

Performance 2024



Content team

Hello content team! This is your personal doc to collaborate on and plan the contents of your chapter. [Click Request edit access above to get started.](#)

Please add your name and email address below so we can @tag each other in the comments. You can also subscribe to all comments by opening the comment history, clicking the notification bell, and selecting All.

Authors: Eugenia Zigisova, Ines Akrap

Reviewers: ...

Analysts: ...

The objective of your chapter is to write a data-driven answer to this big question:

“What is the state of Web Performance in 2024?”

Learn more about the [chapter lifecycle](#) and refer to your chapter’s [tracking issue](#) on GitHub for more info. Thank you all for your contributions! [Official call](#)

Outline

The purpose of this section is to define the scope of the chapter by creating an ordered list of all of the topics to be explored. You can think of this outline as the chapter's table of contents. This list will become your narrative, so consider how the content should be sequenced and how much additional depth is needed for major topics. You may choose to start with last year's outline and add or remove content as needed. Every chapter must have an introduction and conclusion, but everything in between is up to you.

Every chapter must also be data-driven, so for each topic in the outline below, clearly enumerate which metrics you'll need to substantiate your narrative. Work with your analysts to clarify what data is needed and how the results should be formatted. For example, if you're measuring the usage of a particular HTTP header value, you can measure it as the percentage of pages having that header, as the percentage of headers having that value, as a distribution of values, what the largest value is, etc. Clarify those expectations upfront so that the analysts know how to write the corresponding queries and whether the metrics are even feasible in the dataset.

First meeting to outline the chapter contents by May 1 

Custom metrics completed by June 1 

HTTP Archive crawl by June 1 

Querying all metrics and saving the results by August 15 

First draft of chapter by September 15

Reviewing & Editing of chapter by October 10

Publication of chapter (Markdown & PR) by October 15

Reviewer's Key

 - Ready for initial review

 - Reviewer approval, ready for final draft

Content team	1
Outline	2
Metrics	3
Draft	5
Introduction.....	5
Web Performance Inequality Issue.....	5
Web Vitals.....	6
Core Web Vitals.....	6
Loading.....	6
Time to First Byte (TTFB).....	6
First Contentful Paint (FCP).....	6
Largest Contentful Paint (LCP).....	6
Loading Conclusion.....	7
Interactivity.....	7
Total Blocking Time (TBT).....	7
Time to Interactive (TTI).....	8
FID deprecation.....	8
Interaction to Next Paint (INP).....	8
Interactivity Conclusion.....	8
Visual Stability.....	8
Cumulative Layout Shift (CLS).....	8
Visual Stability Conclusion.....	8
Conclusion.....	8
Brainstorming.....	9

Metrics

Custom metrics completed by June 1

HTTP Archive crawl by June 1

Querying all metrics and saving the results by August 15

Draft

Reviewing & Editing of chapter by October 1

Publication of chapter (Markdown & PR) by October 15

Introduction

No one ever complained about a fast website, but a slow-loading and sluggish website quickly frustrates users. Website speed and overall performance directly impact user experience and the success of a website. Moreover, if a website is slow, it becomes less accessible to users, which is against the fundamental goal of the web — to provide universal access to the universe of information.

In recent years, Core Web Vitals performance metrics have improved, showing positive trends across many performance metrics. However, some inconsistencies can be observed. For example, the gap between high-end and low-end devices is widening, especially in mobile web performance, as highlighted in Alex Russell's research in [The Performance Inequality Gap](#). Web performance is tied to what devices and networks people can afford. Fortunately, more developers are aware of these challenges and are actively working to improve performance.

In the performance chapter, we focus on Core Web Vitals, as they are key user-centric metrics for assessing web performance. However, we also analyze the web performance from a broader perspective: loading, interactivity, and visual stability, adding supportive metrics like First Contentful Paint. This allows us to explore other performance and user experience-related metrics to get a more comprehensive picture of how websites performed in 2024.

What's new this year?

- [Interaction to Next Paint \(INP\)](#) has officially replaced [First Input Delay \(FID\)](#) as part of Core Web Vitals. INP helps to evaluate overall interactivity performance more accurately.
- [Long Animation Frames \(LoAF\)](#) data is available for the first time, providing new insights into the reasons for poor INP.
- As of this year, the Performance chapter also includes an analysis of the data for secondary pages in addition to home pages. This allows us to compare the home page with the secondary page performance.

Notes on Data Sources

The HTTPArchive contains only lab performance data. In other words, it is data from a single website load event. This is useful but limited if we want to understand how users experience performance.

Thus, in addition to the data HTTP Archive, most of this report is based on real user data from the [Chrome User Experience Report \(CrUX\)](#). Note that while Chrome is the most widely used browser worldwide, it doesn't reflect performance across all browsers and all regions of the world.

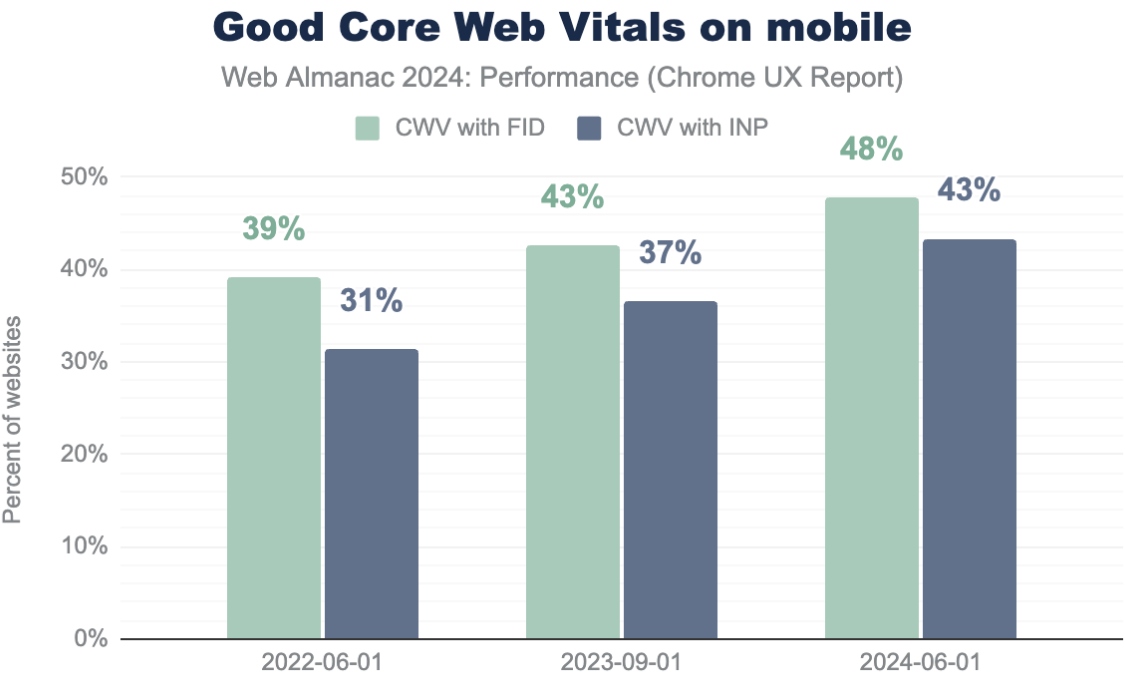
CrUX is a great source of data, but it doesn't contain certain metrics like LCP and INP sub-parts, as well as Long Animation Frames. Luckily, the performance monitoring platform [RUMvision](#) has provided us with this data for the period from 1st January to 6th October

2024. Compared to HTTP archive, RUMvision tests a smaller amount of websites, which is why the results for the same metrics might be slightly different.

Core Web Vitals

Core Web Vitals (CWV) are user-centric metrics designed to measure the different aspects of web performance. These include the Largest Contentful Paint (LCP), which tracks loading performance, Interaction to Next Paint (INP), which measures interactivity, and Cumulative Layout Shift (CLS), which assesses visual stability.

Starting this year, INP has officially replaced First Input Delay (FID) and became a part of the CWV. While INP measures the full delay of all interactions experienced by a user, FID only focuses on the input delay of the first interaction. This wider scope makes INP a better reflection of the full user experience.



```
{{ figure_markup(
  image="good-core-web-vitals-fid-devices-years.png",
  caption: "The percent of websites having good CWV using FID and INP, segmented by year."
  description="Bar chart showing the percentage of websites with good Core Web Vitals (CWV) on mobile, comparing CWV with FID (First Input Delay) and CWV with INP (Interaction to Next Paint) over time. In 2022, 39% of websites had good CWV with FID, while 31% had good CWV with INP. In 2023, the percentage increased to 43% for CWV with FID and 37% for CWV with INP. In 2024, 48% of websites had good CWV with FID, and 43% had good CWV with INP.",
```

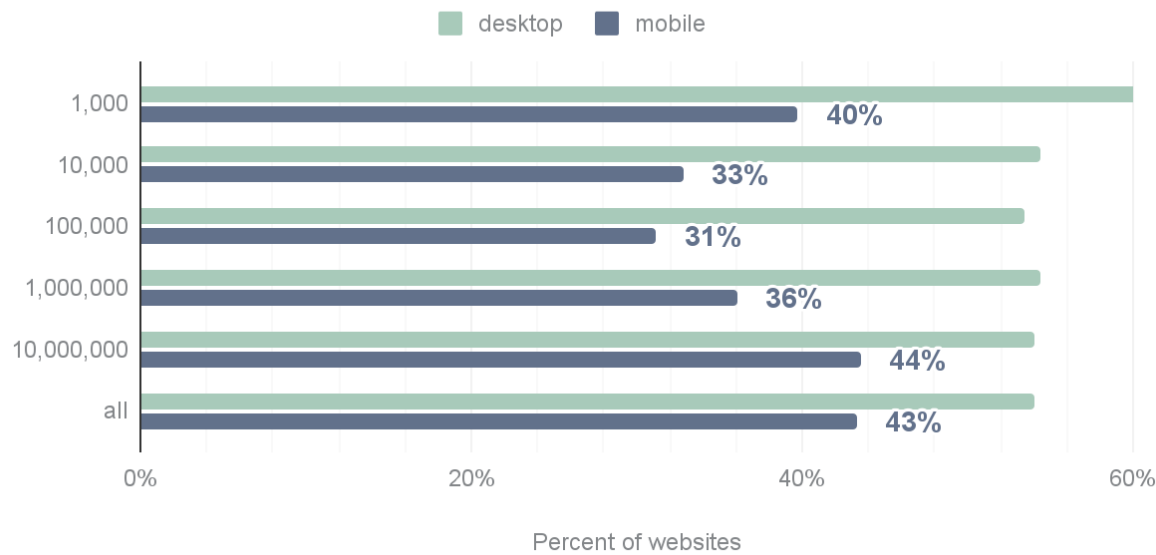
```
chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ahQDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=1908072353&format=interactive",
sheets_gid="1535582002",
sql_file="web_vitals_by_device.sql"
)
}}
```

The replacement of the FID with the INP metric significantly impacted the percentage of websites with good CWV on mobile. This doesn't mean the user experience has worsened, just that is now reflected more accurately due to the metric update. If we still used FID as a measure of interactivity, 48% of the websites would have good CWV on mobile devices. However, with the INP metric, this figure drops to 43%. Interestingly, performance on desktop devices stays the same regardless of which responsiveness metric we use, i.e., 54%.

In the period from 2020 to 2022, we saw that mobile web performance measured by CWV with FID was improving faster than desktop one, and the gap between them was closing, reaching just 5% in 2022. As CWV with INP chart shows, in 2024, the websites on the desktop performed 11% better than on mobile, so the introduction of the INP shows that the gap is much bigger.

Websites with good CWV by rank

Web Almanac 2024: Performance (Chrome UX Report)



```
{{ figure_markup(
  image="XXX.png",
  caption: "The percent of websites having good CWV, segmented by rank and desktop vs mobile."
}}
```

description="Bar chart showing the percentage of websites with good CWV (Core Web Vitals) performance by rank for desktop and mobile. For the top 1,000 websites, 40% of mobile websites have good CWV, compared to 54% of desktop websites. In the top 10,000, 33% of mobile websites have good CWV, while 46% of desktop websites do. In the top 100,000, 31% of mobile websites and 43% of desktop websites have good CWV. In the top 1,000,000, 36% of mobile websites have good CWV, compared to 48% of desktop websites. For websites ranked 10,000,000 and beyond, 44% of mobile websites and 43% of desktop websites achieve good CWV.",

chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ahQDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=1814767865&format=interactive",

sheets_gid="355582610",

sql_file="web_vitals_by_rank_and_device.sql"

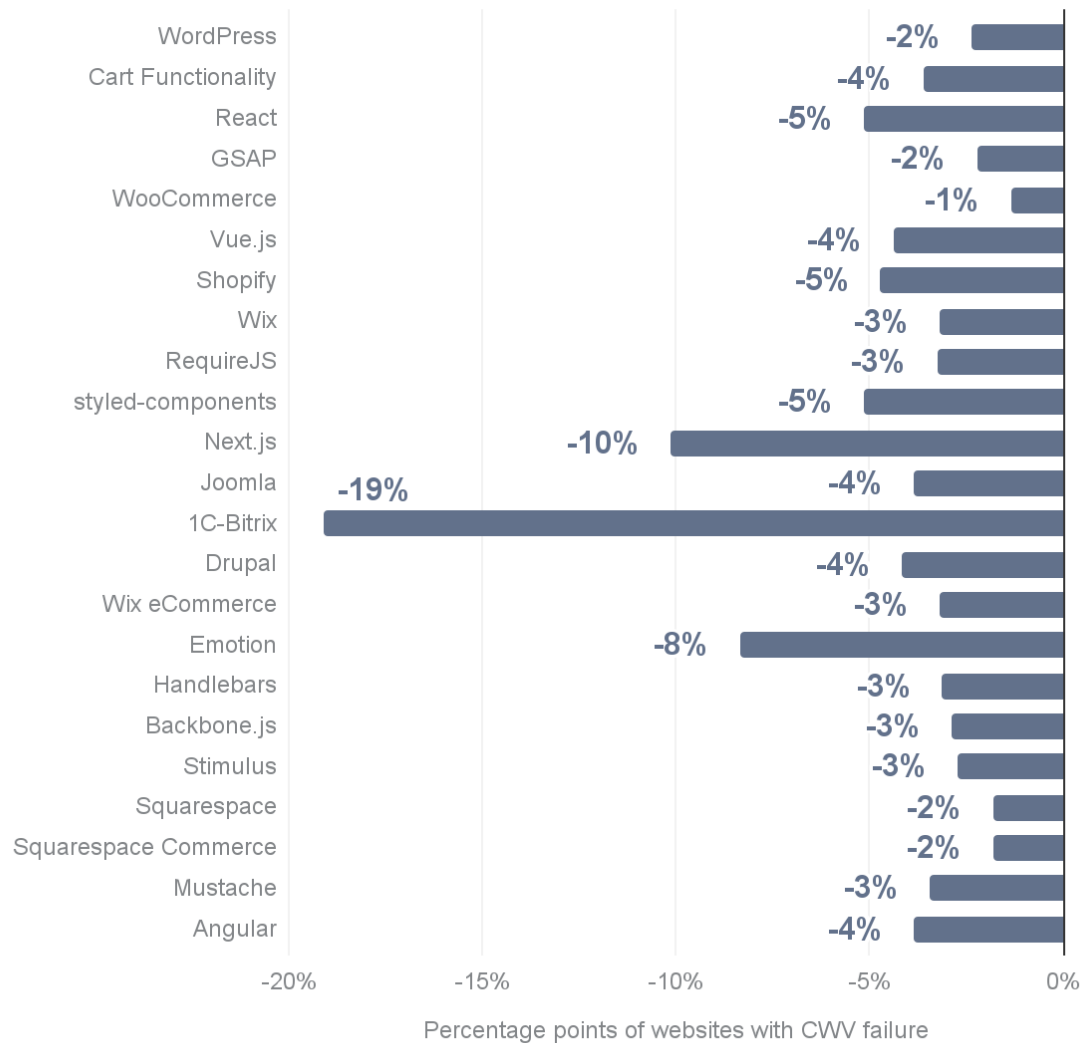
)

}}

CWV with INP shows a new tendency when analyzing websites by rank. Previously, the most popular websites [tended to have the best CWV experience](#), however, this year's statistics shows the opposite: 40% of 1000 most popular websites on mobile have good CWV which is lower than total website CWV of 43%.

