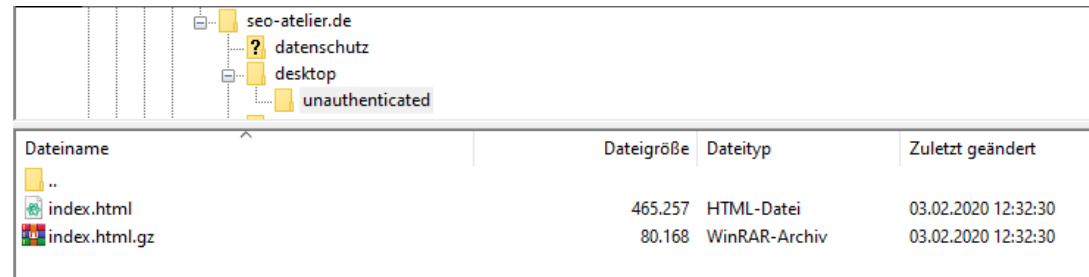
- Für ein schnelles & einfaches Setup

### WP Rocket

- Ähnlich umfangreich wie Swift-Performance

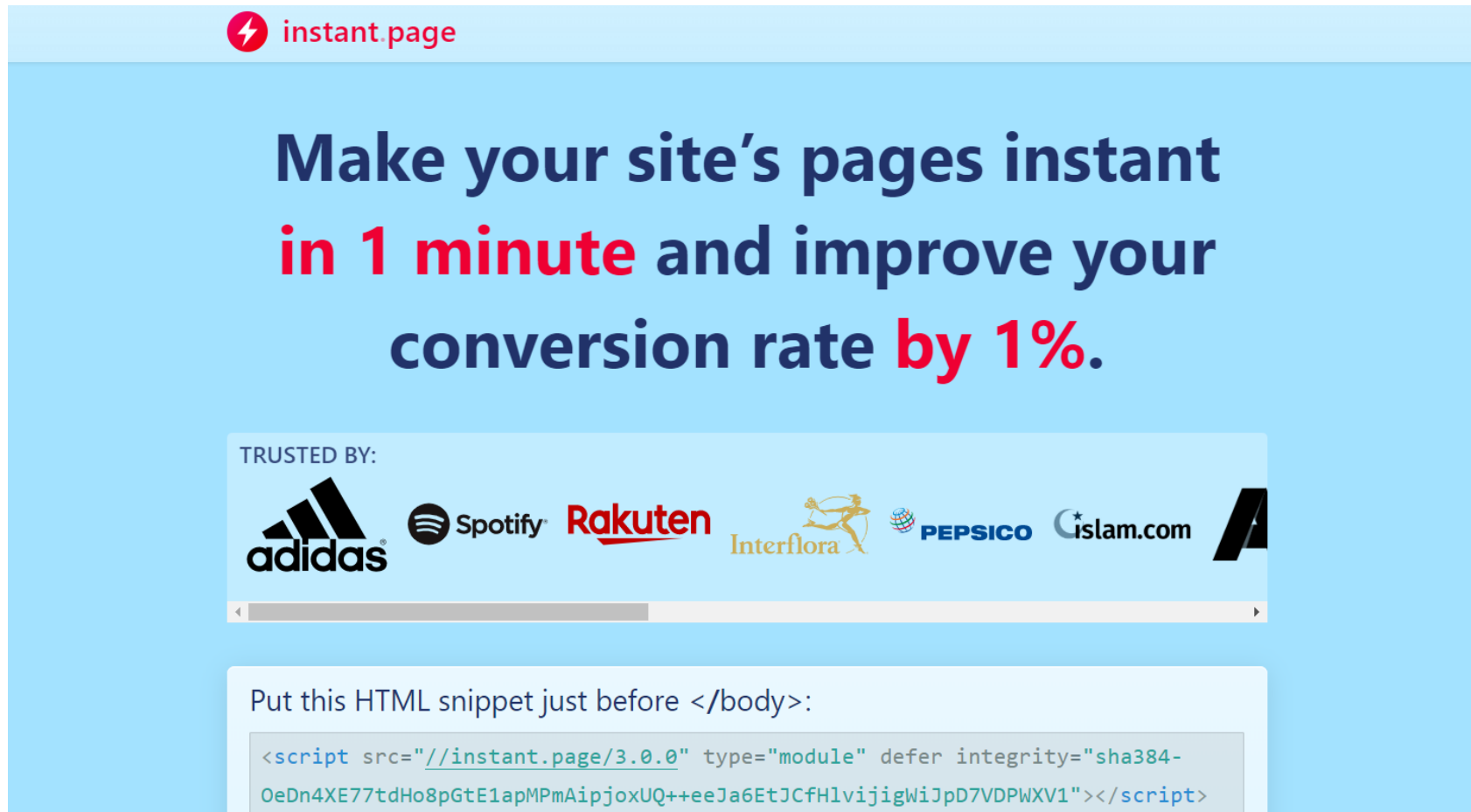
### W3 Total Cache

- Sehr umfangreich
- Benötigt tiefere Kenntnisse
- Ideal, sofern weiterführende Funktionalität benötigt wird.



