

WordPress-Performance

Walter Ebert
wee.press



wpmeetup-frankfurt.de

Rich Snippets jetzt international verfügbar

Schrägstrich oder nicht?

Wie man unerwünschte Inhalte loswird, Teil III: Entfernen von Inhalten, die nicht euch gehören

Unterstützt Google bei der Indexierung eurer Videos

Malware-Funktion in den Search Console aktualisiert

Webmaster – Google-Dienste im Hosting-Bereich konfigurieren

Mehr Daten und Diagramme in „Häufigste Suchanfragen“

Bessere Rezepte im Web: Einführung von Rich Snippets für Rezepte

Websitegeschwindigkeit im Ranking der Internetsuche verwenden

Wann und wieso wurde meine Website wegen Malware gekennzeichnet? Informationen sind fast in Echtzeit verfügbar.

Hinzufügen von Bildern zu euren XML-Sitemaps

[Startseite](#) > [Search Central](#) > [Blog von Google Search Central](#)

War das hilfreich?  

[Feedback geben](#)

Websitegeschwindigkeit in den Suchergebnissen

Mittwoch, 26. Mai 2010

Ihr wisst wahrscheinlich schon, dass wir bei Google Geschwindigkeit lieben – in [unseren Produkten](#) und [im Web](#). Als Teil dieser Bemühungen berücksichtigen wir seit einiger Zeit in unseren Suchranking-Algorithmen auch die Geschwindigkeit von Websites. Die Websitegeschwindigkeit gibt an, wie schnell eine Website auf Webanfragen reagiert.

Es ist wichtig, die Geschwindigkeit von Websites zu verbessern – und zwar nicht nur für Website-Inhaber, sondern für alle Internetnutzer. Schnellere Websites bedeuten glückliche Nutzer – eine langsame Reaktionszeit einer Website führt laut unseren [internen Studien](#) dazu, dass die Besucher weniger Zeit dort verbringen. Schnellere Websites verbessern jedoch nicht nur die Nutzererfahrung. Eine Verbesserung der Websitegeschwindigkeit bedeutet, wie aktuelle Daten zeigen, auch [niedrigere Betriebskosten](#). Genauso wie uns ist auch unseren Nutzern Geschwindigkeit sehr wichtig. Daher haben wir uns dafür entschieden, die Websitegeschwindigkeit bei unseren Suchrankings zu berücksichtigen. Wir ziehen viele Faktoren heran, um die Geschwindigkeit einer Website im Vergleich zu anderen Websites zu ermitteln.

Wenn ihr Website-Inhaber, Webmaster oder Webautoren seid, könnt ihr die Geschwindigkeit eurer Websites mithilfe folgender kostenloser Tools ermitteln:

translated by Google Diese Seite wurde von der [Cloud Translation API](#) übersetzt.[SWITCH TO ENGLISH](#)

Startseite > Blog

War das hilfreich?  

Interaction to Next Paint wird am 12. März zu einem Core Web Vitals-Messwert

„Interaction to Next Paint“ wird offiziell zu einem Core Web Vitals-Messwert und ersetzt am 12. März First Input Delay.



Jeremy Wagner



Rick Viscomi



Das [Web Vitals-Programm](#) enthält Messwerte, mit denen Webentwickler wichtige Aspekte der Nutzererfahrung im Web analysieren können. Der [First Input Delay \(FID\)](#) repräsentierte den Reaktionsaspekt des Programms. Im Laufe der Zeit wurde jedoch deutlich, dass ein neuer Messwert erforderlich war, um Aspekte der Interaktivität im Web zu erfassen, die FID nicht bot. Das Chrome-Team hat dies erkannt und im Mai 2022 [Interaction to Next Paint \(INP\)](#) als [experimentellen Messwert](#) eingeführt. Letztes Jahr wurde INP zu einem [ausstehenden Messwert](#) und [wir haben angekündigt](#), dass INP im März 2024 auf den [stabilen Status](#) hochgestuft wird.

Heute geben wir bekannt, dass **INP offiziell ein Core Web Vital wird und FID am 12. März dieses Jahres ersetzt**. Außerdem wird **FID im Rahmen dieser Umstellung eingestellt**.

[May 2022](#)[May 2023](#)[March 2024](#)

Auf dieser Seite

Wie kann ich mich vorbereiten?

Was passiert, sobald INP ein Core Web Vitals-Messwert wird?