New mobile CWV failures due to INP

Web Almanac 2024: Performance (Chrome UX Report)



Bar chart showing the percentage points of websites with new mobile CWV failures due to INP across various platforms and technologies. 1C-Bitrix has the highest percentage of new failures at -19%, followed by Next.js at -10%, and Emotion at -8%. Other platforms such as WordPress, React, Vue.js, and Drupal show smaller decreases, ranging from -2% to -5%. The chart also displays a range of smaller decreases for various technologies, including Handlebars, Backbone.js, Squarespace, and Angular, all seeing decreases of around -2% to -5%.

```

chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=655
066315&format=interactive",
sheets_gid="869409419",
sql_file="web_vitals_by_technology.sql"
)
}}

```

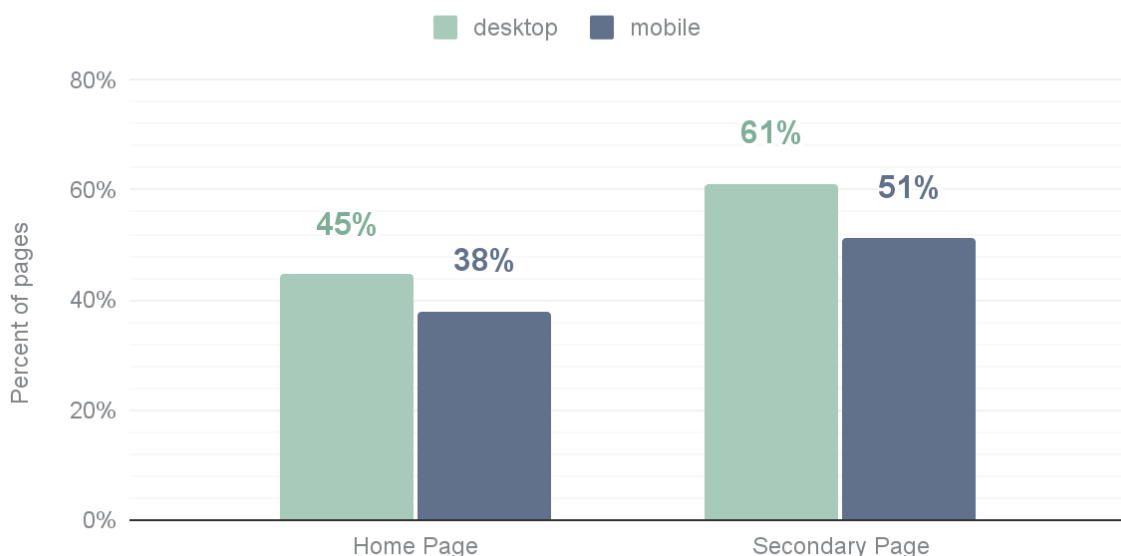
As mentioned earlier, the CWV scores have decreased due to the switch of the INP metric. We investigated how different technologies have been affected by this shift. The diagram above illustrates the percent point drop in the percentage of websites with good CWV across various technologies after the INP was introduced.

Several technologies were significantly impacted, including a 19% drop for 1C-Bitrix (a popular CMS in Central Asia), a 10% drop for Next.js (a React-based framework), and an 8% drop for Emotion (a CSS-in-JS tool). We can't be entirely certain that the decline in CWV scores is solely due to the technology used. Next.js has server-side rendering (SSR) and static site generation (SSG) features, which should theoretically enhance INP, but it has still seen a significant decline. As Next.js is based on React, many websites rely on client-side rendering, which can negatively impact INP. This could serve as a reminder for developers to leverage the SSR and SSG capabilities of the framework they use.

As of this year, secondary pages are available to compare with homepage data.

Good CWV for home pages and secondary pages

Web Almanac 2024: Performance (Chrome UX Report)



```

{{ figure_markup(
  image="XXX.png",
  caption: "The percent of websites having good CWV, segmented by page type"

```

```
description="Bar chart showing the percentage of pages with good CWV (Core Web Vitals)
for home pages and secondary pages on desktop and mobile. For home pages, 45% of
desktop pages have good CWV, while 38% of mobile pages achieve good CWV. For
secondary pages, 61% of desktop pages have good CWV, compared to 51% of mobile
pages.",
chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx\_5nC9ahQDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=1034225442&format=interactive",
sheets_gid="1159394005",
sql_file="web_vitals_by_device_secondary_pages.sql"
)
}}
```

Secondary pages demonstrate significantly better CWV results than home pages. The percentage of the desktop secondary pages with good CWV is by 14 percentage points better than for home pages. For mobile websites, the difference is 13 percentage points. By looking at CWV data only, it is hard to identify what kind of performance experience is better. We will explore these aspects—layout shift, loading, and interactivity—in the corresponding sections.

Loading Speed

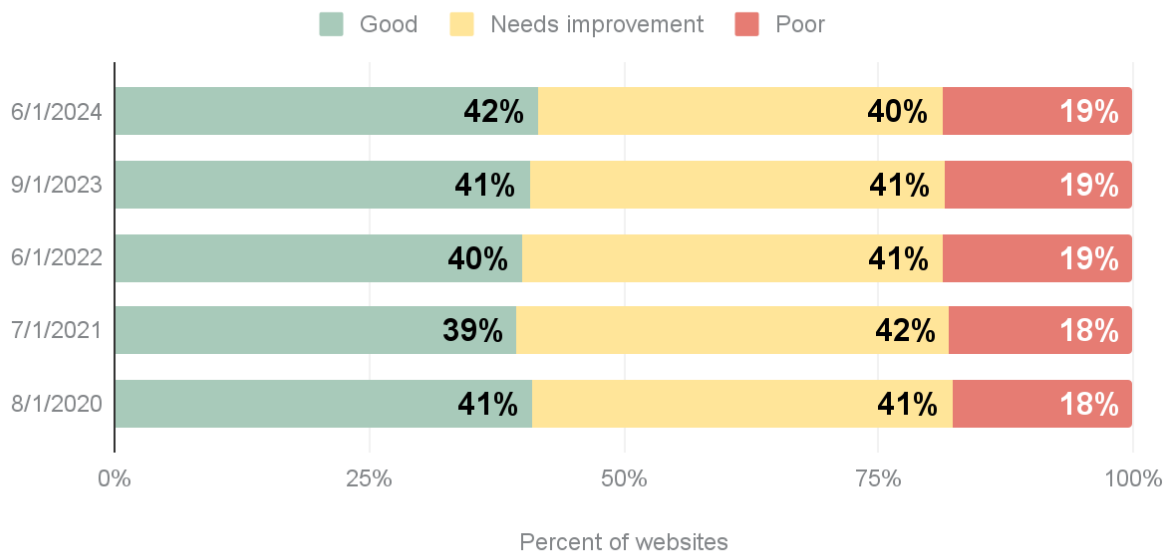
People often refer to website loading speed as a single metric, but in fact, the loading experience is a multi-stage process. No single metric fully captures it. Every stage has an impact on the speed of a website.

Time to First Byte (TTFB)

[Time to First Byte](#) (TTFB) measures the time from when a user initiates loading a page until the browser receives the first byte of the response. It includes phases like redirect time, DNS lookup, connection and TLS negotiation, and request processing. Reducing latency in connection and server response time can improve TTFB. 800ms is considered the threshold for good TTFB.

TTFB performance for mobile websites

Web Almanac 2024: Performance (Chrome UX Report)

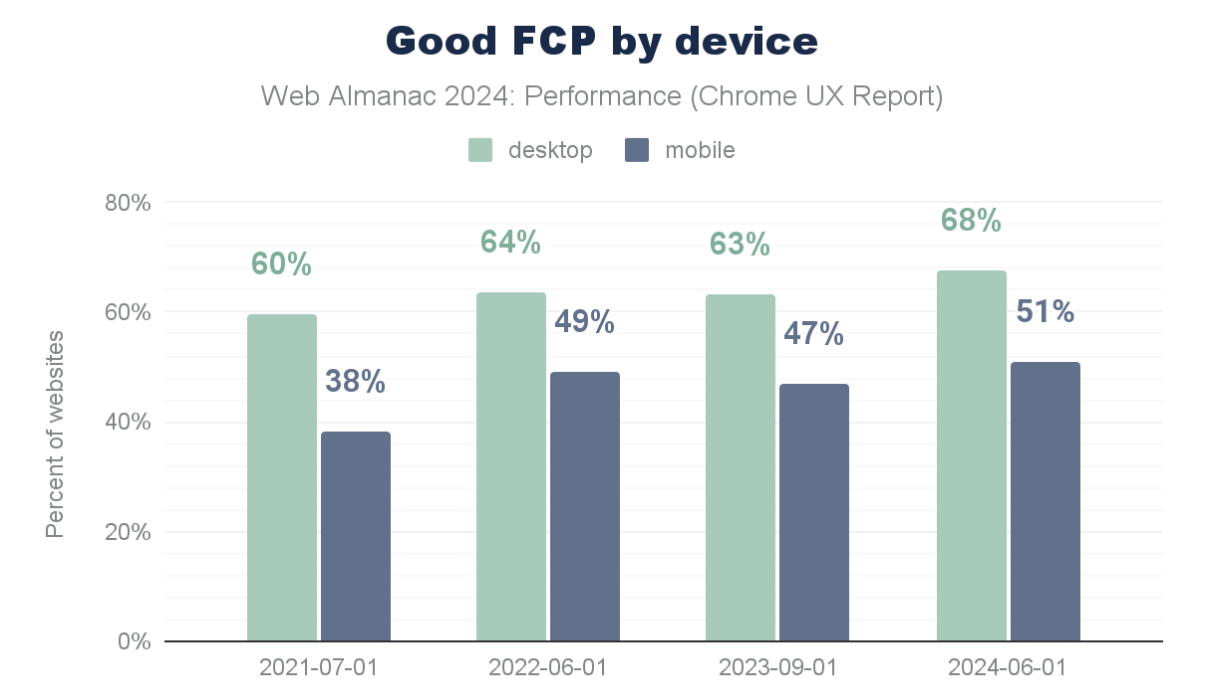


```
{{ figure_markup(  
  image="XXX.png",  
  caption: "The percent of websites having good TTFB, segmented by device and year."  
  description="Stacked bar chart showing TTFB (Time to First Byte) performance for mobile  
websites from 2020 to 2024, categorized as good, needs improvement, and poor. In 2024,  
42% of mobile websites had good TTFB, 40% needed improvement, and 19% were poor. In  
2023, 41% were good, 41% needed improvement, and 19% were poor. In 2022, 40% of  
websites had good TTFB, 41% needed improvement, and 19% were poor. In 2021, 39%  
were good, 42% needed improvement, and 18% were poor. In 2020, 41% of mobile websites  
had good TTFB, 41% needed improvement, and 18% were poor.",  
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah  
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=192  
5312055&format=interactive",  
  sheets_gid="1535582002",  
  sql_file="web_vitals_by_device.sql"  
)  
}}
```

Over the past five years, the percentage of mobile web pages with good TTFB has remained stable, from 41% in 2021 to 42% in 2024. The percentage of pages that need TTFB improvements has decreased by 1%, and unfortunately, the percentage of pages with poor TTFB remains the same. Since this metric has not changed significantly, we can conclude that there have been no major improvements in connection speed or backend latency.

First Contentful Paint (FCP)

First Contentful Paint (FCP) is a performance metric that helps indicate how quickly users can start seeing content. It measures the time from when a user first requests a page until the first piece of content is rendered on the screen. A good FCP should be under 1.8 seconds.



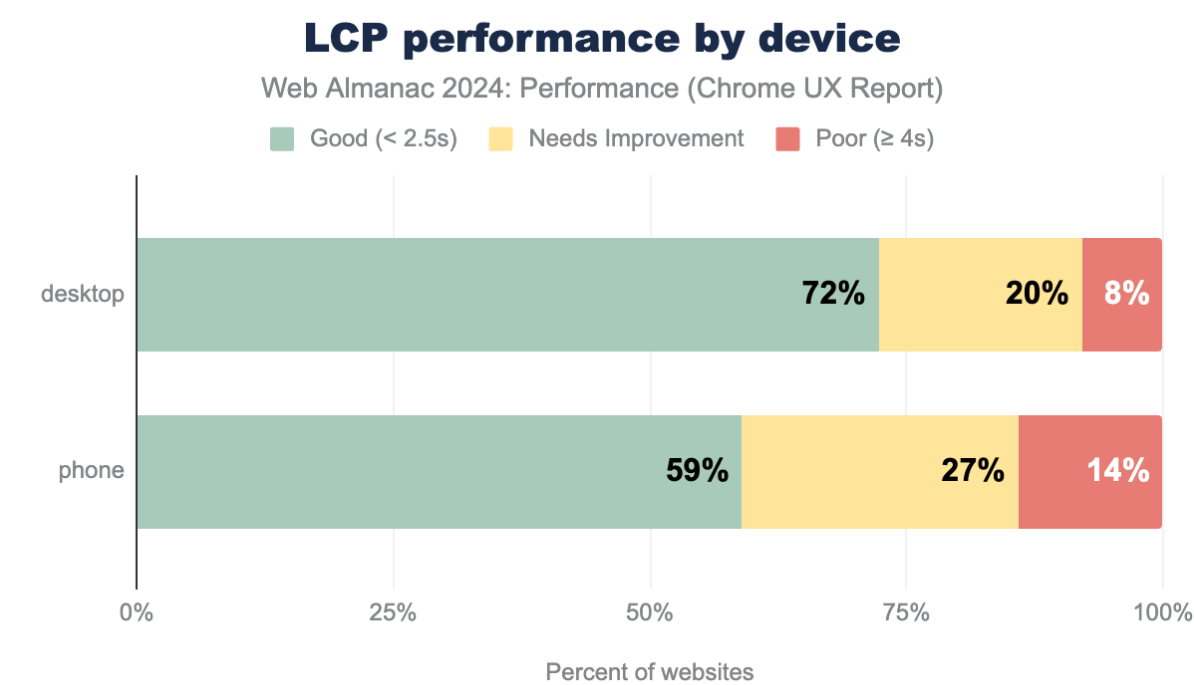
```
{{ figure_markup(  
  image="XXX.png",  
  caption: "The percent of websites having good FCP, segmented by device and year."  
  description="Bar chart showing the percentage of websites with good FCP (First Contentful  
Paint) performance by device over time. In July 2021, 60% of desktop websites had good  
FCP, compared to 38% of mobile websites. By June 2022, 64% of desktop websites and  
49% of mobile websites had good FCP. In September 2023, 63% of desktop websites had  
good FCP, while 47% of mobile websites did. By June 2024, the percentage increased to  
68% for desktop websites and 51% for mobile websites.",  
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah  
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=105  
8680176&format=interactive",  
  sheets_gid="1535582002",  
  sql_file="web_vitals_by_device.sql"  
)  
}}
```

FCP has shown improvements over the past few years. Although there was a slight decline in 2023, the metric recovered in 2024, reaching 68% for desktop and 51% for mobile websites. Overall, this reflects a positive trend in how fast the first content is loaded. Taking into account that the TTFB metric remained mostly unchanged, FCP improvements might be driven by client-side rendering rather than server-side optimizations.

Interestingly, website performance is not the only factor that influences FCP. In the research "[How Do Chrome Extensions Impact Browser Performance?](#)" Matt Zeunert found that browser extensions can significantly affect page loading times. Many extensions start running their code as soon as a page starts loading, delaying the first contentful paint. For instance, some extensions can increase FCP from 100ms to 250ms.