Dateiname	Dateigröße	Dateityp	Zuletzt geändert
..	465.257	HTML-Datei	03.02.2020 12:32:30
index.html	80.168	WinRAR-Archiv	03.02.2020 12:32:30
index.html.gz	80.168	WinRAR-Archiv	03.02.2020 12:32:30

# instant.page



**Make your site's pages instant  
in 1 minute and improve your  
conversion rate by 1%.**

TRUSTED BY:

adidas Spotify Rakuten Interflora PEPSICO Islam.com

Put this HTML snippet just before `</body>`:

```
<script src="//instant.page/3.0.0" type="module" defer integrity="sha384-OeDn4XE77tdHo8pGtE1apMPmAipjoxUQ+eeJa6EtJCfHlvijigWiJpD7VPWXV1"></script>
```

Lädt die Seite vor, wenn der User mit der Maus über einen Link fährt. Seite liegt damit beim Klick bereits im Browser-Cache.

# How to do it in WP

Für die Integration steht ein WP-Plugin zur Verfügung: <https://wordpress.org/plugins/instant-page/>

## Besser selber machen:

1. Lade dir das Script herunter: <https://github.com/instantpage/instant.page>
2. Lade es auf per FTP auf deinen Server
3. Rufe es in deiner footer.php-Datei (direkt vor dem schließendem Body-Tag) auf

```
54     </div><!-- .section-inner -->
55
56     </footer><!-- #site-footer -->
57
58     <?php wp_footer(); ?>
59     <script src="instantpage-5.1.0.js" type="module" defer></script>
60 </body>
61 </html>
62
```

# Schriften



# Tipp Nr. 1 – Ladeketten vermeiden

■ Wichtige Anforderungen vorab laden 0,18 s ^

Mit `<link rel=preload>` können Sie das Abrufen von Ressourcen priorisieren, die derzeit beim Seitenaufbau erst später angefordert werden. [Weitere Informationen.](#)

URL	Mögliche Einsparungen
...fonts/amethysta-v9-latin-regular.woff2 (tutorial.seo-atelier.com)	180 ms

## Problem



## Besser



## Wie

- Einsatz von `link rel=„preload“`
- Setze im HTML-Head Preloads auf all deine Schriften ein, welche above the fold genutzt werden
- Beschränke dich auf das Format woff2

```
<link rel="preload" href="https://www.domain.de/fonts/open-sans-v16-latin-regular.woff2" as="font" crossorigin="anonymous"/>
```