Der Weg in die Zukunft

Website carbon results for: wpmeetup-frankfurt.de



Hurrah! This web page achieves a carbon rating of A⁺

This is cleaner than **95%** of all web pages globally



Learn about our [rating system](#)

This page was last tested on 11 Mar, 2024.

Copy URL



Total Kilobytes

The sum of [transfer size](#) kilobytes of all resources requested by the page.

See also: [Page Weight](#)

MEDIAN DESKTOP
2831.9 KB
▲23.7%

MEDIAN MOBILE
2553.7 KB
▲11.9%

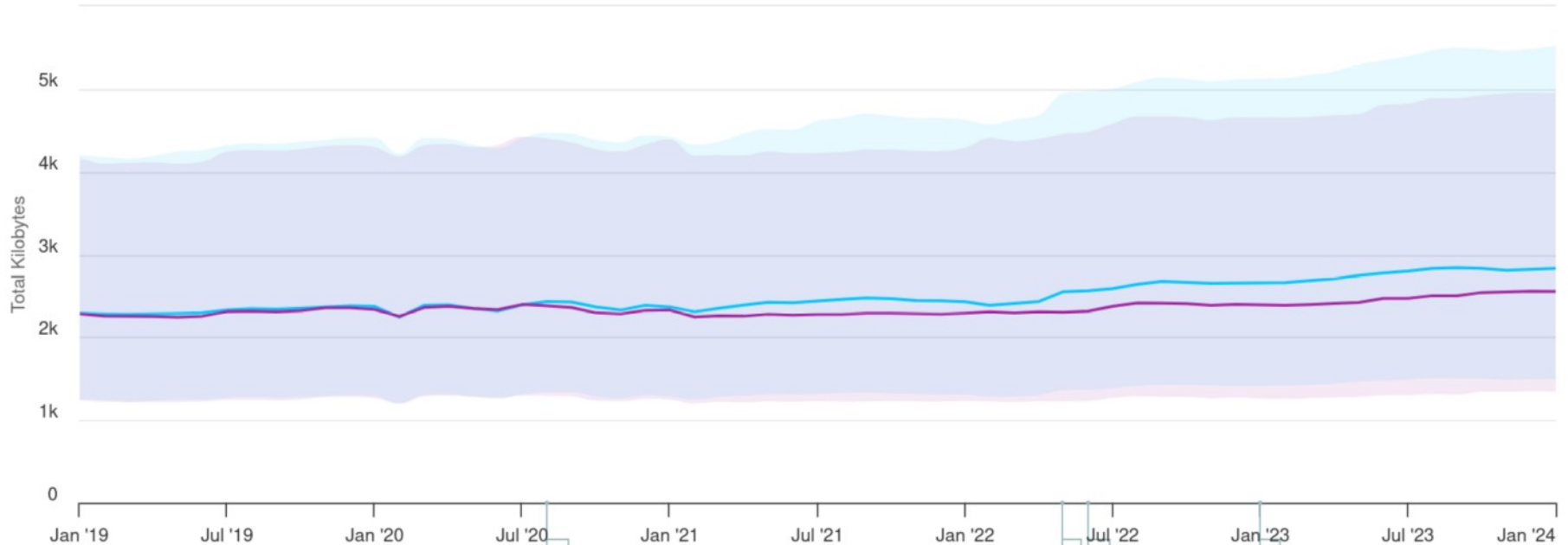
WordPress: Timeseries of Total Kilobytes

Source: httparchive.org



Zoom 1m 3m 6m YTD 1y 3y All

1 Jan 2019 → 1 Jan 2024



PageSpeed Insights



Mobil



Desktop

0 / 100 Übersicht über die Vorschläge

Behebung erforderlich:

Bilder optimieren

▶ [Fehlerbehebung anzeigen](#)

Komprimierung aktivieren

▶ [Fehlerbehebung anzeigen](#)

Browser-Caching nutzen

▶ [Fehlerbehebung anzeigen](#)

Behebung empfohlen:

JavaScript- und CSS-Ressourcen, die das Rendering blockieren, in Inhalten "above the fold" (ohne Scrollen sichtbar) beseitigen

▶ [Fehlerbehebung anzeigen](#)







Joe McGill

3:56 am on November 10, 2015

Responsive Images in WordPress 4.4

WordPress 4.4 will add native responsive image support by including `srcset` and `sizes` attributes to the image markup it generates. For background on this feature, read the [merge proposal](#).

How it works

WordPress automatically creates several sizes of each image uploaded to the media library. By including the available sizes of an image into a `srcset` attribute, browsers can now choose to download the most appropriate size and ignore the others—potentially saving bandwidth and speeding up page load times in the process.