Largest Contentful Paint (LCP)

LCP is an important metric as it indicates how quickly the largest element in the viewport is loaded. A best practice is to ensure the LCP resource starts loading as early as possible. A good LCP should be under 2.5 seconds.



```
{{ figure_markup(  
  image="XXX.png",  
  caption: "The percent of websites having good, need improvements and poor LCP,  
segmented by device"  
  description="Stacked bar chart showing LCP performance by device, categorized as good  
(under 2.5 seconds), needs improvement (2.5–4 seconds), and poor (over 4 seconds). For  
desktop, 72% of websites have good LCP, 20% need improvement, and 8% perform poorly.  
For phones, 59% of websites have good LCP, 27% need improvement, and 14% perform  
poorly.",
```

```

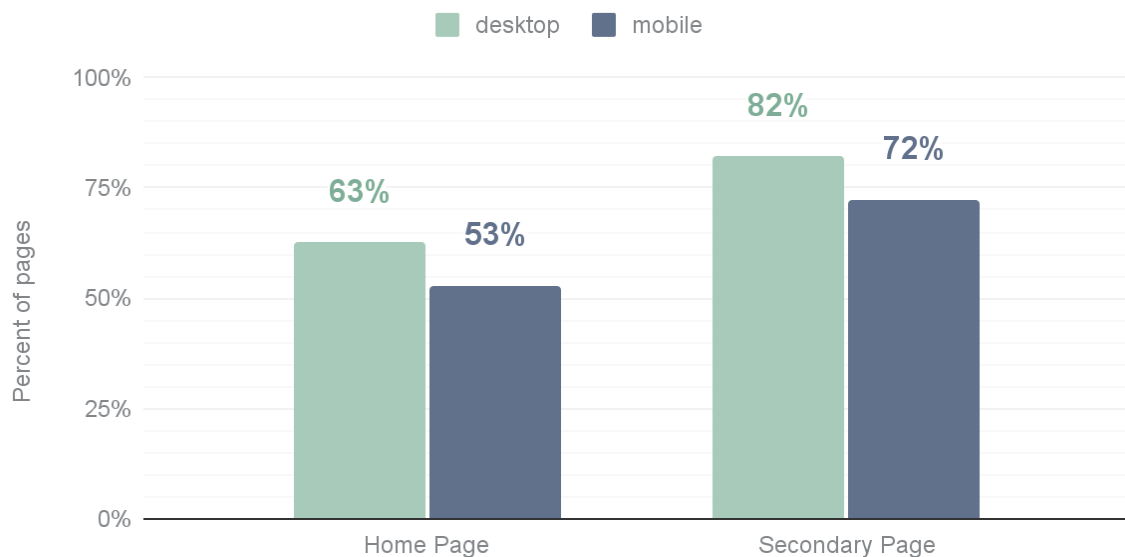
chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx\_5nC9ahQDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=2074458485&format=interactive",
sheets_gid="1535582002",
sql_file="web_vitals_by_device.sql"
)
}}

```

LCP has also improved in recent years (from 44% of pages with good LCP in 2022 to 54% in 2024) following the overall positive tendency in CWV. In 2024, 59% of mobile pages achieved a good LCP score. However, there is still a significant gap compared to desktop sites, where 74% have good LCP. This firmly established trend is explained by differences in device processing power and network quality. However, it also highlights that many web pages are still not optimized for mobile use.

Good LCP for home pages and secondary pages

Web Almanac 2024: Performance (Chrome UX Report)



```

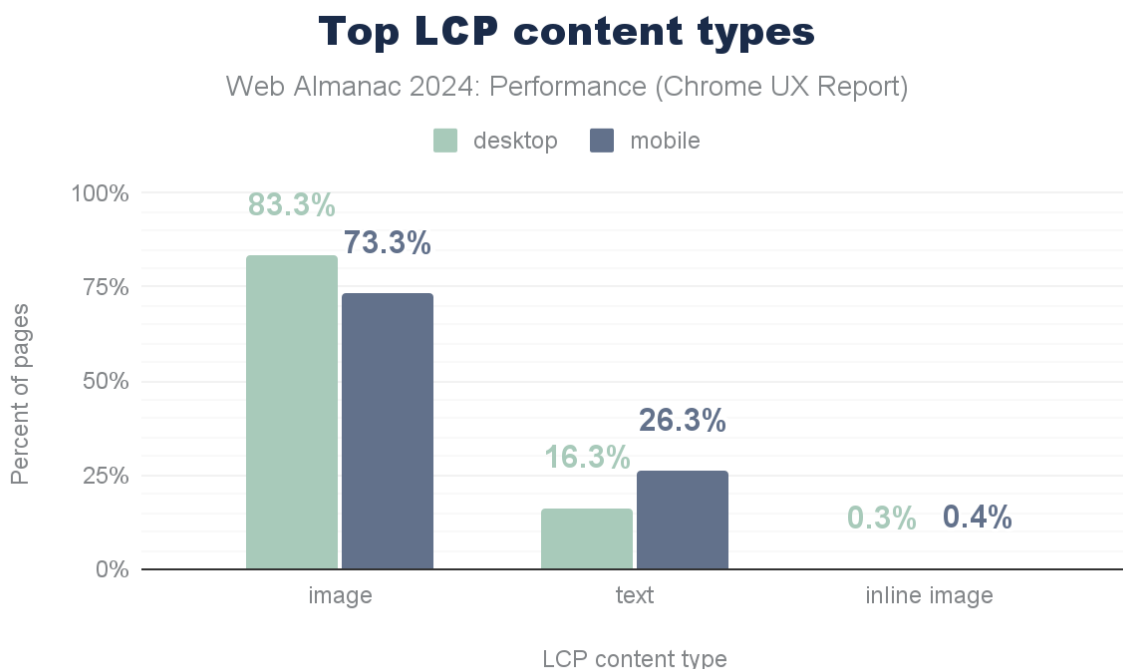
{{ figure_markup(
  image="XXX.png",
  caption: "The percent of websites having good LCP, segmented by device and page type."
  description="Bar chart showing the percentage of pages with good LCP for home pages and secondary pages on desktop and mobile. For home pages, 63% of desktop pages have good LCP, while 53% of mobile pages achieve the same. For secondary pages, 82% of desktop pages have good LCP, compared to 72% of mobile pages."
),
chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx\_5nC9ahQDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=1404370023&format=interactive",
sheets_gid="1159394005",

```

```
sql_file="web_vitals_by_device_secondary_pages.sql"
)
}}
```

The comparison between home pages and secondary pages reveals an interesting trend: 72% of all secondary pages have good LCP, which is 20% higher than the result for home pages. This is likely because users typically navigate on the home page first, causing the initial load to happen on the home page. After they navigate to secondary pages, many of the resources are already loaded and cached, speeding up the LCP element to render. Another possible reason is that the home page often contain more media-rich content such as video and images, compared to secondary pages.

LCP content types



```
{{ figure_markup(
  image="largest-contentful-paint-top-content-types.png",
  caption: "Top three LCP content types segmented by device."
  description="Bar chart showing the top LCP content types for desktop and mobile in 2024.
  For desktop, 83.3% of pages have images as the LCP content type, while 73.3% of mobile
  pages have images as their LCP content. Text accounts for 16.3% of LCP content on
  desktop and 26.3% on mobile. Inline images are rare, making up 0.3% of LCP content on
  desktop and 0.4% on mobile.",
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah
  QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=113
  4330296&format=interactive",
  sheets_gid="1760287339",
```

```
sql_file="lcp_resource_type.sql"  
)  
}}
```

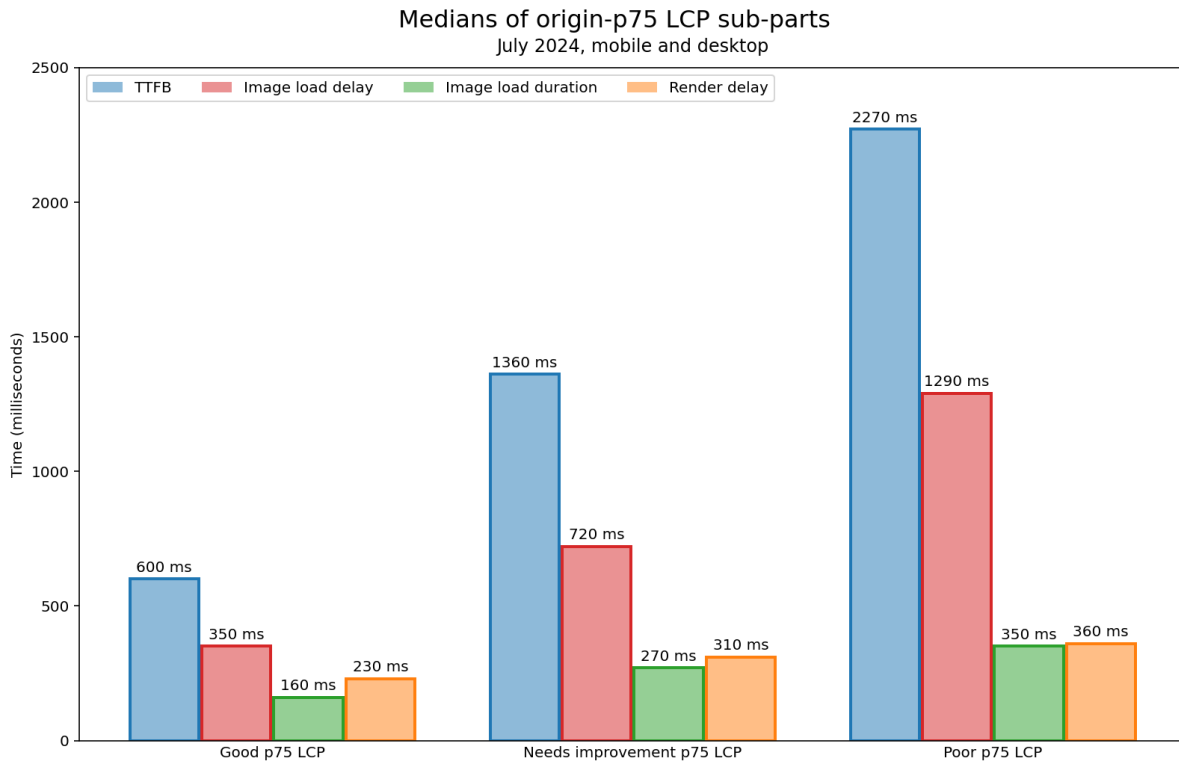
Most LCP elements, or 73% of mobile pages, are images. Interestingly, this percentage is 10% higher on desktop pages. The situation is reversed for text content. Compared to desktop, 10% more mobile webpages use text as their LCP element. This difference is likely because desktop websites can accommodate more visual content due to larger viewport sizes and generally higher performance.

LCP sub-parts

Several stages of processing must occur before the LCP element can be fully rendered:

- **Time to First Byte (TTFB)** — the time it takes the server to begin responding to the initial request.
- **Resource Load Delay**, which is how long after TTFB the browser begins loading the LCP resource. The LCP elements that originate as inline resources, such as text-based elements or inline images (data URIs), will have a 0 ms load delay. Those that require another asset to be downloaded, like an external image, might experience a load delay.
- **Resource Load Duration** which measures how long it takes to load the LCP resource; this stage is also 0 ms if no resource is needed.
- **Element Render Delay** which is the time between when the resource finished loading and the LCP element finished rendering.

In the article "[Common Misconceptions About How to Optimize LCP](#)," Brendan Kenny analyzed a breakdown of LCP sub-parts using recent CruX data.



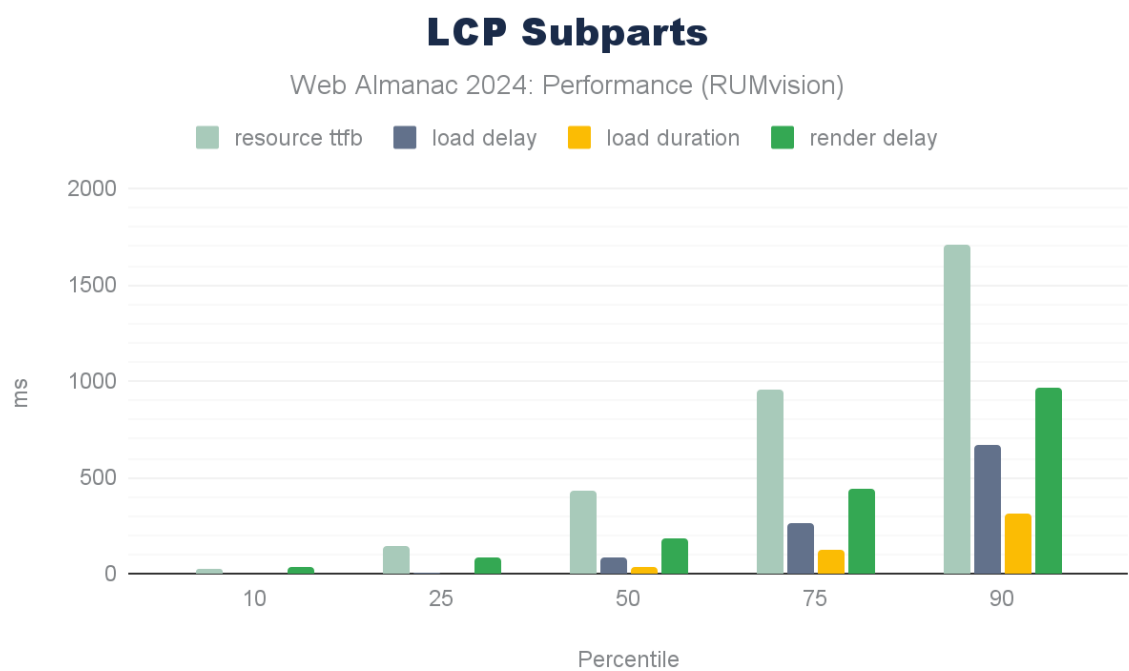
```
{{ figure_markup(
  image="XXX.png",
  caption: "Time spent in each LCP subpart, grouped into LCP buckets of good, needs
improvement, and poor."
  description="Bar chart showing the medians of origin p75 LCP sub-parts for good, needs
improvement, and poor p75 LCP in July 2024, across mobile and desktop. For good p75
LCP, TTFB is 600 ms, image load delay is 350 ms, image load duration is 160 ms, and
render delay is 230 ms. For needs improvement p75 LCP, TTFB is 1360 ms, image load
delay is 720 ms, image load duration is 270 ms, and render delay is 310 ms. For poor p75
LCP, TTFB is 2270 ms, image load delay is 1290 ms, image load duration is 350 ms, and
render delay is 360 ms.",
  chart_url="https://web.dev/static/blog/common-misconceptions-lcp/images/median-subpart-p
75s.png",
)
}}
```

The study showed that image load duration has the least impact on LCP time, taking only 350 ms at the 75th percentile for websites with poor LCP. Although resource load duration optimization techniques like image size reduction are often recommended, they don't offer as much time savings as other LCP sub-parts, even for sites with poor LCP.

TTFB is the largest part among all LCP sub-parts due to the network requests for external resources. Websites with poor LCP spend 2.27 seconds on TTFB alone, which is almost as long as the threshold for a good LCP (2.5 seconds). As we saw in the TTFB section, there hasn't been much improvement in the percentage of websites with good TTFB, indicating that this metric offers significant opportunities for LCP optimization.

Surprisingly, websites spend more time on resource load delay than on load duration, regardless of their LCP status. This makes load delay a good candidate for optimization efforts. One way to improve load delay is by ensuring that the LCP element starts loading as early as possible, which will be explored in detail in the section on LCP statical discoverability.

This year, we analyzed LCP sub-part data from another source: RUMvision. Although RUMvision has a different population of websites, it's interesting to compare it with the larger CruX website population. We assume that websites using performance monitoring tools like RUMvision should have more insights into performance optimization opportunities than the average website represented in CruX. Naturally, the LCP sub-part results from two different datasets show some differences.



```
{{ figure_markup(  
  image="",  
  caption="Time spent in each LCP subpart by percentile ",  
  description="Bar chart showing the distribution of LCP subparts in milliseconds (ms) across  
different percentiles. At the 10th percentile, all subparts have minimal values. At the 25th  
percentile, resource TTFB and load delay remain under 100 ms. At the 50th percentile,  
resource TTFB increases to around 200 ms, with small increases in load delay, load  
duration, and render delay. At the 75th percentile, resource TTFB exceeds 500 ms, and  
render delay also shows an increase. At the 90th percentile, resource TTFB is over 1500  
ms, and render delay rises to over 600 ms, with load delay and load duration also  
increasing.",
```

```
chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=249
678580&format=interactive",
  sheets_gid="1987931132"
)
}}
```

According to RUMvision data, TTFB is also the largest contributor to the LCP time in comparison to the other LCP sub-parts. However, the results of other sub-parts vary. Render delay is the second largest contributor to LCP, taking 184 ms. At the 75th percentile, the render delay grows to 443 ms. This reflects a tendency that is different from the CruX dataset, where LCP load delay is the second largest sub-part.

Typically, LCP element rendering takes a long time if the LCP element hasn't been added to the DOM yet—a common issue with client-side generated content that we explore in the next section. Also, the main thread blocked by long tasks can contribute to the delay. In addition, render-blocking resources like stylesheets or synchronous scripts in the <head> can delay rendering.

It's interesting to observe the different LCP challenges that websites across various datasets face. While an average website from the CruX dataset struggles with image load delay, websites from the RUMvision dataset often face rendering delay issues. Nevertheless, all websites can benefit from using performance monitoring tools with Real User Monitoring (RUM), as these tools provide deeper insights into the performance issues experienced by real users.

LCP lazy-loading

One of the ways to optimize the LCP resource load delay is to ensure the resource can be discovered as early as possible. If you make the resource discoverable in the initial HTML document, it enables the LCP resource to begin downloading sooner. A big obstacle to LCP resource discoverability is lazy loading of the LCP resource.