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <meta name="robots" content="none"/>
    <link href="css/style.css" rel="stylesheet" />
    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet" />
    <link rel="preload" href="/fonts/open-sans-v16-latin-700.woff2" as="font" crossorigin="anonymous"/>
```

## Largest Contentful Paint (LCP) - ohne Bilder

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor

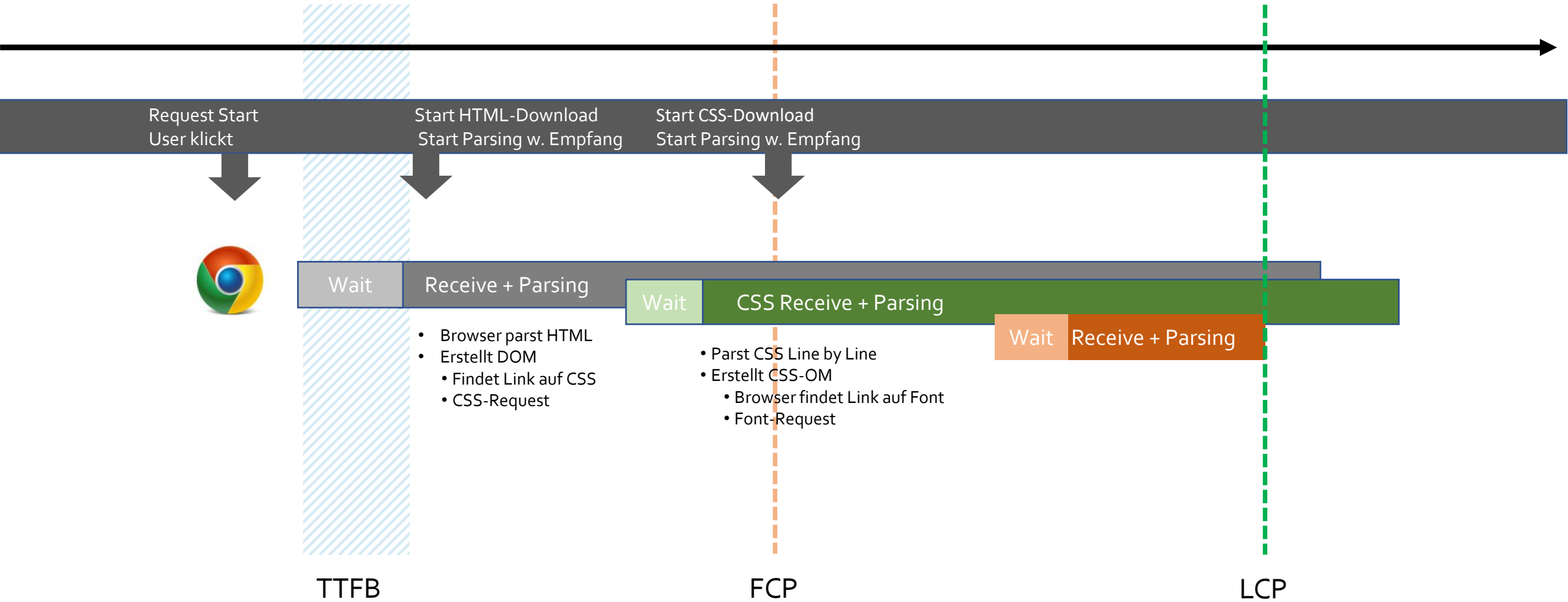
At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

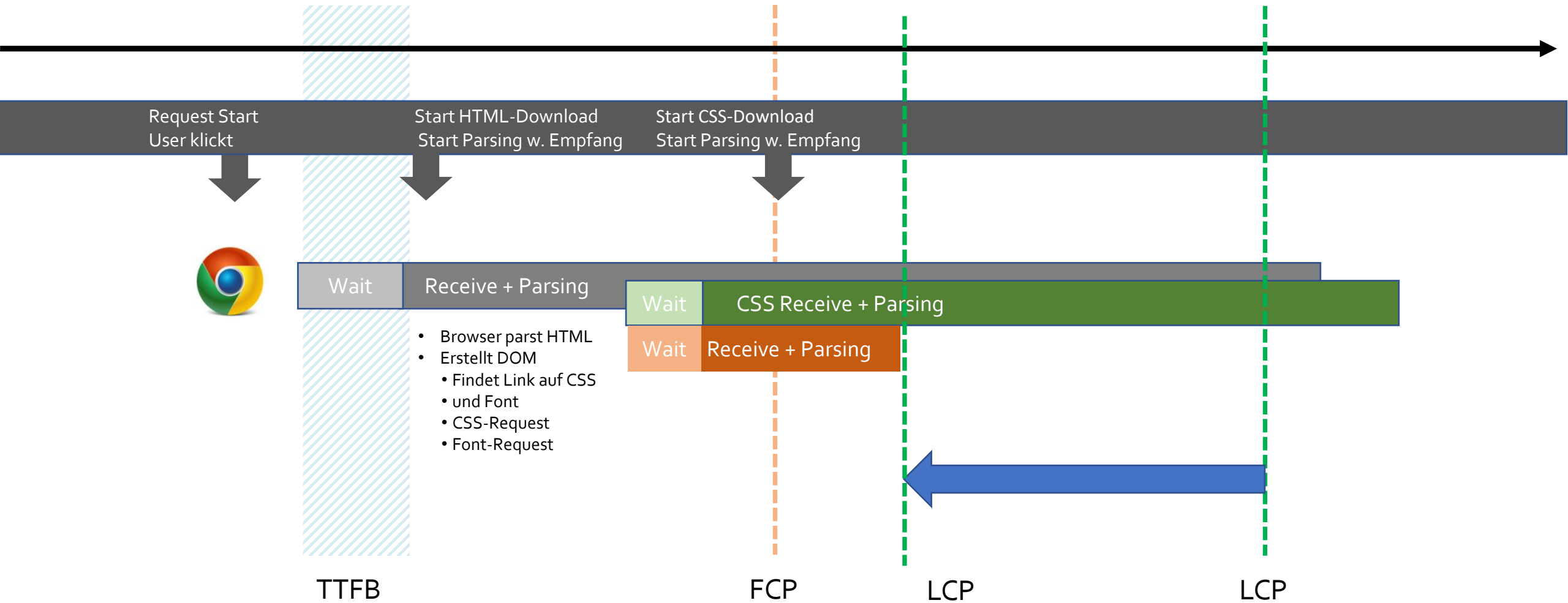
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et justo duo dolores et ea rebum.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy

# Effekt



# Effekt



# Unkritische Schriften

## Problem

- FOIT vs. FOUT
- Text wird nicht dargestellt, bis Font-Datei geladen und verarbeitet