To help browsers select the best image from the source set list, we also include a default `sizes` attribute that is equivalent to `(max-width: {{image-width}}px) 100vw, {{image-width}}px`. While this default will work out of the box for a majority of sites, themes should customize the default `sizes` attribute as needed using the `wp_calculate_image_sizes` [filter](#).

Note that for compatibility with existing markup, neither `srcset` nor `sizes` are added or modified if they already exist in content [HTML](#).

Search ...

Email Updates

Enter your email address to subscribe to this blog and receive notifications of new posts by email.

Email Address

Subscribe

Join 5,868 other subscribers

Recent Updates

Recent Comments

No Replies

Current Release

The current release in progress is



Justin Ahinon

10:59 pm on October 9, 2019

Introducing handling of big images in WordPress 5.3

The way WordPress handles large images has always been a topic of discussion for users and developers.

There are generally two types of images that are uploaded:

- Images that have been edited or created in an image editing application.
- Photos that are uploaded either directly from the camera or haven't been edited.

In the first case, the images are usually “web-ready”. They may have been scaled down to an appropriate size and optimized.

In the second case, the images are usually much bigger than needed and are not optimized for web use. A photo taken with an average modern smartphone is easily over 5MB in file size. Photos taken with a good quality camera can be much larger.

WordPress 5.3 introduces a new way to manage these images by detecting big images and generating a “web-optimized maximum size” of them.

How does it work?

When a new image is uploaded, WordPress will detect if it is a “big” image by checking

Search ...

Email Updates

Enter your email address to subscribe to this blog and receive notifications of new posts by email.

Email Address

Subscribe

Join 5,868 other subscribers

🏠 Recent Updates

💬 Recent Comments

💬 No Replies

Current Release

The current release in progress is



Felix Arntz

12:24 am on July 15, 2020

Lazy-loading images in 5.5

In WordPress 5.5, images will be lazy-loaded by default, using the [native HTML loading attribute](#) which [became a web standard earlier in 2020](#). This will drastically save bandwidth on both servers as well as user agents across sites where images further down the page used to be loaded right away, even in the case the user might never scroll towards them.

By default, WordPress will add `loading="lazy"` to all `img` tags that have `width` and `height` attributes present. Technically this is handled on page output, similar to how responsive images are facilitated in WordPress by adding `srcset` and `sizes` attributes. To improve server-side performance of the two features, a new `wp_filter_content_tags()` function has been introduced so that `img` tags only need to be parsed once while then deferring the modifications to more specific functions related to the feature.

See [#44427](#) for the overarching [Trac](#) ticket.

Reduced layout shifting as a prerequisite

A common user experience problem in modern website is so-called [layout shifting](#), often caused by slow-loading media resources like images: By default, only after an image is loaded, the browser can layout the page correctly, which results in the

Search ...

Email Updates

Enter your email address to subscribe to this blog and receive notifications of new posts by email.

Email Address

Subscribe

Join 5,868 other subscribers

🏠 Recent Updates

💬 Recent Comments

💬 No Replies

Current Release

The current release in progress is

Keyboard Shortcuts | Hide comment threads



Bethany Chobanian Lang

8:08 pm on July 26, 2022



...

Welcome to the Core Performance Team blog!

Welcome to the Core Performance Team blog!

Because our work is so integrated with WordPress core, plugins, and themes, our blog posts are typically shared on those blogs instead of being posted here.

You can view all performance-related blog posts – including core feature and merge proposals, weekly chat agenda and meeting notes, and other discussions – by checking out our tags on those other blogs:

- <https://make.wordpress.org/core/tag/performance/>
- <https://make.wordpress.org/plugins/tag/performance/>

Search ...

Email Updates

Enter your email address to subscribe to this blog and receive notifications of new posts by email.

Email Address

Subscribe

Join 29 other subscribers



Performance Lab

Von [WordPress Performance Team](#)

[Herunterladen](#)[Details](#)[Rezensionen](#)[Installation](#)[Entwicklung](#)[Support](#)

Beschreibung

Das Plugin Performance Lab ist eine Sammlung von Modulen, die darauf spezialisiert sind, die Leistung deiner Website zu verbessern. Die meisten davon sollen letztendlich zum WordPress-Core hinzugefügt werden. Das Plugin erlaubt es, die Module einzeln zu aktivieren und zu testen, bevor diese im WordPress-Core zur

Community-Plugin

[Mitwirken](#)

Dieses Plugin wird von einer Community entwickelt und unterstützt.