Overall, lazy-loading images is a helpful performance technique that should be used to postpone loading of non-critical resources until they are near the viewport. However, using lazy-loading on the LCP image will delay the browser from loading it quickly. That is why lazy-loading should not be used on LCP elements. In this section, we explore how many sites use this performance anti-pattern.

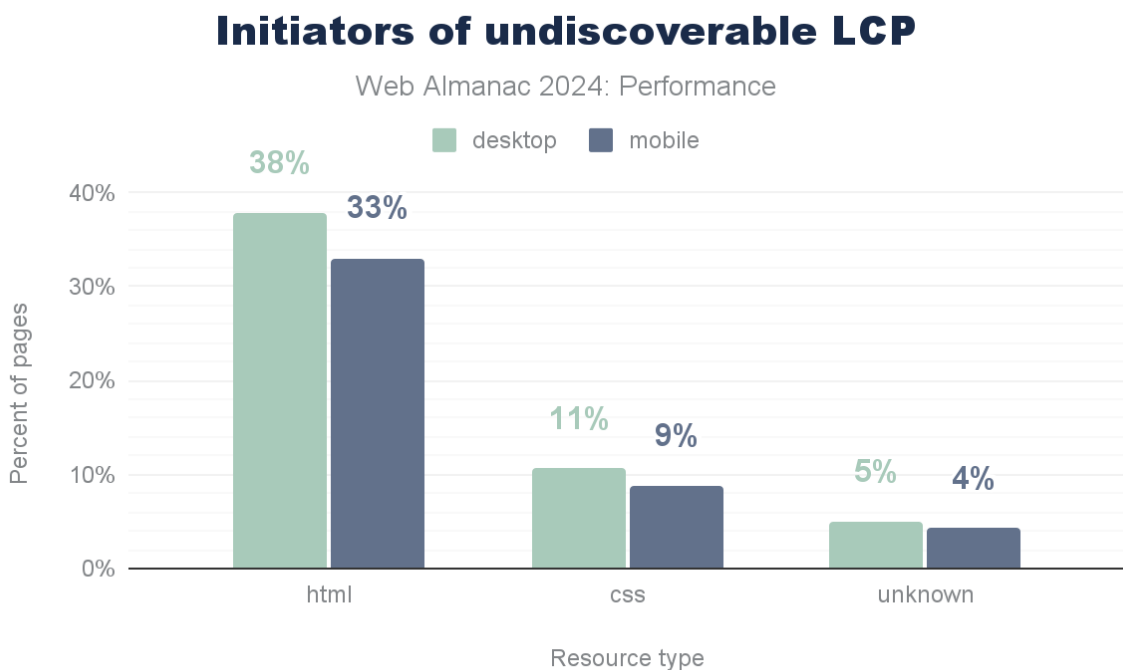
15.7%

```
{{ figure_markup(
  caption="The percent of mobile pages having image-based LCP that use native or custom
lazy-loading on it",
```

```
content="15.7%",
classes="big-number",
sheets_gid="1048885241",
sql_file="lcp_lazy.sql"
)
}}
```

The good news is that in 2024, fewer websites are using this performance anti-pattern. In 2022, 18% of mobile websites were lazy-loading their LCP images. By 2024, this decreased to 15.7%.

CSS background images



```
{{ figure_markup(
  image="",
  caption="The percent of pages whose LCP is not statically discoverable and initiated from a
given resource. ",
  description="Bar chart showing the initiators of undiscoverable LCP (Largest Contentful
Paint) for desktop and mobile, categorized by resource type. For desktop, 38% of pages
have HTML as the initiator of undiscoverable LCP, while for mobile, this figure is 33%. CSS
is responsible for 11% of undiscoverable LCP on desktop pages and 9% on mobile. An
unknown resource type accounts for 5% of undiscoverable LCP on desktop and 4% on
mobile.",
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=427
707611&format=interactive",
  sheets_gid="1131647963",
```

```
sql_file="lcp_initiator_type.sql"
)
}}
```

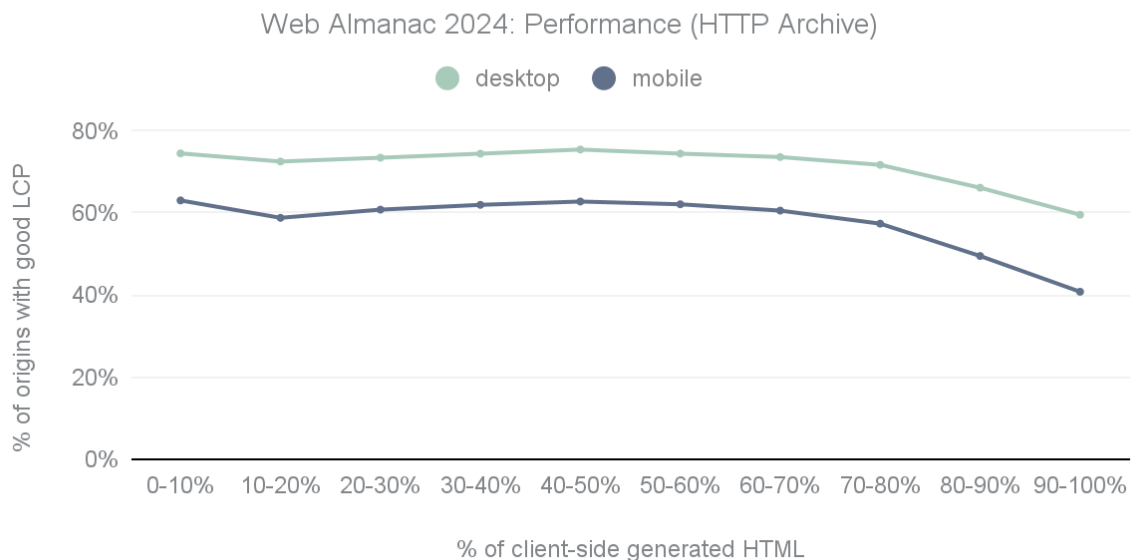
Also, websites that initiate LCP elements as CSS background images delay LCP static discovery until the CSS file is processed. The data shows that 9% of mobile pages initialize the LCP resource from CSS. Compared to 2022, this metric has remained unchanged.

Dynamically added images

One more common reason for non-discoverable LCP elements is dynamically added images. These images are added to the page through JavaScript after the initial HTML is loaded, making them undiscoverable during the HTML document scan.

The chart below illustrates the distribution of client-side generated content. It compares the initial HTML with the final HTML (after JavaScript runs) and measures the difference. It displays how the percentage of websites with good LCP changes as the percentage of client-side generated content increases.

Percentage of origins with good LCP vs client-side generated HTML



```
{{ figure_markup(
  image="",
  caption="The percent of websites with good LCP vs percentage of client-side generated
content on a page",
  description="Line chart showing the percentage of origins with good LCP compared to the
percentage of client-side generated HTML for both desktop and mobile. For desktop, the
```

percentage of origins with good LCP starts around 75% for pages with 0-10% client-side generated HTML and remains relatively stable, peaking slightly around 40-50% client-side HTML usage, before gradually declining to about 65% at the 90-100% range. For mobile, the percentage of good LCP starts lower, around 60% for the 0-10% range, and follows a similar trend, peaking slightly in the 30-40% range before declining more sharply to about 45% at the 90-100% client-side HTML usage.",

```
chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=750
231640&format=interactive",
  sheets_gid="829333856",
  sql_file="inp_long_tasks.sql"
)
}}
```

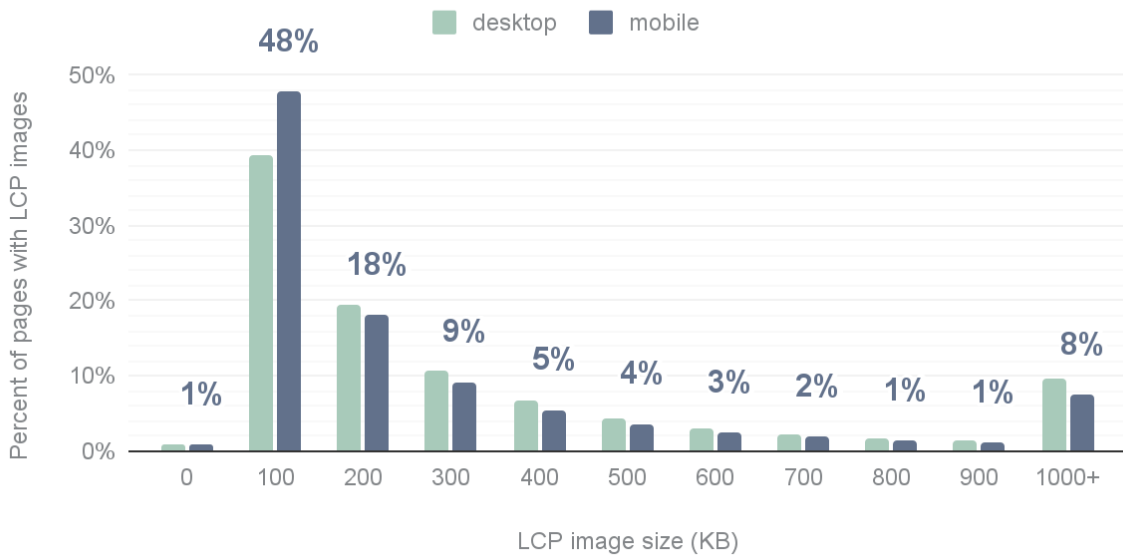
The percentage of pages with good LCP stays at approximately 60% for mobile devices until the amount of client-side generated content reaches 70%. After this threshold, the percentage of websites with good LCP starts to drop at a faster rate until ending at 40%. This suggests that a combination of server- and client-side generated content doesn't significantly impact how fast the LCP element gets rendered. However, fully rendering a website on the client side has a significantly negative impact on LCP.

LCP size

The [CruX and RUMvision data on LCP sub-parts](#) showed that resource load duration is rarely the main bottleneck for a slow LCP. However, it is still valuable to analyze the key optimization factors, such as the size and format of the LCP resource.

LCP image size histogram

Web Almanac 2024: Performance (HTTP Archive)



```
{{ figure_markup(  
  image="",  
  caption="Distribution of LCP image sizes, segmented by device type",  
  description="Histogram showing the distribution of LCP image sizes for desktop and mobile  
pages, measured in kilobytes (KB). For mobile, 48% of pages have LCP image sizes  
between 100 and 200 KB, while 18% of desktop pages fall into this range. For desktop, 30%  
of pages have LCP images between 0 and 100 KB, compared to 1% for mobile. In the 200 to  
300 KB range, 9% of desktop pages and 5% of mobile pages have LCP images. The  
percentages gradually decrease as image size increases, with only a small portion of pages  
having LCP images larger than 700 KB. At the highest range, 8% of both desktop and  
mobile pages have LCP images over 1000 KB.",
```

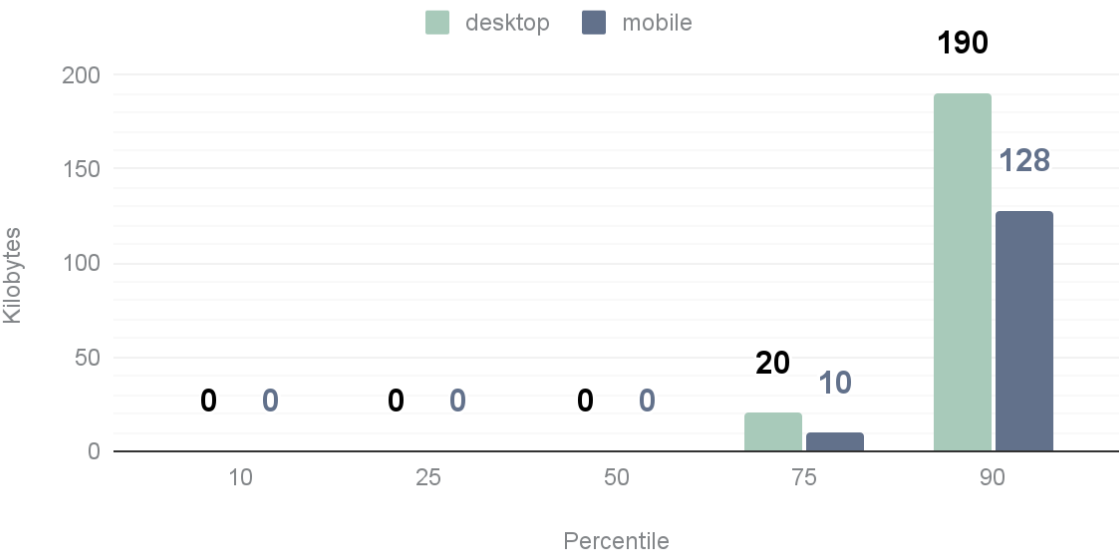
```
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah  
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=164  
375992&format=interactive",  
  sheets_gid="1329122831",  
  sql_file="lcp_bytes_histogram.sql"  
)  
}}
```

In 2024, 48% of mobile websites used an LCP image that was 100KB or less. Though, for 8% of the mobile pages the LCP element size is more than 1000KB.

This aligns with the [Lighthouse audit on unoptimized images](#), which also reports the amount of wasted kilobytes that could be saved by image optimization.

Distribution of wasted kilobytes on LCP image

Web Almanac 2024: Performance (Lighthouse)



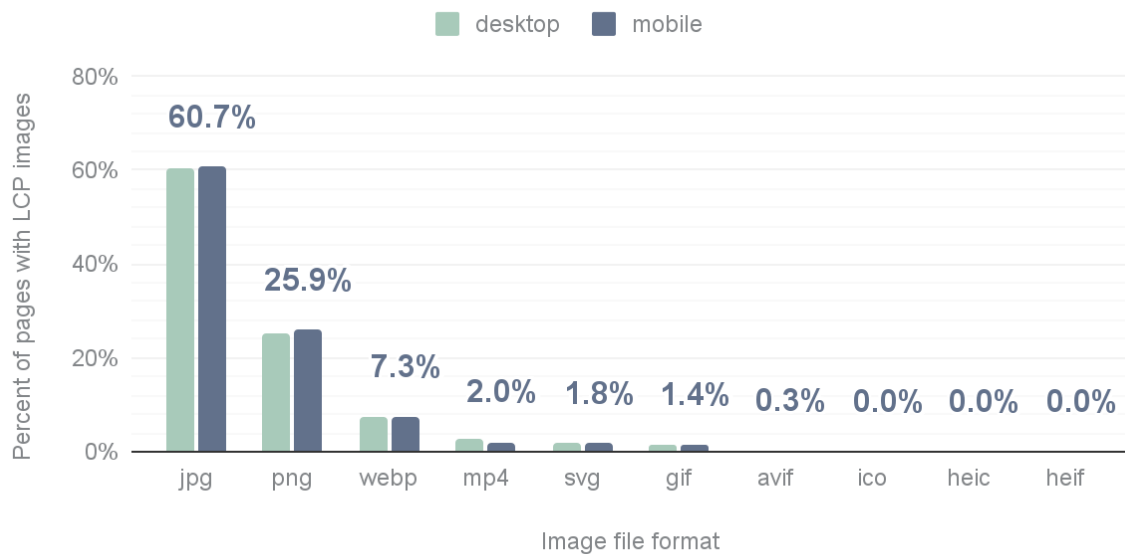
```
{{ figure_markup(  
  image="",  
  caption="Distribution of wasted kilobytes on LCP image",  
  description="Bar chart showing the distribution of wasted kilobytes on LCP images for  
desktop and mobile across percentiles. At the 10th, 25th, and 50th percentiles, both desktop  
and mobile pages have 0 wasted kilobytes. At the 75th percentile, desktop pages waste 20  
kilobytes, while mobile pages waste 10 kilobytes. At the 90th percentile, desktop pages  
waste 190 kilobytes, and mobile pages waste 128 kilobytes.",  
  
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah  
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=321  
466279&format=interactive",  
  sheets_gid="1984265626",  
  sql_file="lcp_wasted_bytes.sql"  
)  
}}
```

The audit results indicate that the median website wastes 0 KB on LCP images, i.e. serves optimized images. This leads to the conclusion that many sites are optimizing their LCP resources effectively, although some still need to improve.

You can reduce image sizes through resizing dimensions and increasing compression. Another way to reduce image sizes is by using new image formats like Webp and AVIF, which have better compression algorithms.