## Besser

Schriften „unkritisch“ einbinden

## Wie

Nutzung von CSS-Eigenschaft „font-display: swap;“

Vgl.:

<https://fonts.googleapis.com/css2?family=Roboto>

<https://fonts.googleapis.com/css2?family=Roboto&display=swap>

- Lokale Nutzung von Schriften ändert nichts an der Einbindung

- Analog mit allen Schriften verfahren, außer Icon-Fonts.

▲ Darauf achten, dass der Text während der Webfont-Ladevorgänge sichtbar bleibt ^

Verwenden Sie die CSS-Funktion font-display, damit der Text für Nutzer sichtbar ist, während Webfonts geladen werden. [Weitere Informationen.](#)

URL

...fonts/amethysta-v9-latin-regular.woff2 (tutorial.seo-atelier.com)

Mögliche  
Einsparungen

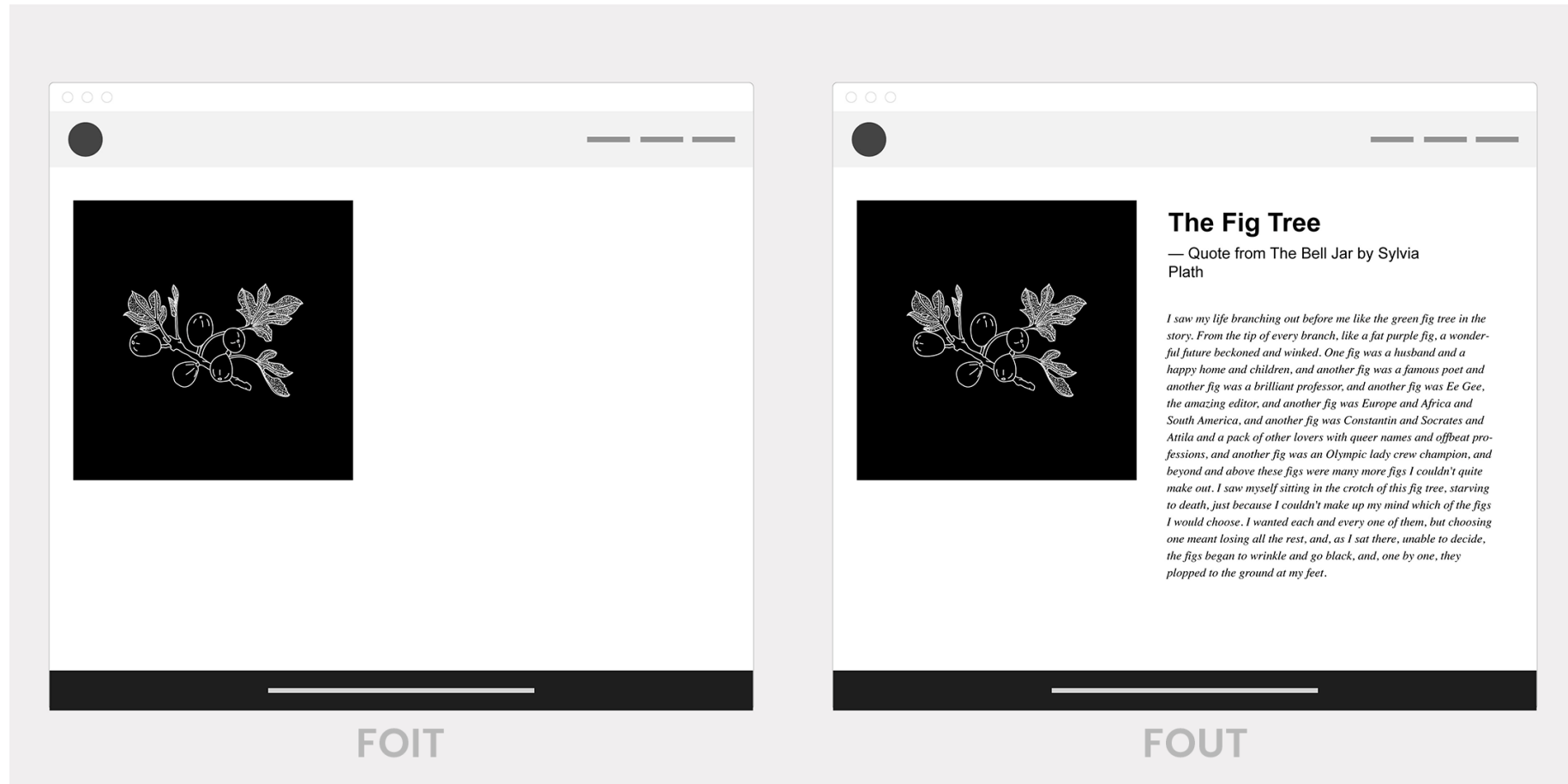
70 ms

# Unkritische Schriften – lokale Einbindung im Detail