Version:

2.9.0



Datum des Berichts: 12.03.2024, 12:00:00

https://wpmeetup-frankfurt.de/

Analysieren



Mobil



Computer



So sieht die Leistung auf der Nutzerseite aus



Keine Daten



Leistungsprobleme diagnostizieren

100

Leistung

100

Barrierefreiheit

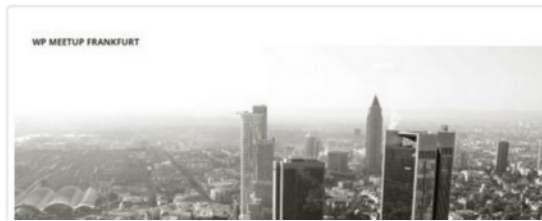
100

Best Practices

100

SEO

100





Mobil



Computer



Leistungsprobleme diagnostizieren



Leistung



Barrierefreiheit



Best Practices



SEO



Leistung

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



0–49



50–89



90–100

MESSWERTE

Ansicht maximieren



Datum des Berichts: 12.03.2024, 12:59:52

https://de.wordpress.org/

Analysieren



Mobil



Computer



So sieht die Leistung auf der Nutzerseite aus

Diese URL

Ursprung



Core Web Vitals-Bewertung: **nicht bestanden** ?

Ansicht maximieren

● Largest Contentful Paint (LCP)

1,5 s



● Interaction to Next Paint (INP)

25 ms



■ Cumulative Layout Shift (CLS)

0,2



ANDERE WICHTIGE MESSWERTE

● First Contentful Paint (FCP)

1,5 s



● First Input Delay (FID) ⓘ

2 ms



● Time to First Byte (TTFB) ⓘ

0,8 s





Mobil



Computer

DIAGNOSE

- ▲ Largest Contentful Paint-Element — 2.430 ms ▼
- ▲ Bilder in modernen Formaten bereitstellen — Mögliche Einsparung von 721 KiB ▼
- ▲ Bilder richtig dimensionieren — Mögliche Einsparung von 618 KiB ▼
- ▲ Ressourcen beseitigen, die das Rendering blockieren — Mögliche Einsparung von 140 ms ▼
- ▲ Nicht sichtbare Bilder aufschieben — Mögliche Einsparung von 470 KiB ▼
- ▲ Reduziere nicht verwendetes JavaScript — Mögliche Einsparung von 91 KiB ▼
- ▲ Reduziere nicht verwendete CSS — Mögliche Einsparung von 75 KiB ▼
- Bildelemente haben keine explizite `width` und `height` ▼
- JavaScript-Ausführungszeit — 0,2 s ▼
- Minimiert den Aufwand für den Hauptthread — 0,6 s ▼
- Drittanbiaternutzung minimieren — Code von Drittanbiatern hat den Hauptthread 20 ms lang blockiert ▼
- Lange Hauptthread-Aufgaben vermeiden — 2 lange Aufgaben gefunden ▼
- Avoid large layout shifts — 1 layout shift found ▼

Yellow Lab Tools

Tested url: <https://wpmeetup-frankfurt.de>



New test



Test again



Dashboard

Global score



96/100



A	jQuery	<div></div> jQuery version	3.7.1
		<div></div> Several jQuery loaded	1
B	CSS complexity	<div></div> Rules count	1831
		<div></div> Complex selectors	342
		<div></div> Colors count	199
		<div></div> Similar colors	25
		<div></div> Breakpoints count	10
		<div></div> Not mobile-first media queries	6
B	Bad CSS	<div></div> CSS syntax error	0
		<div></div> Uses of @import	0
		<div></div> Duplicated selectors	18
		<div></div> Duplicated properties	146
		<div></div> Empty rules	2
		<div></div> Uses of !important	171
		<div></div> Old IE fixes	0
		<div></div> Old prefixes	124
		<div></div> Redundant body selectors	14
		<div></div> Redundant tags selectors	251
A	Web fonts	<div></div> Webfonts number	2
		<div></div> Overweighted webfonts	0 bytes
		<div></div> Unused Unicode ranges	0

WP MEETUP FRANKFURT

Datenbankabfragen

Timings

Protokolle

Request

Blocks

Template: baylys/page.php

Skripte (7)

Stile (11)

Hooks & Actions

Sprachen

HTTP-API-Abrufe

Transient-Aktualisierungen

Berechtigungs-Prüfungen

Arbeitsumgebung

`is_front_page()`

`is_page()`

`is_singular()`

`is_ssl()`

- Dashboard
- Beiträge
- Medien
- Seiten
- Kommentare
- Design
- Plugins
- Benutzer
- Werkzeuge**
- Verfügbare Werkzeuge
- Daten importieren
- Export
- Website-Zustand**
- Personenbezogene Daten exportieren
- Personenbezogene Daten löschen
- Einstellungen
- SEO
- Toolbelt

Website-Zustand

Gut

Status

Bericht

Zustand der Website

Der Test zum Website-Zustand zeigt Informationen zu deiner WordPress-Konfiguration und zu Elementen, die möglicherweise deine Aufmerksamkeit erfordern.