LCP image formats

Web Almanac 2024: Performance



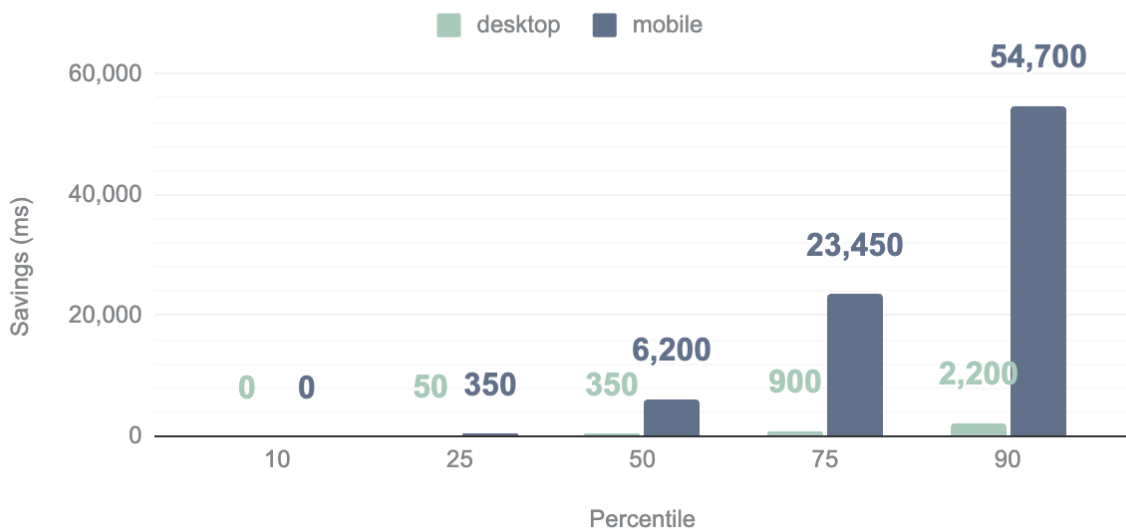
```
{{ figure_markup(  
  image="",  
  caption="The percent of pages that use a given image file format for their LCP images",  
  description="Bar chart showing the distribution of LCP (Largest Contentful Paint) image  
formats for desktop and mobile. JPG is the most common format, used by 60.7% of desktop  
pages and a similar percentage of mobile pages. PNG is the second most common format,  
used by 25.9% of pages. WebP follows with 7.3%, while other formats such as MP4, SVG,  
GIF, and AVIF are used by less than 2% of pages. ICO, HEIC, and HEIF formats are barely  
used, with their percentages rounding to 0% for both desktop and mobile.",
```

```
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah  
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=208  
6275423&format=interactive",  
  sheets_gid="240287365",  
  sql_file="lcp_format.sql"  
)  
}}
```

JPG and PNG still have the highest proportion of adoption at 86.6% combined, however WEBP and AVIF formats are both increasing in adoption. In comparison to 2022, WEBP image format usage increased from [4%](#) to 7.3%. Also, AVIF usage increased slightly from 0.1% to 0.3%. According to [Baseline](#), AVIF format is newly available across major browsers, so we expect to see higher adoption in the future.

Distribution of potential savings on FCP from render-blocking resources

Web Almanac 2024: Performance



Loading Speed Conclusions

- The percentage of websites with good FCP and LCP has improved, though TTFB showed no significant change.
- One cause for slow LCP is lazy-loading the LCP element. Usage of this antipattern has decreased, but 15% of websites still fail this test and could benefit from removing lazy-loading for their LCP elements.
- The adoption of modern image formats like AVIF and WEBP is growing for LCP elements.

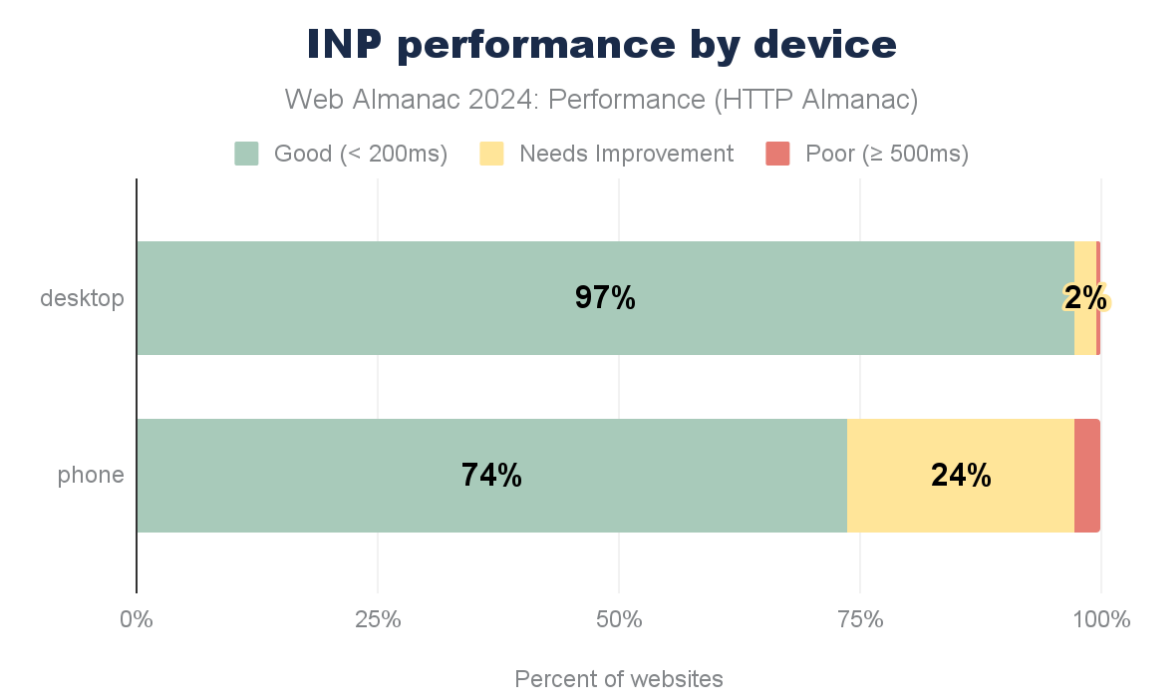
Interactivity

Interactivity on a website refers to the degree to which users can engage with and respond to content, features, or elements on the page. Measuring interactivity involves assessing the performance for a range of user interactions, such as clicks, taps, and scrolls, as well as more complex actions like form submissions, video plays, or drag-and-drop functions.

Interaction to Next Paint (INP)

Interaction to Next Paint ([INP](#)) is calculated by observing all the interactions made with a page during the session and reporting the worse latency (for most sites). An interaction's latency consists of the single longest duration of a group of event handlers that drive the interaction, from the time the user begins the interaction to the moment the browser is next able to paint a frame.

For an origin to receive a “good” INP score, at least 75% of all sessions need an INP score of 200ms or less. The INP score is the slowest or near-slowest interaction time for all interactions on the page. See [Details on how INP is calculated](#) for more information.

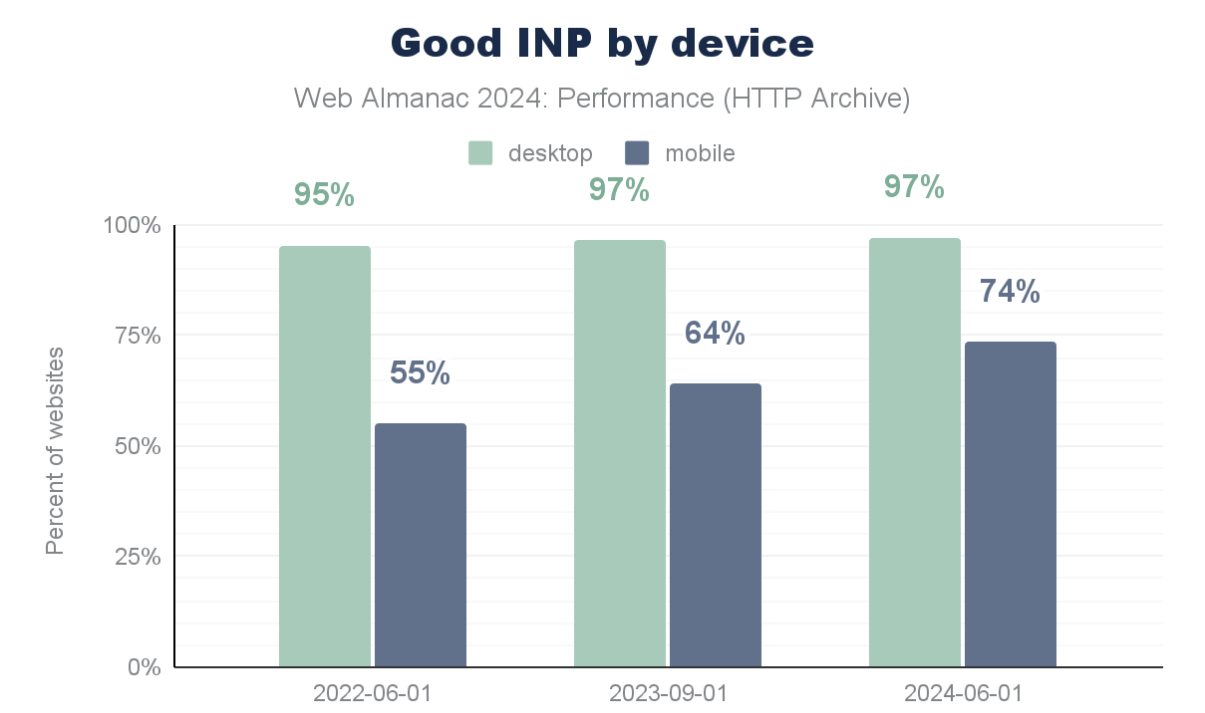


```
{{ figure_markup(  
  image="",  
  caption="Distribution of INP performance by device.",  
  description="Stacked bar chart showing INP performance by device, with desktop and  
phone categorized into good (under 200 ms), needs improvement (200–500 ms), and poor  
(over 500 ms). For desktop, 97% of websites have good INP, 2% need improvement, and  
less than 1% perform poorly. For phones, 74% of websites have good INP, 24% need  
improvement, and 2% perform poorly.",  
  
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah  
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=667  
078190&format=interactive",  
  sheets_gid="1535582002",  
  sql_file="web_vitals_by_device.sql"  
)  
}}
```

In 2024, 74% of mobile and 97% of desktop websites had good INP. Interestingly, the gap between mobile and desktop is huge, i.e. more than 20%.

The primary reason for weaker performance on mobile is its lower processing power and frequently poor network connections. Alex Russell’s article "[The Performance Inequality Gap](#)" (2023) raises the issue of the growing performance inequality gap caused by the

affordance of high-end vs low-end devices. As the prices of high-end devices rise, fewer users can afford them, widening the inequality gap.



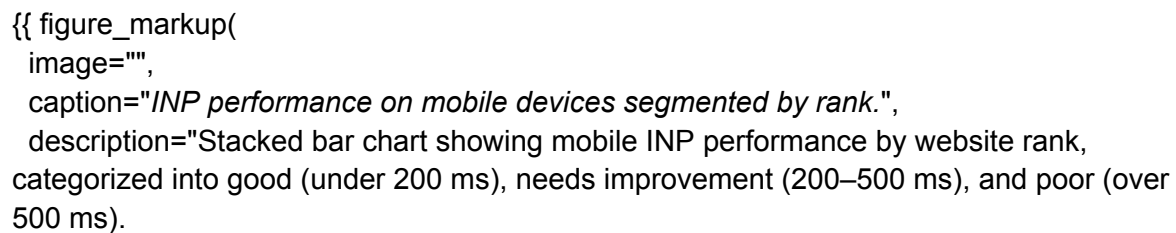
```
{{ figure_markup(  
  image="",  
  caption="Good INP score by device.",  
  description="Bar chart showing the percentage of websites with good INP performance by  
device (desktop and mobile) across three years. In 2022, 95% of desktop websites had good  
INP, while 55% of mobile websites achieved good INP. In 2023, the percentage of websites  
with good INP improved to 97% for desktop and 64% for mobile. By 2024, 97% of desktop  
websites maintained good INP performance, while mobile improved further to 74%."  
  ,  
  
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah  
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=416  
359271&format=interactive",  
  sheets_gid="1535582002",  
  sql_file="web_vitals_by_device.sql"  
  )  
}}
```

Although the INP metric displays worse results than the FID, there has been a positive tendency over the past three years.



The percentage of mobile pages having good INP increased from 55% in 2022 to 74% in 2024. This is a significant increase, and even though we can't be exactly sure what to

Mobile INP metric by rank reveals an interesting trend. [In the 2022 chapter](#), we assumed that the more popular a website is, the more performance optimizations it would have, leading to better performance. However, when it comes to INP, the opposite seems to be true.



For the top 1,000 websites, 53% have good INP, 41% need improvement, and 6% perform poorly. For the top 10,000 websites, 49% are in the good range, 44% need improvement, and 7% are poor. In the top 100,000, 51% are good, 43% need improvement, and 6% are poor. For the top 1,000,000 websites, 61% have good INP, 35% need improvement, and 4% are poor. As the rank increases to the top 10,000,000 websites, 73% are good, 24% need improvement, and 3% are poor. Finally, for the top 100,000,000 websites, 74% have good INP, 24% need improvement, and 2% are poor.

",

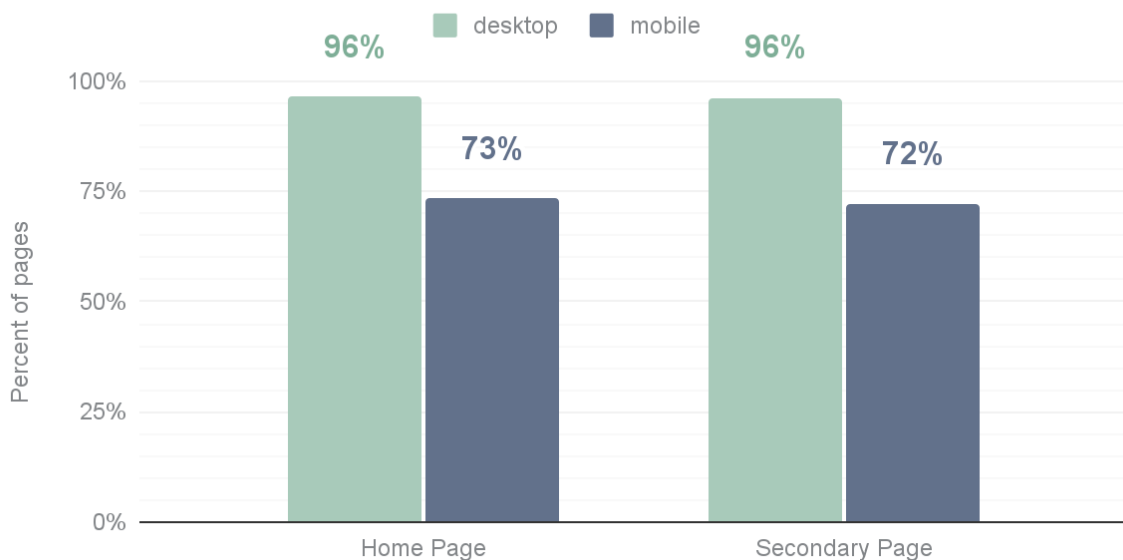
```
chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=296
559964&format=interactive",
  sheets_gid="355582610",
  sql_file="web_vitals_by_rank_and_device.sql"
)
}}
```

Fewer websites in the top 1,000 rank have good INP compared to the results for all websites. For example, 53% of the top 1,000 websites have a good INP score, while a much bigger percentage of all websites, i.e. 74%, meet this threshold.

This could be because the most visited websites often have more user interactions and complex functionality. Logically, the INP for an interactive e-commerce site would differ from a simple, static blog.