```
/* open-sans-300 - latin */
@font-face {
  font-family: 'Open Sans';
  font-style: normal;
  font-weight: 300;
  src: local('Open Sans Light'), local('OpenSans-Light'),
       url('../fonts/open-sans-v15-latin-300.woff2') format('woff2'), /* Chrome 26+, Opera 23+, Firefox 39+ */
       url('../fonts/open-sans-v15-latin-300.woff') format('woff'); /* Chrome 6+, Firefox 3.6+, IE 9+, Safari 5.1+ */
  font-display: swap;
}
/* open-sans-300italic - latin */
@font-face {
  font-family: 'Open Sans';
  font-style: italic;
  font-weight: 300;
  src: local('Open Sans Light Italic'), local('OpenSans-LightItalic'),
       url('../fonts/open-sans-v15-latin-300italic.woff2') format('woff2'), /* Chrome 26+, Opera 23+, Firefox 39+ */
       url('../fonts/open-sans-v15-latin-300italic.woff') format('woff'); /* Chrome 6+, Firefox 3.6+, IE 9+, Safari 5.1+ */
  font-display: swap;
}
/* open-sans-regular - latin */
@font-face {
  font-family: 'Open Sans';
  font-style: normal;
  font-weight: 400;
  src: local('Open Sans Regular'), local('OpenSans-Regular'),
       url('../fonts/open-sans-v15-latin-regular.woff2') format('woff2'), /* Chrome 26+, Opera 23+, Firefox 39+ */
       url('../fonts/open-sans-v15-latin-regular.woff') format('woff'); /* Chrome 6+, Firefox 3.6+, IE 9+, Safari 5.1+ */
  font-display: swap;
}
```

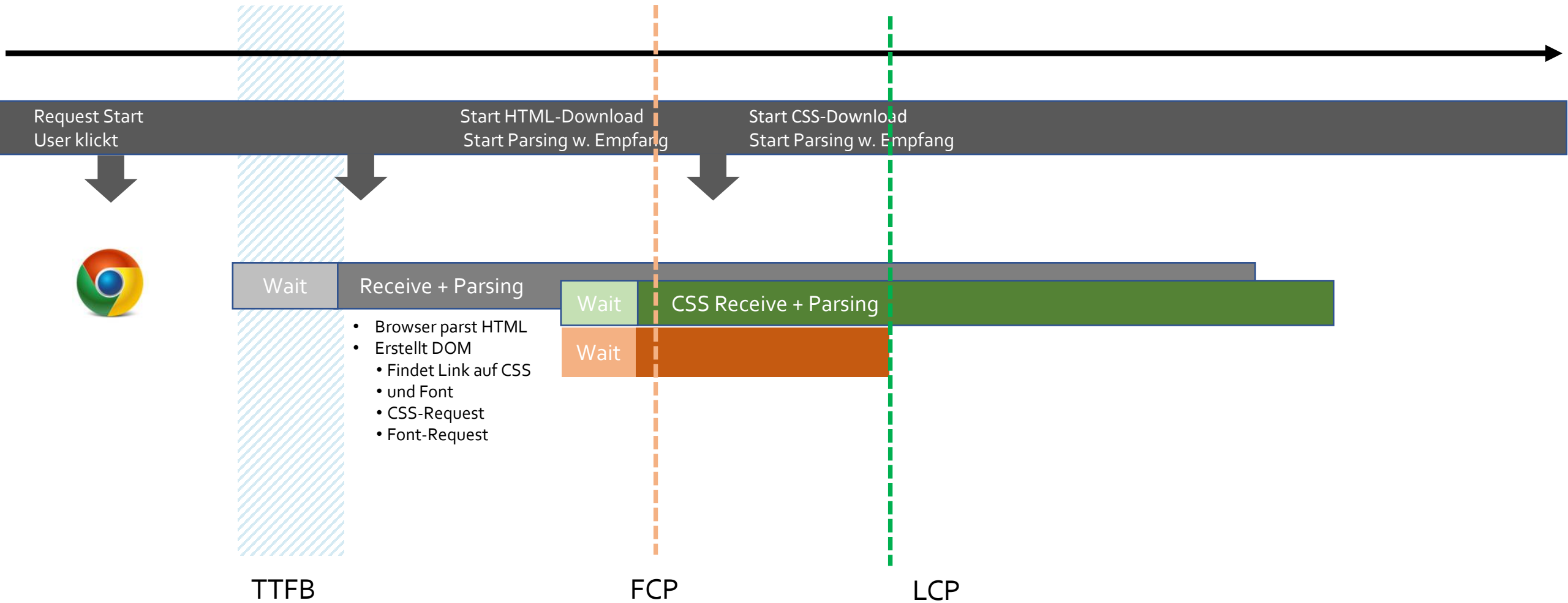
- Füge in die Aufrufe der Schriften im CSS diese Eigenschaft ein!
- Lasse dabei Icon-Fonts aus!

# Schriften - FOIT vs. FOUT

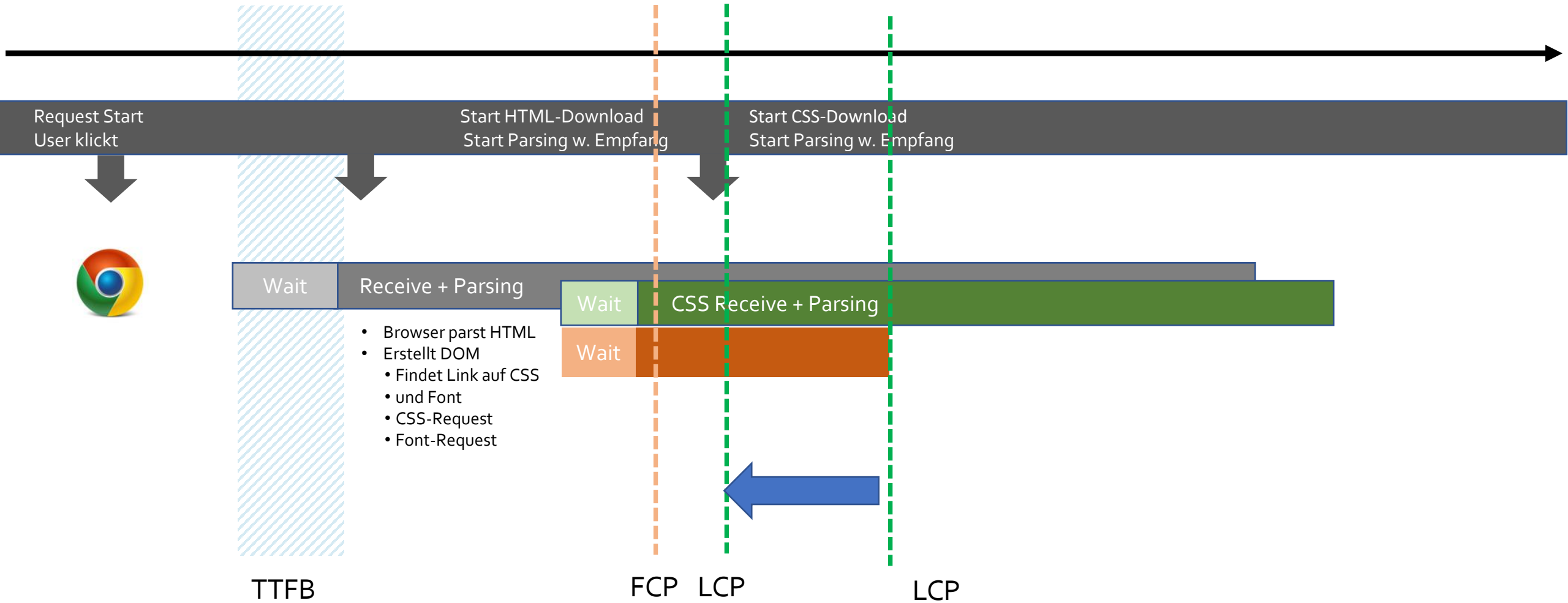




# Effekt



# Effekt



**CSS**

# CSS Optimierung

## Problem

- CSS ist keine Parser-Blocking Ressource
- CSS ist aber Rendering-Blocking

## CSS bereinigen

- @import vermeiden bzw. entfernen !
- Keine Style-Tags innerhalb des Bodys
- !important vermeiden
- Kombinieren von CSS-Klassen
- Mobile-First-Ansatz

## CSS-Analyse

- <https://stylelint.io/>

# Inline critical CSS

Above-the-fold-CSS kann Inline im Head integriert werden.