1 empfohlene Verbesserung

Empfohlene Punkte werden als vorteilhaft für deine Website angesehen, auch wenn sie nicht so wichtig sind wie ein kritisches Problem. Sie können Verbesserungen in Bereichen wie Leistung, Benutzerfreundlichkeit und mehr beinhalten.

Du solltest einen persistenten Objekt-Cache verwenden

Leistung



Bestandene Tests ▾

Performancebudget

Beispiel

- Webseite < 1 MB
- Ladezeit < 3 s.
- Antwortzeit Server $< 0,5$ s.
- Performance-Score ≥ 90
- ...

activitypub-remote-reply script enqueues 29 dependencies on all is_singular() requests #705

New issue

Open

kasparsd opened this issue last week · 0 comments · May be fixed by #706

kasparsd commented last week

...

Quick summary

Introduced in #689 the activitypub-remote-reply script handle now enqueues itself with 29 dependencies:

wp-polyfill-inert, regenerator-runtime, wp-polyfill, wp-hooks, wp-i18n, wp-url, wp-api-fetch, react, react-dom, wp-dom-ready, wp-a11y, wp-deprecated, wp-dom, wp-escape-html, wp-element, wp-is-shallow-equal, wp-keycodes, wp-priority-queue, wp-compose, moment, wp-date, wp-html-entities, wp-primitives, wp-private-apis, wp-redux-routine, wp-data, wp-rich-text, wp-warning, wp-components

This happens on posts and pages that (1) don't support comments, (2) don't have comments enabled.

This also adds a combined total of almost 2MB of potentially unused JS to all of those requests.

Steps to reproduce

1. Enable the plugin.

2. Visit any single post or page and confirm that all of the mentioned dependencies are enqueued in the HTML source

Assignees

No one assigned

Labels

Needs triage [Type] Bug

Projects

None yet

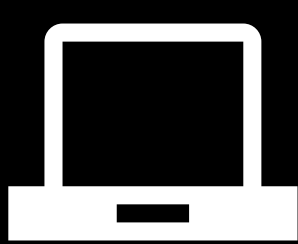
Milestone

No milestone

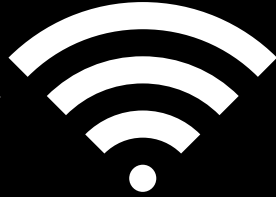
Development

Successfully merging a pull request may close this issue.

Enqueue reply assets only if there is anything...



Client
(Browser)



Netzwerk
(DSL / Kabel / 5G, 4G, ...)



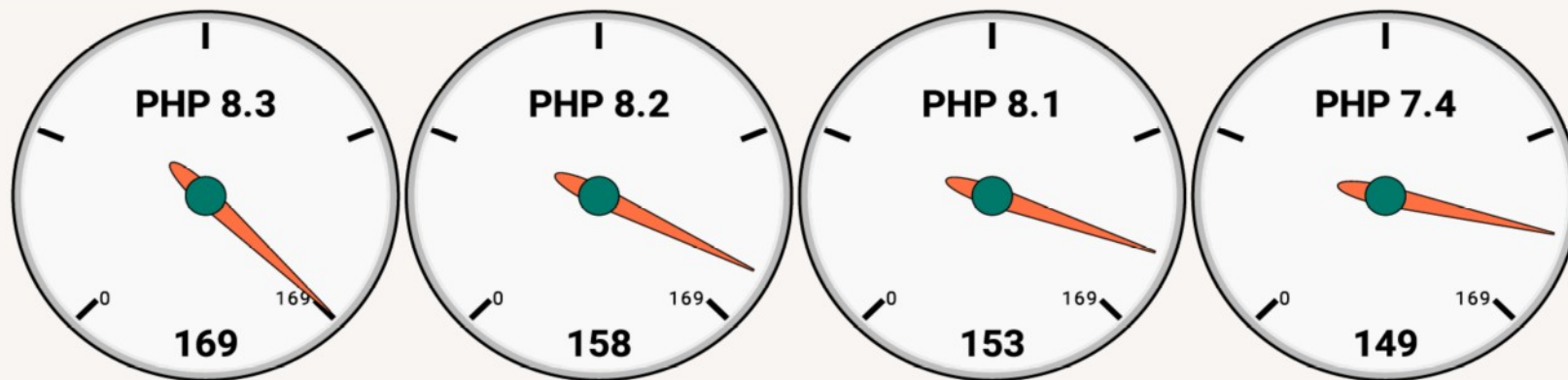
Server
(WordPress)