Good INP for home pages and secondary pages

Web Almanac 2024: Performance (Chrome UX Report)



```
{{ figure_markup(
  image="",
  caption="Good INP performance on Home and Secondary page by device.",
  description="Bar chart showing the percentage of pages with good INP for home pages
and secondary pages, separated by desktop and mobile performance. For home pages,
96% of desktop pages have a good INP, while 73% of mobile pages achieve a good INP. For
secondary pages, 96% of desktop pages also have a good INP, with 72% of mobile pages
reaching this performance level.",
```

```

chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=148
3510539&format=interactive",
  sheets_gid="1159394005",
  sql_file="web_vitals_by_device_secondary_pages.sql"
)
}}

```

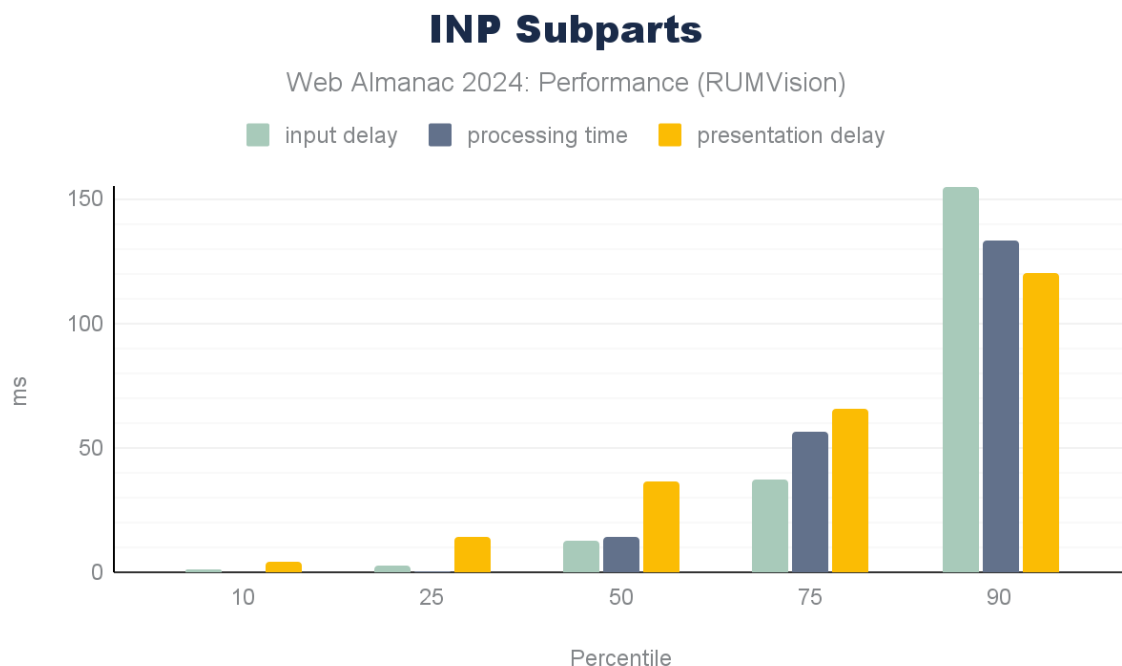
Unlike other performance metrics like FCP and LCP, the percentage of secondary pages with good INP does not differ from the home page results. This is likely because INP isn't as impacted by caching as loading speed is.

INP Subparts

An INP score can be broken down into three sub-parts:

- **Input Delay:** the time spent to finish processing the tasks that were already in the queue at the moment of the interaction
- **Processing Time:** the time spent processing the event handlers attached to the element which the user interacted with
- **Presentation Delay:** the time spent figuring out the new layout, if changed, and painting the new pixels on the screen

To optimize your website's interactivity, it's important to identify the duration of every sub-part.



```

{{ figure_markup(
  image="",
  caption="INP Subparts by percentile, Source: RUMvision",
  description="Bar chart showing the distribution of INP subparts in milliseconds (ms) by percentile. At the 10th percentile, all subparts (input delay, processing time, and presentation delay) are minimal. At the 25th percentile, the values slightly increase but remain below 10 ms. At the 50th percentile, input delay and processing time stay modest, while presentation delay reaches around 20 ms. At the 75th percentile, input delay increases to around 50 ms, with processing time and presentation delay also rising. At the 90th percentile, input delay reaches around 150 ms, and both processing time and presentation delay exceed 100 ms.",

  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ahQDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=226800794&format=interactive",
  sheets_gid="731456372",
)
}}

```

The INP sub-part duration distribution data from RUMvision shows that presentation delay (36ms) contributes the most to the median INP. As percentiles increase, input delay and processing time become longer. At the 75th percentile, input delay reaches 37ms and processing delay 56ms. By the 90th percentile, input delay jumps to 155ms, which makes it the biggest contributor to poor INP. One way to optimize input delay is by avoiding long tasks, which we explore in the Long Tasks section.

Long Tasks

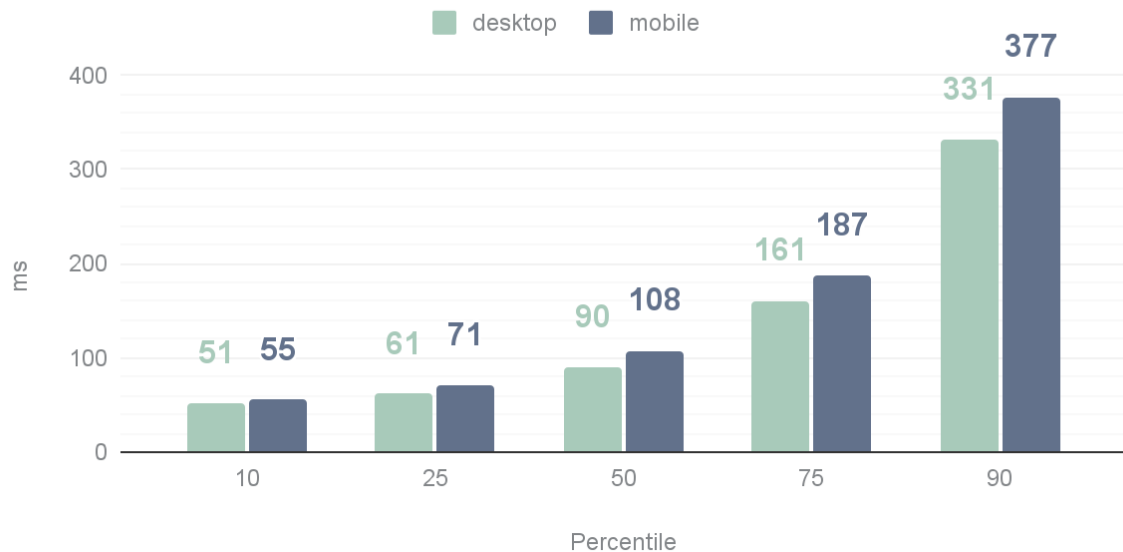
One of the sub-parts of INP is input delay, which can be longer than it should be due to various factors, including long tasks. [A task](#) is a discrete unit of work that the browser executes, and JavaScript is often the largest source of tasks. When a task exceeds 50 milliseconds, it is considered a long task. These long tasks can cause delays in responding to user interactions, directly affecting interactivity performance.

Due to the lack of same-source data for long tasks and INP, we decided not to correlate them.

We will, however, explore the average Long Task duration using data from RUMvision.

Task duration distribution

Web Almanac 2024: Performance (RUMvision)



```
{{ figure_markup(  
  image="xyz.png",  
  caption: "Task duration, segmented by device."  
  description="Bar chart showing the distribution of task duration in milliseconds (ms) by  
percentile and device type. At the 25th percentile, the task duration is 61ms for desktop and  
71 for mobile. At the 50th percentile, it increases to 90ms for desktop and 108ms for mobile.  
At the 75th percentile, task duration is 161ms for desktop and 187ms for mobile; at the 90th  
percentile, it reaches 331ms for desktop and 377 for mobile. This distribution shows that task  
durations grow significantly as we move from the 25th to the 90th percentile.",  
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah  
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=688  
921860&format=interactive",  
  sheets_gid="1272522211"  
  )  
}}
```

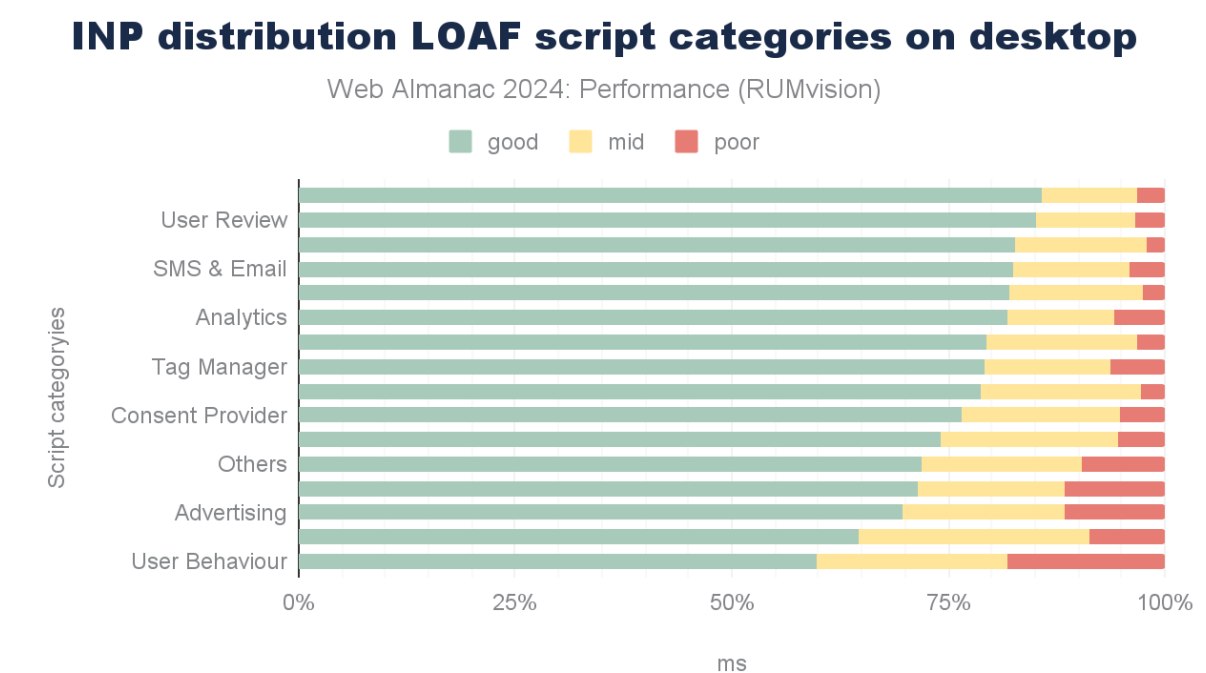
The task duration distribution shows a median task duration of 90ms for desktop and 108ms for mobile, which is twice more than the best practice recommendation of under 50ms. Less than 25% of websites have an optimal task duration below 50ms. We can also see that in every percentile, task duration on mobile sites is longer than on desktop sites, with the gap increasing as the percentile increases. On the 90th percentile, there is a 46ms difference between the average task duration between device types. This correlates well with INP scores that show better results on desktop compared to mobile.

Task duration data was retrieved using the Long Tasks API, which provides some useful data about performance issues, but it has limitations when it comes to accurately measuring sluggishness. It only identifies when a long task occurs and how long it lasts. It might

overlook essential tasks such as rendering. Due to these limitations, we will explore the Long Animation Frames API in the next section, which offers more detailed insights.

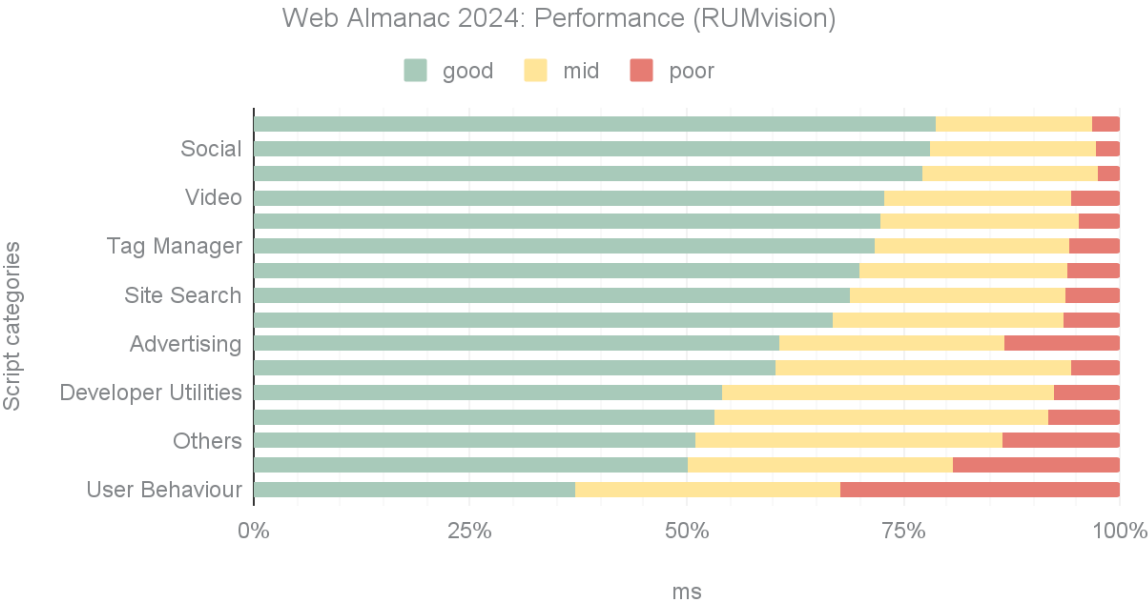
Long Animations Frames

Long Animation Frames (LoAF) are a performance timeline entry for identifying sluggishness and poor INP by tracking when work and rendering block the main thread. LoAF tracks animation frames instead of individual tasks like the Long Tasks API. A long animation frame is when a rendering update is delayed beyond 50 milliseconds (the same as the threshold for the Long Tasks API). It helps to find scripts that cause INP performance bottlenecks. This data allows us to analyze INP performance based on the categories of scripts responsible for LoAF.



```
{{ figure_markup(  
  image="xyz.png",  
  caption: "Distribution of INP performance segmented by script categories on desktop."  
  description="Stacked bar chart showing the distribution of INP across LOAF script  
categories for desktop, measured in milliseconds (ms). User Review, SMS & Email, and  
Analytics scripts perform best, with most of their INP in the good range. Tag Manager and  
Consent Provider scripts have more mid-range INP, with some falling into the poor category.  
Advertising, Others, and User Behaviour scripts perform worse, with the majority of INP  
falling into the poor range.",  
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah  
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=197  
5914925&format=interactive",  
  sheets_gid="947269170",  
  )  
}}
```

INP distribution LOAF script categories on mobile



```
{{ figure_markup(  
  image="xyz.png",  
  caption: "Distribution of INP performance segmented by script categories on mobile."  
  description="Stacked bar chart showing the distribution of INP across LOAF script  
categories for mobile, measured in milliseconds (ms). For Social scripts, most are in the  
good range, with few in the poor range. Video and Tag Manager scripts also have a majority  
in the good range but with a larger portion in the mid-range. Site Search and Advertising  
scripts have a more even distribution, with a significant part in the mid-range and some in  
the poor range. Developer Utilities, Others, and User Behaviour scripts perform worse, with  
most falling in the poor range.",  
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah  
QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=197  
8447282&format=interactive",  
  sheets_gid="947269170",  
  )  
}}
```

The top two categories contributing the most to slow INP scores on mobile and desktop devices are User Behavior (37% of mobile and 60% of desktop pages with good INP) and CDN/Hosting (50% of mobile and 65% of desktop pages with good INP).

User Behavior includes scripts from hosts like "script.hotjar.com", "smartlook.com", "newrelic.com", etc. While these tools provide valuable insights about users, our data shows that they can significantly degrade user experience by slowing down website interactions. CDN and Hosting script category examples come from domains like "cdn.jsdelivr.net", "ajax.cloudflare.com", "cdnjs.cloudflare.com", "cdn.shopify.com", "sdk.aws.waf.com", "cloudfront.net", "s3.amazonaws.com" and others. Having CDNs among the categories with the poorest INP results seems controversial because CDNs are usually recommended as a

performance optimization technique that reduces server load and delivers content faster to users. However, the CDNs included in this category usually deliver first- or third-party JavaScript resources, which contribute to LoAF and negatively impact interactivity.

On mobile devices, Consent Providers seem to have a significant impact on INP, resulting in only 53% of mobile pages having good INP when using one. This category consists of providers like "consentframework.com", "cookiepro.com", "cookiebot.com", "privacy-mgmt.com", "usercentrics.eu", and many others. On desktop devices, Consent Provider scripts show much better results, i.e. 76% of pages with good INP. This difference is likely due to the more powerful processors on desktop devices.