```
1 <!DOCTYPE html>
2 <html lang="de-DE" prefix="og: http://ogp.me/ns#">
3   <head>
4     <meta charset="UTF-8">
5     <link href="css/style.css" rel="stylesheet" />
6     <title>Testseite | SEO-Atelier.de</title>
7     <script async type='text/javascript' src='js/all.min.js'></script>
8   </head>
9   <body>
```

# Inline critical CSS

Above-the-fold-CSS kann Inline im Head integriert werden.

```
1 <!DOCTYPE html>
2 <html lang="de-DE" prefix="og: http://ogp.me/ns#">
3   <head>
4     <meta charset="UTF-8">
5     <style>
6       body {
7         background-color: #eee;
8       }
9       h1 {
10        color: black;
11        margin-left: 40px;
12      }
13    </style>
14    <title>Testseite | SEO-Atelier.de</title>
15    <script async type='text/javascript' src='js/all.min.js'></script>
16  </head>
17  <body>
```

# Inline critical CSS

- Nachträgliche Implementierung oftmals etwas „tricky“
- Verursacht gerne sog. FOUC (Flash of unstyled Content)

## Achtung:

- Kann im Client nicht mehr gecached werden.
- Macht nur Sinn, wenn Inline-Style samt HTML nicht zu groß.
  - Google empfiehlt max. 1-3 Roundtrips für den above-the-fold Code (HTML + inlined CSS).
  - TCP congestion control limitiert den Durchsatz bei Roundtrips.
  - 10 TCP-Pakete sind im ersten Roundtrip max. möglich.
  - Ab einer Größe von 98 KB von <HTML> - „Above the fold-Ende“ ist diese Technik nachteilig (14 KB + 28 KB + 56 KB)
  - Besser: Unter 14,6KB

## Tools

- [critical](#) (Google-Empfehlung)
- [criticalCSS](#) (Google-Empfehlung)
- [grunt-critical](#)

# Media-Queries im CSS-Aufruf

Media-Queries können nicht nur im CSS-File hinterlegt werden.

```
10 <link rel="stylesheet" media="all" href="cssbase.css" />
11 <link rel="stylesheet" media="(min-width: 672px)" href="csswide.css" />
12 <link rel="stylesheet" media="(orientation:landscape)" href="landscape.css">
```

- Lediglich deutlich kleinere Files müssen bei Bedarf heruntergeladen werden.
- Rendering-kritisches CSS kann somit auf ein Minimum reduziert werden.




# Preloading im HTTP-Header

Einige ms lassen sich zusätzlich dadurch herausholen, dass das HTML-Parsing (bis zum Finden der CSS-Verlinkung) nicht abgewartet werden muss.

.htaccess-Code:

```
<IfModule mod_headers.c>
  <FilesMatch "\.(html)$">
    Header add Link "<https://www.domain.de/style.css>; rel=preload; as=style; nopush"
  </FilesMatch>
</IfModule>
```



Wichtig, da andernfalls der Preload zum HTTP2 Serverpush wird, was (zum aktuellen Stand) ein Browser-Caching unmöglich machen würde.

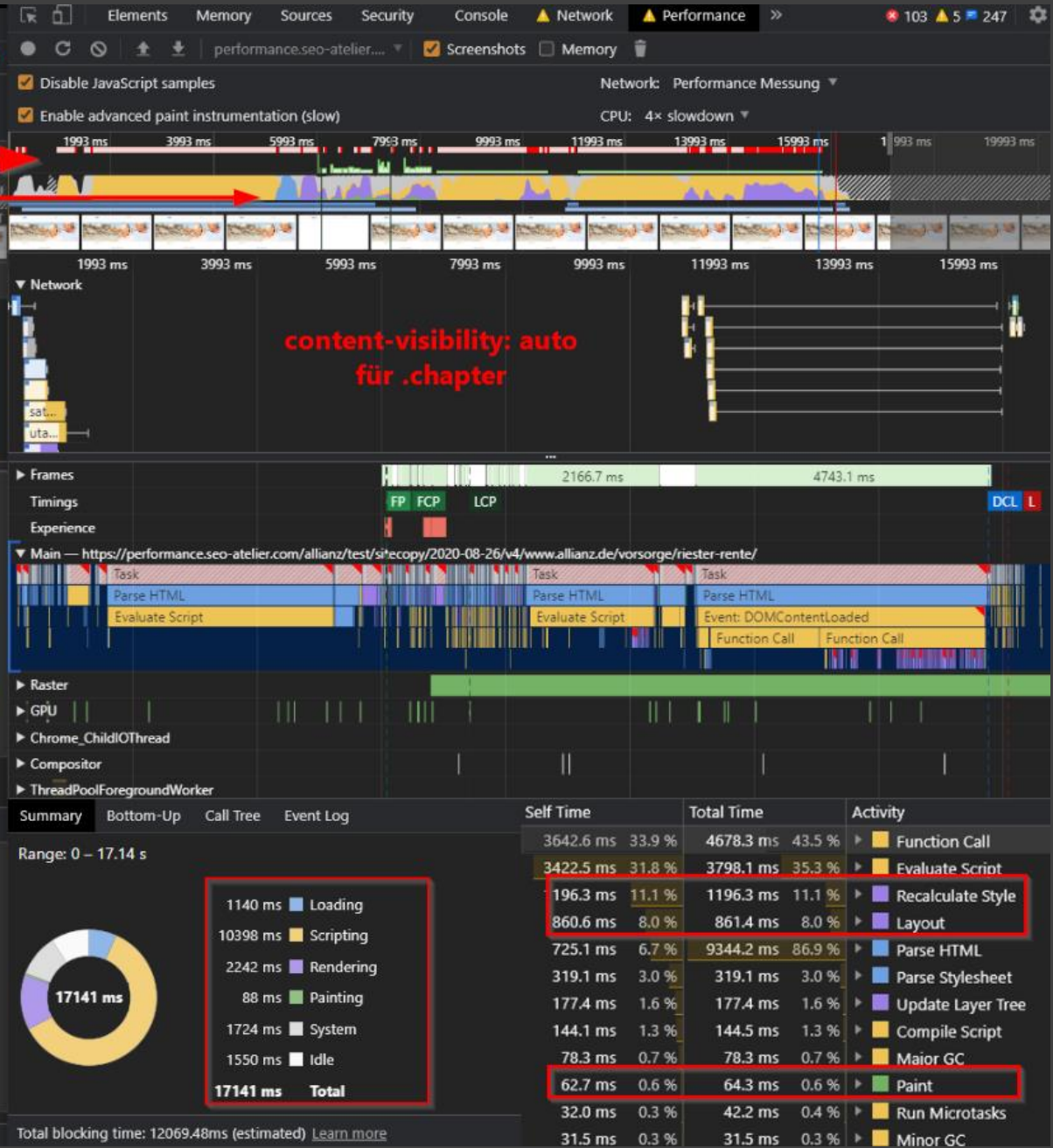
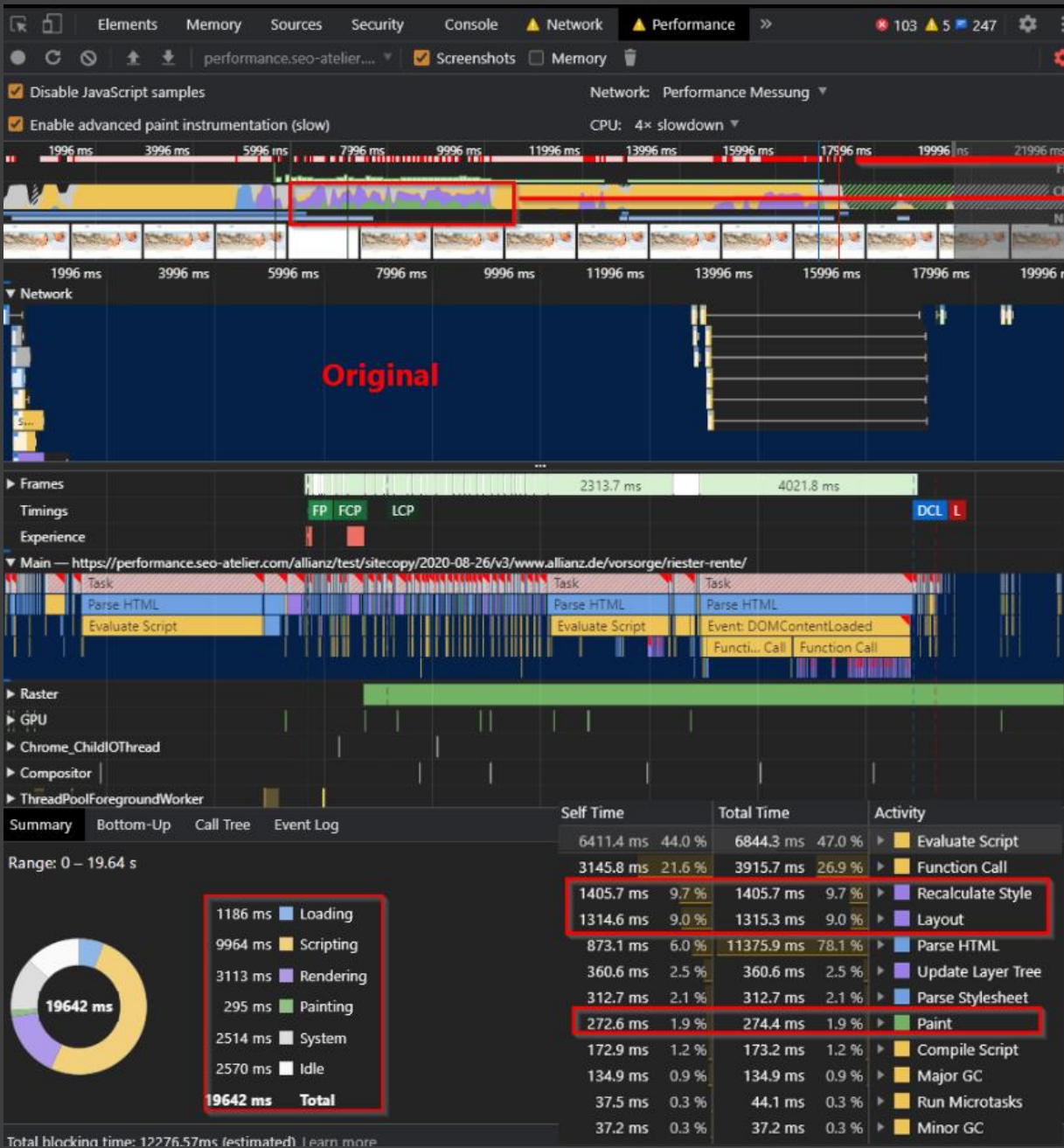
Funktioniert auch für Schriften, dann müssen allerdings die zusätzlichen Angaben "type=font/woff2; crossorigin;" angefügt werden.

# Rendering-Optimierung mit content-visibility

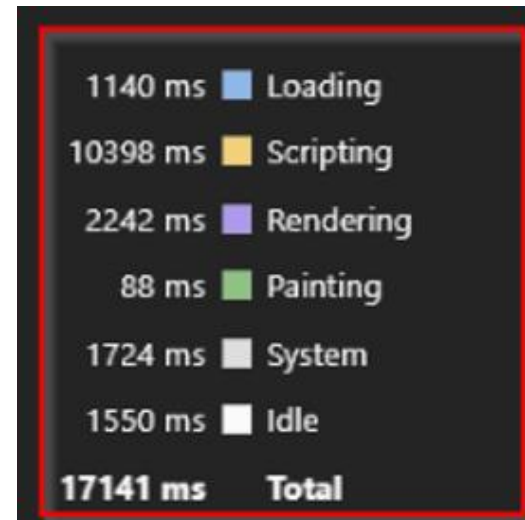
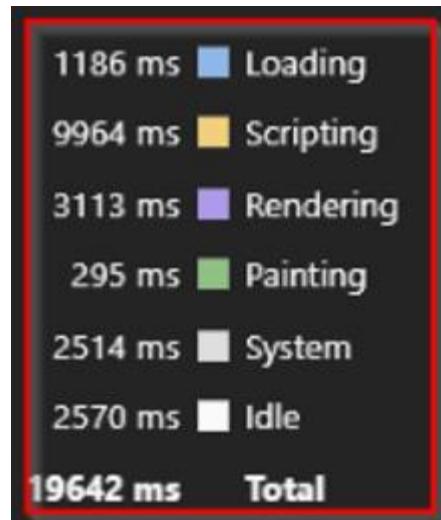
Seit kurzem unterstützt Chrome "Lazy Rendering".