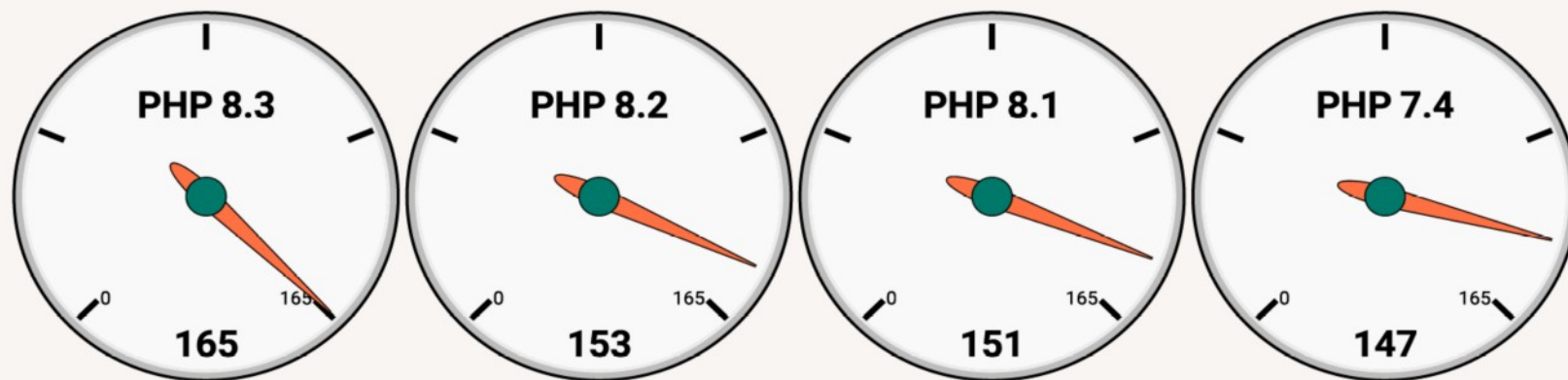
HTML, CSS, JavaScript,
Bilder, Videos

(wird zum Browser gesendet)

PHP + MySQL/MariaDB
(läuft auf'm Server)



WordPress 6.4.2



WordPress 6.2.2

Full Page Caching

Surge

<https://wordpress.org/plugins/surge/>

Hyper Cache

<https://wordpress.org/plugins/hyper-cache/>

WP Super Cache

<https://wordpress.org/plugins/wp-super-cache/>

WP Rocket (Bezahlte)

<https://wp-rocket.me/>

Object Cache



Batcache (Memcached)