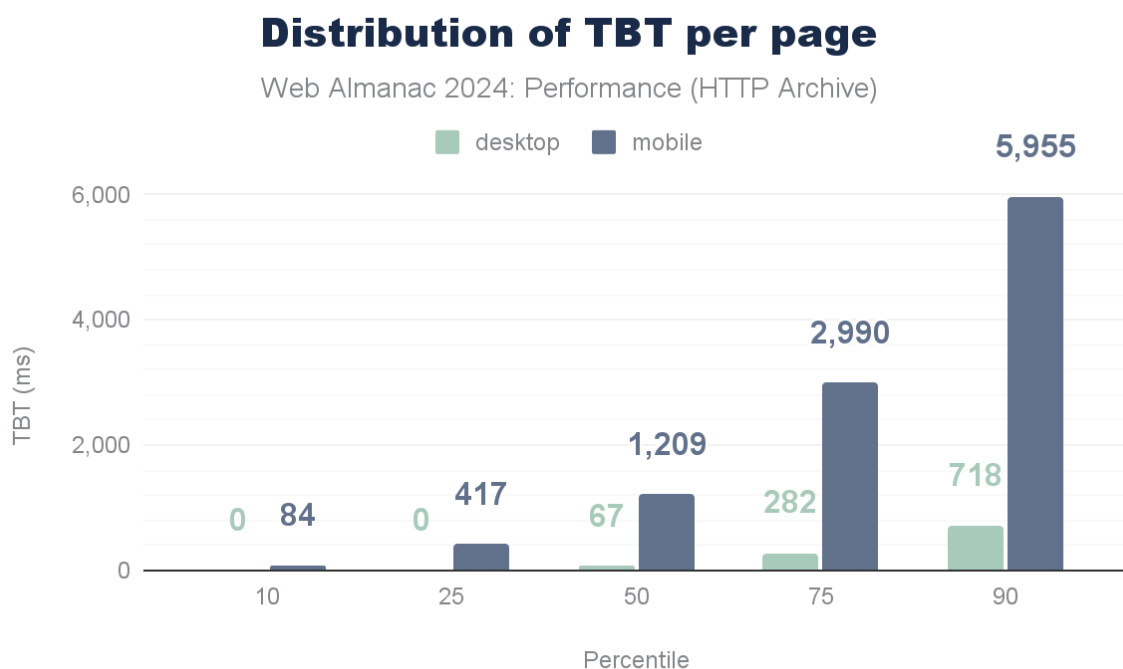
It is worth noting that the monitoring category, which also includes performance monitoring tools, has one of the least impacts on poor INP results. This is a good argument in favor of using web performance monitoring tools, as they help with valuable web performance insights without significantly affecting interactivity performance.

Total Blocking Time (TBT)

[Total Blocking Time](#) (TBT) measures the total amount of time after First Contentful Paint (FCP) where the main thread was blocked for long enough to prevent input responsiveness.

TBT is a lab metric and is often used as a proxy for field-based responsiveness metrics, such as INP, which can only be collected using real user monitoring, such as CrUX and RUMvision.

[Lab-based TBT and field-based INP](#) are correlated, meaning TBT results generally reflect INP trends. A TBT below 200ms is considered good, but most mobile websites exceed this target significantly.



```

{{ figure_markup(
  image="xyz.png",
  caption: "TBT per page by percentile."
  description="Bar chart showing the distribution of Total Blocking Time (TBT) per page in milliseconds (ms) by percentile. At the 10th percentile, both desktop and mobile TBT are near 0 ms. At the 25th percentile, desktop TBT is 84 ms, while mobile is 417 ms. At the 50th percentile, desktop has 67 ms of TBT, and mobile rises significantly to 1,209 ms. At the 75th percentile, desktop reaches 282 ms, with mobile at 2,990 ms. Finally, at the 90th percentile, desktop TBT is 718 ms, and mobile climbs to 5,955 ms",
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ahQDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=1525715716&format=interactive",
  sheets_gid="89045350",
  sql_file="js_bytes_rank.sql"
  )
}}

```

The median TBT on mobile is 1209 ms, which is 6 times higher than the best practice. In contrast, desktop websites show much better performance, with a median TBT of just 67ms. It is important to emphasize that the lab results use an emulated low-power device and a slow network, which may not reflect the real user data, as actual device and network conditions can vary. However, even with that in mind, these results still show that in the 90th percentile, user on mobile device will need to wait almost 6 seconds before the site becomes interactive.

With TBT being caused by long tasks it is not surprising to notice the same trend per percentiles as well as similar trend in gap between mobile and desktop in the two metrics results. It is also important to note that high TBT can be contributing to the input delay part of the INP, negatively impacting the overall INP score.

Interactivity Conclusion

The main takeaways of the interactivity results are:

- Despite the improvement in INP each year, a significant gap between desktop (97% good INP) and mobile (74% good INP) performance still exists.
- The top visited websites show poorer INP results compared to less popular ones.
- INP can be divided into three sub-parts—Input Delay, Processing Time, and Presentation Delay, and Presentation Delay has the biggest share of the median INP.
- Scripts from user behavior tracking, consent provider, and CDN categories are the main contributors to poor INP scores.

Visual Stability

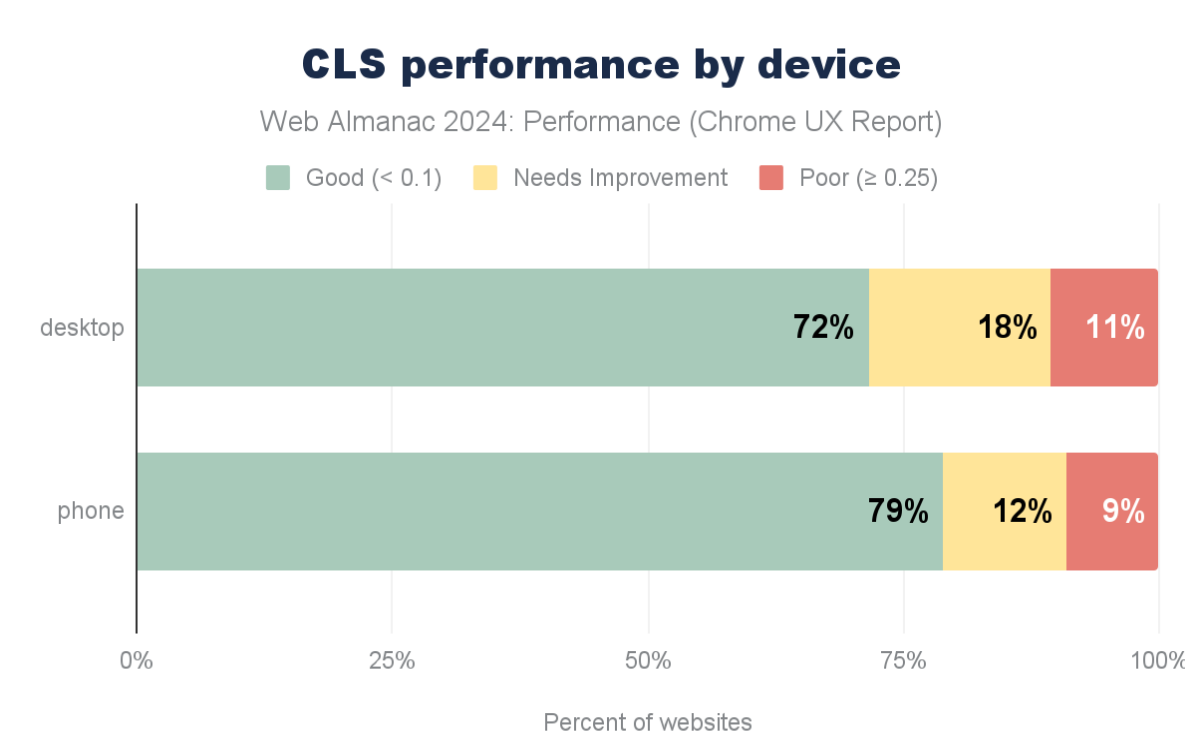
Visual stability on a website refers to the consistency and predictability of visual elements as the page loads and users interact with it. A visually stable website ensures that content does not

unexpectedly shift, move, or change layout, which can disrupt the user experience. These shifts often happen due to assets without specified dimensions (images and videos), third-party ads, heavy fonts, etc. The primary metric for measuring visual stability is [Cumulative Layout Shift \(CLS\)](#).

Cumulative Layout Shift (CLS)

[CLS](#) measures the biggest burst of layout shift scores for any unexpected layout shifts that happen while a page is open. Layout shifts occur when a visible element changes its position from one place to another.

A CLS score of 0.1 or less is considered good, meaning the page offers a visually stable experience, while scores between 0.1 and 0.25 indicate the need for improvement, and scores above 0.25 are considered poor, indicating that users may experience disruptive, unexpected layout shifts.



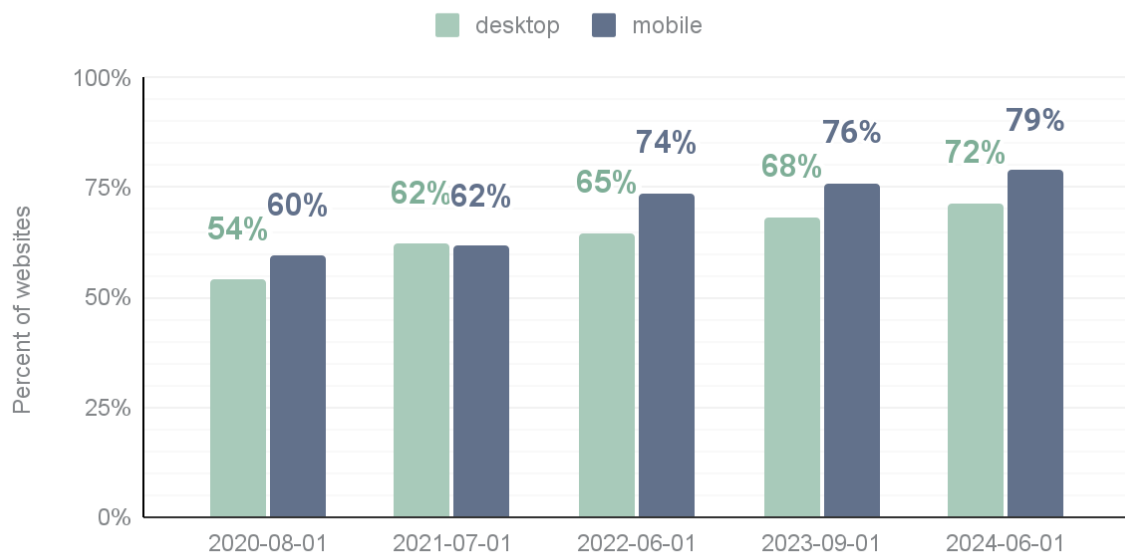
```
{{ figure_markup(  
  image="good-cls-by-device-2024.png",  
  caption: "CLS performance by device for 2024."  
  description="Stacked bar chart showing CLS performance for 2024 by device. For desktop  
  sites, 72% of sites had good CLS score, 18% need improvement, and 11% are considered  
  poor. For mobile sites, 79% of sites have a good score, 12% need improvement, and 9%  
  have a poor score.",  
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ah  
  QDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=127  
  1338928&format=interactive",  
  sheets_gid="1535582002",  
  sql_file="web_vitals_by_device.sql"
```

```
)  
}}
```

In 2024, 72% of websites achieved good CLS scores, while 11% had poor ones. We can also see that websites on mobile devices provide a better user experience when it comes to site stability than desktop sites.

Good CLS by device

Web Almanac 2024: Performance (Chrome UX Report)



```
{{ figure_markup(  
  image="good-cls-by-device.png",  
  caption: "The percent of websites having good CLS, segmented by device and year."  
  description="Bar chart showing the number of websites with good CLS segmented by  
device and years. The percentage of mobile sites having good CLS was 60% for year 2020.,  
62% for 2021, 74% for 2022, 76% for 2023, and 79% for 2024. For desktop sites, 54% had  
good CLS in 2020, 62% in 2021, 65% in 2022, 68% in 2023, and 72% in 2024.",  
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx\_5nC9ahQDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=1974391374&format=interactive",  
  sheets_gid="1535582002",  
  sql_file="web_vitals_by_device.sql"  
)  
}}
```

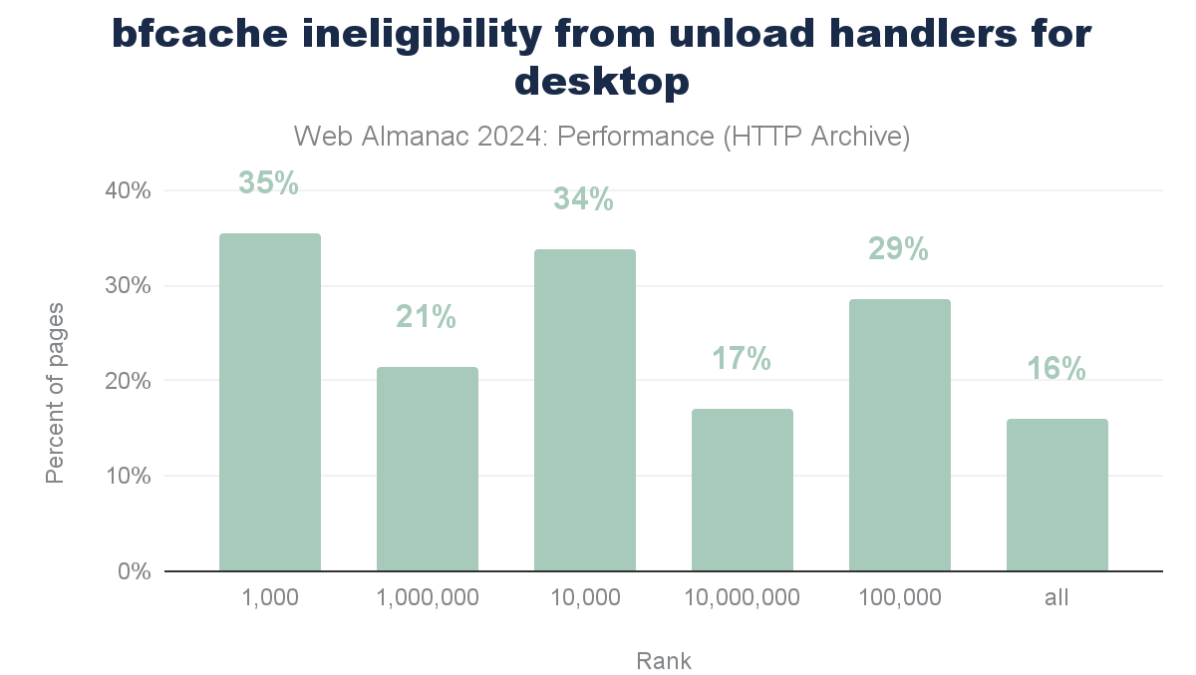
Looking at the metrics over time, we can see a nice upward trend. There is an increase from 60% of websites with good visual stability in 2020 to almost 80% in 2024. A visible jump in mobile data is already addressed in detail and attributed to the introduction of bfcache in [the 2022 Chapter](#). There is still a visible difference from 2022, so we will look in detail at some of the aspects that possibly contributed to it.

Bfcache

The [back/forward cache \(bfcache\)](#) is a browser optimization feature that improves the speed and efficiency of navigating between web pages by caching a fully interactive snapshot of a page in memory when a user navigates away from it. However, not all sites are eligible for bfcache. With an [extensive eligibility criteria](#), the easiest way to check if the site is eligible is to [test it in Chrome DevTools](#).

Let's look deeper by checking a few eligibility criteria that are quite a common cause and easily measurable using lab data.

One of the “usual suspects” is the `unload` event that is triggered when a user navigates away from a page. Due to its unreliable behaviour and its incompatibility with how bfcache preserves a page's stage, `unload` event makes the page ineligible for bfcache. Important to note here is that this feature is specific for browsers on desktops. Mobile sites ignore the `unload` event. This behavior could explain CLS improvement over the years gap between mobile and desktop numbers from *Figure xxx: The percent of websites having good CLS, segmented by device and year*.



```
{{ figure_markup(  
  image="xyz.png",  
  caption: " Usage of unload by site rank."  
  description="Bar chart showing the percentage of pages ineligible for bfcache  
(back-forward cache) due to unload handlers, by rank, for desktop and mobile devices. For  
the top 1,000 websites, 32% of desktop pages and 30% of mobile pages are ineligible. For  
the top 10,000, 30% of desktop and 26% of mobile pages are ineligible. In the top 100,000,  
26% of desktop and mobile pages are ineligible. At the 1,000,000 rank, 20% of desktop  
pages and 19% of mobile pages are ineligible. At the 10,000,000 rank, 17% of desktop and
```

16% of mobile pages are ineligible, while for all ranks, 16% of desktop and mobile pages are ineligible.",

```
chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ahQDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=1163433950&format=interactive",
sheets_gid="1706831462",
  sql_file="bfcache_unload.sql"
)
}}
```

From the above chart showing **unload** events from pages displayed on desktop devices, we can see a few interesting things. Overall event usage is quite low, 16%. However, it increases drastically for the top 1000 sites, to 32%, indicating that more popular sites probably use quite some more 3rd party services that often use this specific event.

It is expected to see this decrease in the use of unload events with major browsers like Google Chrome and Firefox moving towards its deprecation since around 2020 and encouraging the use of alternative events like **pagehide** and **visibilitychange**. These events are more reliable, do not block the browser's navigation, and are compatible with bfcache, allowing pages to be preserved in memory and restored instantly when users navigate back or forward.