```
61  .content-visibility {  
62    content-visibility: auto;  
63    contain-intrinsic-size: 1000px;  
64  }  
65
```

- Geeignete Elemente können alle Abschnitte below-the-fold sein.
- Deren CSS-Klassen können mit der neuen Eigenschaft im Stylesheet nachgerüstet werden.
  - Obiges Beispiels zeigt separate Klasse, welche bei jedem Block below-the-fold eingefügt wurde.
- Eine "intrinsic-size" muss vergeben warden, da sonst 0.
  - Diese muss nicht mit dem späteren Element matchen.



# Rendering-Optimierung mit content-visibility





**JS**

# Synchrones JS

```
<script src="/assets/js/jquery.min.js"></script>
```

**JS kann DOM & CSSOM manipulieren!**

## - **Parser-Blocking & Rendering Blocking**

- HTML-Parsing wird an der Stelle des Script-Tags unterbrochen.
- JS-Download muss heruntergeladen werden.
- Das CSS muss vollständig verarbeitet werden. (Da JS auch das **CSSOM** manipulieren kann.)
- JS muss ausgeführt werden.

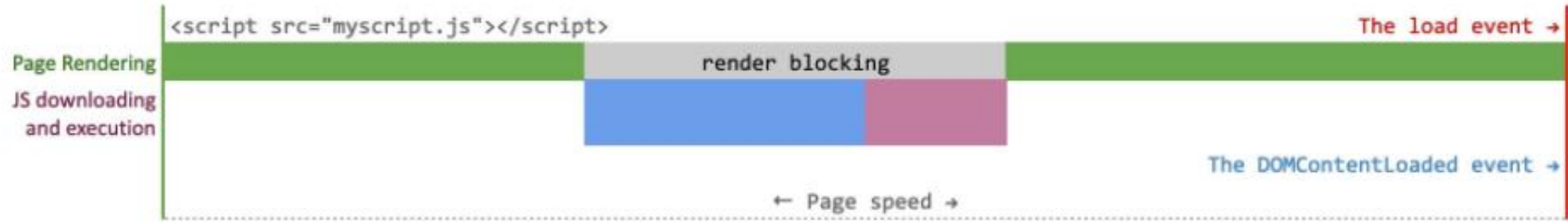
Nach dem Abarbeiten der obigen Punkte kann das HTML-Parsing (und damit das Rendering) fortgesetzt werden.

**! Auch inline-Scripts verursachen diesen Umstand !**

## JS Pagespeed-freundlich einbinden

```
17  
18 <script src="/assets/js/all.js"></script>  
19  
20 <script async src="/assets/js/all.js"></script>  
21  
22 <script defer src="/assets/js/all.js"></script>  
23
```

# JS Pagespeed-freundlich einbinden





Maximilian Bloch  
SEO-Consultant

Tel.: 0821 80735508  
[mb@seo-atelier.de](mailto:mb@seo-atelier.de)  
<https://seo-atelier.de>

SEO-Atelier  
Bachstr. 21  
86420 Diedorf