<https://wordpress.org/plugins/batcache/>

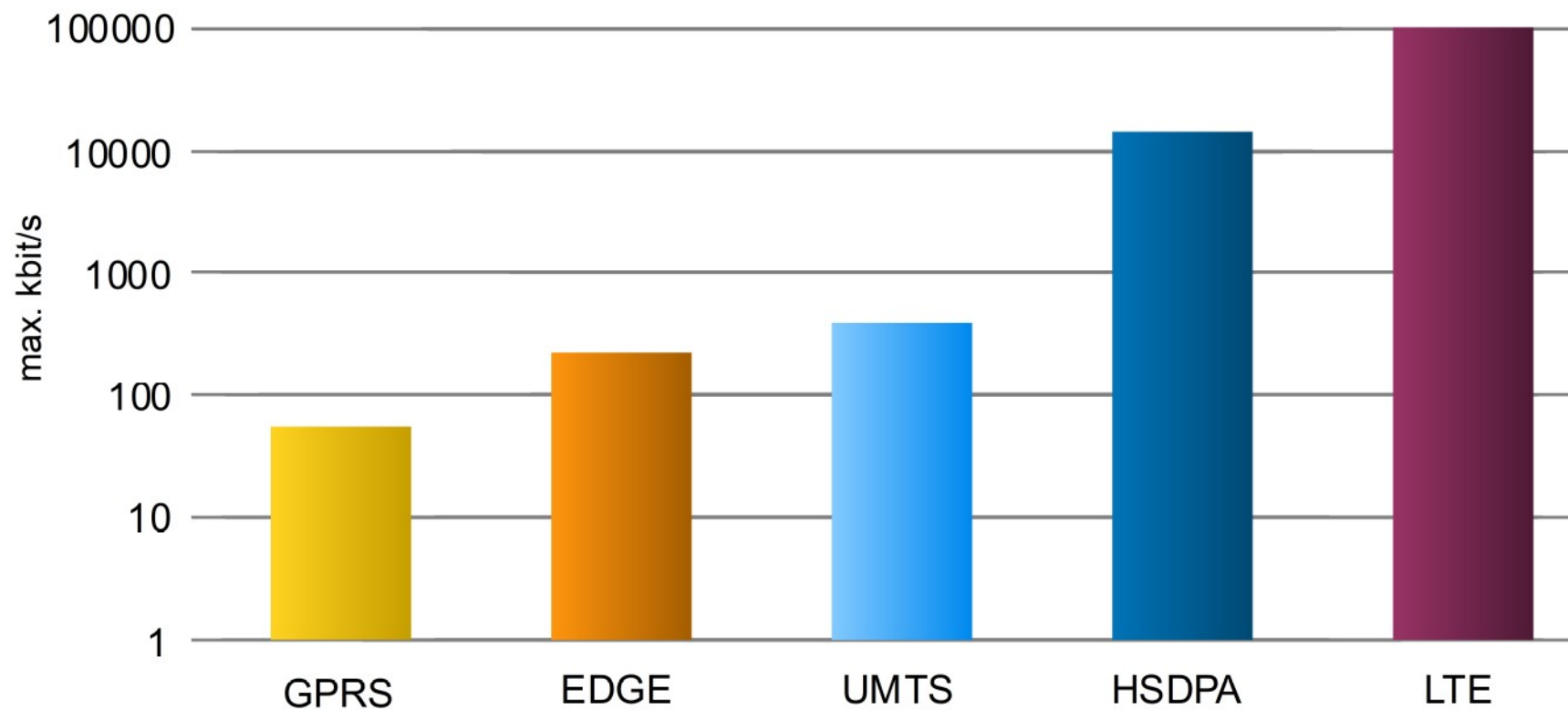
WP Redis (Redis)

<https://wordpress.org/plugins/wp-redis/>



Netzwerk

(DSL / Kabel / 5G, 4G, ...)



Bilder



6000x4000 (original) 11,1 MB

6000x4000 (optimiert) 9,2 MB

1920x1280 1,3 MB

1024x683 0,4 MB

Bildoptimierer

ImageOptim (Mac)

<https://imageoptim.com/mac>

FileOptimizer (Windows)

<https://sourceforge.net/projects/nikkhokkho/>

Trimage / Curtail (Linux)

<https://trimage.org/>

<https://github.com/Hu1uti/Curtail>

Bild-Plugins

EWWW (Lokal / Webdienst)

<https://wordpress.org/plugins/ewww-image-optimizer/>

Imagify (Webdienst)

<https://wordpress.org/plugins/imagify/>

GIF



GIF = 4,3 MB

MP4 = 143 kB

```
ffmpeg -i video.gif -c:v libx264 -an -movflags faststart \  
-pix_fmt yuv420p -s 544x292 video.mp4
```

<https://www.washingtonpost.com/news/the-switch/wp/2015/08/06/tesla-unveils-weird-new-car-charging-robo-snake/>

CSS/JS Kombinieren + Minifizieren

Autoptimize

<https://wordpress.org/plugins/autoptimize/>

WP Rocket (Bezahlte)

<https://wp-rocket.me>

<https://gitlab.com/walterebert/wordpress-project/-/blob/master/src/web/.htaccess>

.htaccess

- **Komprimieren**
- **Cachen**

.htaccess

```
AddOutputFilterByType DEFLATE \  
    application/atom+xml \  
    application/javascript \  
    application/json \  
    application/xml \  
    image/svg+xml \  
    text/css \  
    text/html \  
    text/javascript \  
    text/plain \  
    text/vtt \  
    text/xml \  
    ...
```

.htaccess

ExpiresActive on

ExpiresDefault "access plus 1 week"

ExpiresByType application/json "access plus 0 seconds"

ExpiresByType application/ld+json "access plus 0 seconds"

ExpiresByType application/schema+json "access plus 0 seconds"

ExpiresByType application/vnd.geo+json "access plus 0 seconds"

ExpiresByType application/xml "access plus 0 seconds"

ExpiresByType text/calendar "access plus 0 seconds"

ExpiresByType text/html "access plus 0 seconds"

ExpiresByType text/xml "access plus 0 seconds"

ExpiresByType application/rss+xml "access plus 1 hour"

ExpiresByType application/atom+xml "access plus 1 hour"

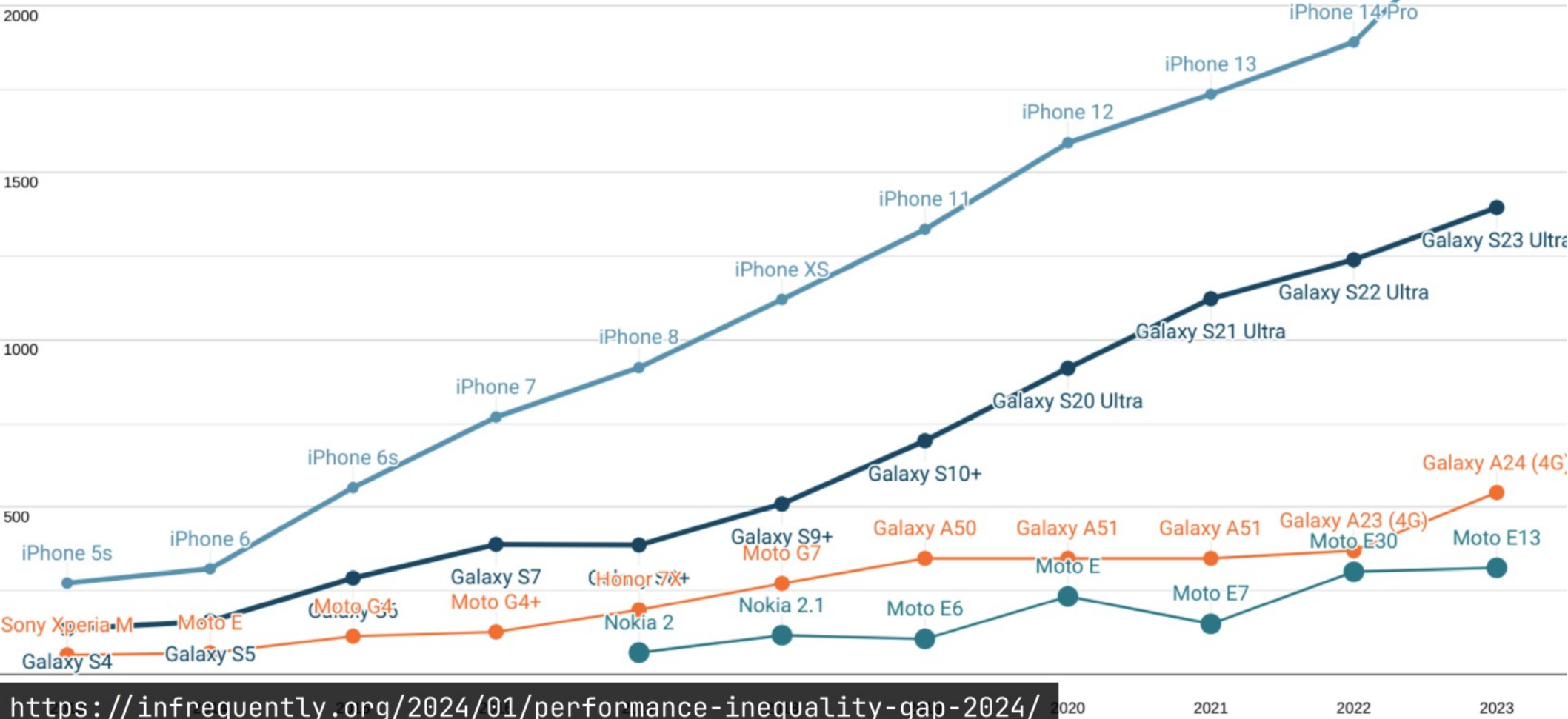


Client
(Browser)

On the **high-end** iPhone 8 it takes just **~4s** to parse/compile CNN's JS compared to **~13s** for an **average phone** (Moto G4)

Geekbench 5 Single-Core Scores

● iOS Single-Core ● Android Single-Core ● Budget Android Single-Core ● Low-End Single-Core





Fragen?

Walter Ebert

`wee.press`

`profiles.wordpress.org/walterebert`

`@walterebert@social.walterebert.com`