Another common reason for websites to fall in the bfcache ineligibility category is the use of the **cache-control: no-store** directive. This cache control header instructs the browser (and any intermediate caches) not to store a copy of the resource, ensuring that the content is fetched from the server on every request.

21.20%

```
{{ figure_markup(
  caption="Percentage of mobile sites using cache-control:no-store",
  content="21.20%",
  classes="big-number",
  sheets_gid="XX",
  sql_file=""
)
}}
```

21.20% of mobile sites and 21.29% of desktop sites are using **cache-control:no-store**. That is a slight decrease from the 2022 report when this measure was about 22%.

When bfcache was first introduced, it brought noticeable improvements to Core Web Vitals. Based on that, Chrome is gradually bringing bfcache to more sites that were previously

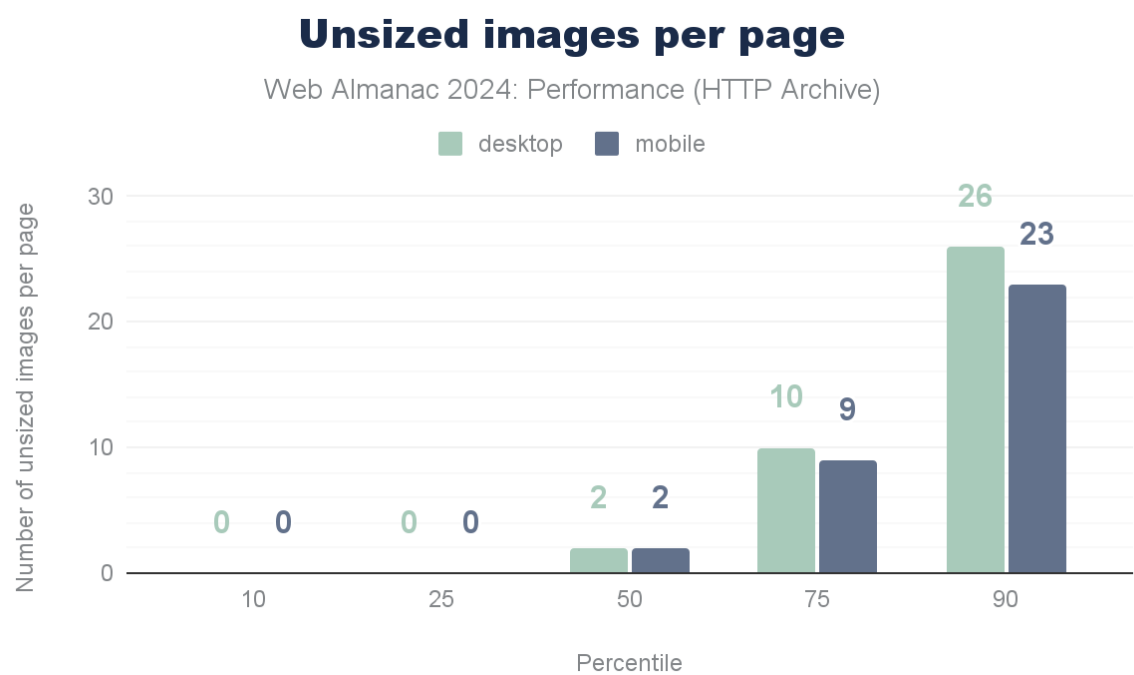
ineligible due to the use of the Cache-Control: no-store header. This change aims to further improve site performance.

Unload event, as well as cache-control:no-store, do not directly affect the page's visual stability. As already mentioned, the concept of bfcache load as a side-effect has this positive impact by eliminating some potential issues affecting metrics directly, such as unsized images or dynamic content. To continue exploring the visual stability aspect of the web, let's check some of the practices that directly impact the CLS.

CLS best practices

Explicit dimensions

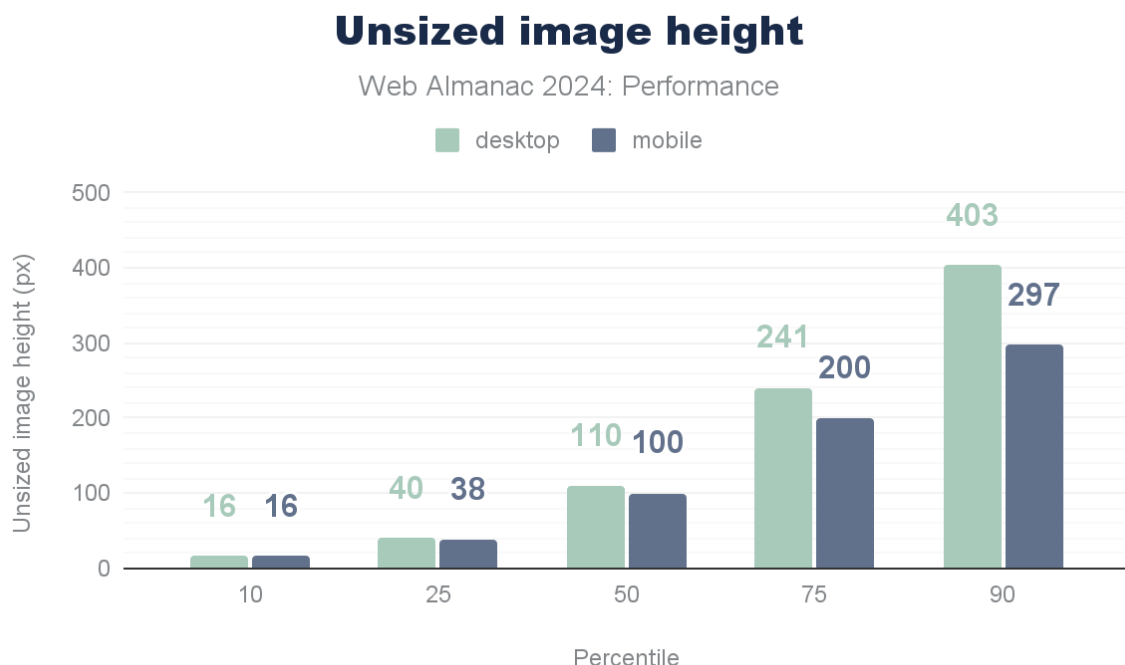
One of the most common reasons for unexpected layout shifts is not preserving space for assets or incoming dynamic content. For example, adding `width` and `height` attributes on images is one of the easiest ways to preserve space and avoid shifts.



```
{{ figure_markup(  
  image="xyz.png",  
  caption: "The number of unsized images per page"  
  description="Bar chart showing the number of unsized images per page by percentile for  
desktop and mobile devices. At the 10th and 25th percentiles, both desktop and mobile  
pages have 0 unsized images. At the 50th percentile, both desktop and mobile pages have 2  
unsized images. At the 75th percentile, desktop pages have 10 unsized images, while
```

mobile pages have 9. At the 90th percentile, desktop pages have 26 unsized images, and mobile pages have 23.",
chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ahQDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=603112996&format=interactive",
sheets_gid="1674162543",
sql_file="cls_unsized_images.sql"
)
}}

The median number of unsized images per web page is two. When we shift to the 90th percentile, that number jumps to 26 for desktop sites and 23 for mobile. Having unsized images on the page can be a risk for layout shift; however, an important aspect to look at is if images are affecting the viewport and if yes, how much.



{{ figure_markup(
image="unsized-images.png",
caption="Distribution of the heights of unsized images.",
description="Bar chart showing the 10, 25, 50, 75, and 90th percentile height of unsized images. The values for mobile are 16, 38, 100, 200, and 297px tall, respectively. The values for the desktop are larger: 16, 40, 110, 241, and 403.",

chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vRiPhLGIGUxomTx_5nC9ahQDRxZBmJXMT3Q0Z2z4Y2pPVqC9kzjsUjRk4hz-JZzaPBjVxyaf7Gtqh93/pubchart?oid=603112996&format=interactive",
sql_file="cls_unsized_image_height.sql"
)
}}

The median mobile site has unsized images of about 100px in height. Our test devices have a mobile viewport height of 512px, representing almost 20% of the screen width. This can potentially be shifted down when an unsized (full-width) image loads, which is not an insignificant shift.

As expected, image heights on desktop pages are larger, with the size on the median being 110px and on the 90th percentile 403px.

Fonts

Fonts can directly impact CLS. When web fonts are loaded asynchronously, a delay occurs between the initial rendering of the page and the time when the custom fonts are applied. During this delay, browsers often display text using a fallback font, which can have different dimensions (width, height, letter spacing) compared to the web font. When the web font finally loads, the text may shift to accommodate the new dimensions, causing a visible layout shift and contributing to a higher CLS score.

85%

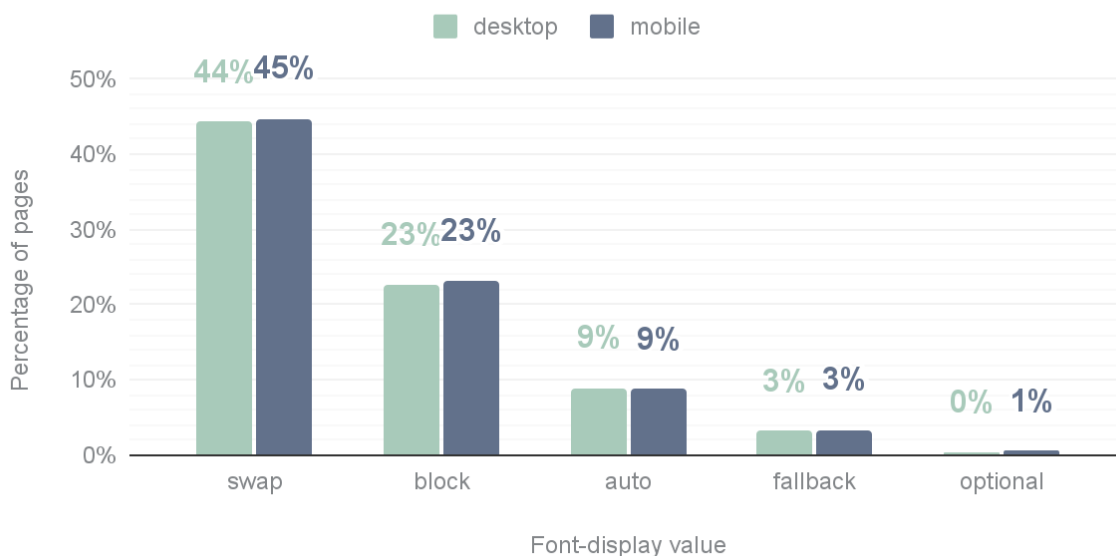
<The percent of mobile pages that use web fonts.>

```
{{ figure_markup(  
  caption="The percent of mobile pages that use web fonts.",  
  content="85%",  
  classes="big-number",  
  sheets_gid="XX",  
  sql_file="font_usage_mobile.sql"  
)  
}}
```

Using system fonts is one way to fix this issue. However, with 85% of mobile pages using web fonts it is not very likely that they will stop being used any time soon. A way to control the visual stability of a site that uses web fonts is to use the `font-display` property in CSS to control how fonts are loaded and displayed. [Different font-display strategies can be used depending on the team's decision about the tradeoff between performance and aesthetics.](#)

Usage of font-display

Web Almanac 2024: Fonts



```
{{ figure_markup(  
  image="font-display-usage.png",  
  caption="Usage of font-display.",  
  description="Bar chart showing the percent of pages that use the various font-display  
values. 45% of mobile pages use swap, 23% block, 9% auto, 3% fallback, and 1% use  
optional. The values for desktop are similar.",  
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vTHmcrit1gMzxfZNeFp9LrA4  
NQSMEh140fapD4CFQ89knpY6LvEKz7VafGaFGlxCAXTdpLZXaVaq8Pg/pubchart?oid=145  
8420916&format=interactive",  
  sql_file=""  
)  
}}
```

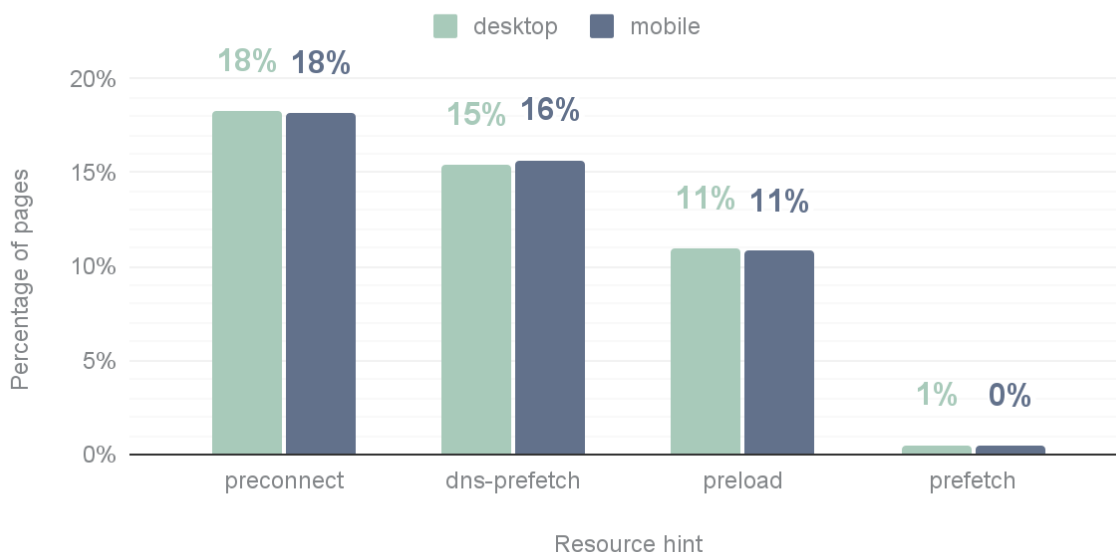
From the data displayed above, we can see that around 44% of both mobile and desktop sites use `font-display:swap` while 23% of sites use `font-display:block`. 9% of sites set the `font-display` property to `auto` and 3% use the `fallback` property. Only around 1% of sites use the `optional` strategy.

Compared to the 2022 data, there is a visible increase in the use of all `font-display` strategies, the biggest one being on `swap`, whose usage on both mobile and desktop pages jumped from around 30% in 2022 to over 44%.

Since most `font-display` strategies can contribute to CLS, we need to look at other strategies for minimizing potential issues. One of those is using resource hints to ensure third-party fonts are discovered and loaded as soon as possible.

Fonts resource hints usage

Web Almanac 2024: Fonts



```
{{ figure_markup(  
  image="fonts-resource-hints.png",  
  caption="Adoption of resource hints for font resources.",  
  description="Bar chart showing the percent of pages that use each type of resource hint on  
web fonts. 18% of mobile pages use preconnect, 16% dns-prefetch, 11% preload, and less  
than 1% prefetch. The values for desktop are almost the same.",  
  chart_url="https://docs.google.com/spreadsheets/d/e/2PACX-1vTHmcrit1gMzxfZNeFp9LrA4  
NQSMEh140fapD4CFQ89knpY6LvEKz7VafGaFGlxCAXtdpLZXaVaq8Pg/pubchart?oid=769  
711215&format=interactive",  
  sql_file="font_resource_hints_usage.sql"  
)  
}}
```

Around 11% of all tested mobile and desktop pages are preloading their web fonts, indicating to the browser that they should download these files, hopefully early enough to avoid shifts due to late font arrival. Note that using preload incorrectly can harm performance instead of helping it. To avoid this, we need to make sure that the preloaded font will be used and that we don't preload too many assets. Preloading too many assets can end up delaying other, more important resources.

18% of sites are using **preconnect** to establish an early connection to a third-party origin. Like with **preload** it is important to use this resource hint carefully and not to overdo it.

Animations

Another cause of unexpected shifts can be **non-composited** CSS animations. These animations involve changes to properties that impact the layout or appearance of multiple elements, which forces the browser to go through more performance-intensive steps like

recalculating styles, reflowing the document, and repainting pixels on the screen. The best practice is to use CSS properties such as `transform` and `opacity` instead.

```
{{ figure_markup(  
  content="39%",  
  caption="The percent of mobile pages that have non-composited animations.",  
  classes="big-number",  
  
  sheets_gid="https://docs.google.com/spreadsheets/d/15038wEloqY53Y_kR8U6QWM-PBO3  
1ZySQGi147ABTNBc/edit?gid=293393420#gid=293393420",  
  sql_file="cls_animations.sql",  
  )  
}}
```

39% of mobile pages and 42% of desktop pages still use non-composited animations, which is a very slight increase from 38% for mobile and 41% for desktop in the analysis from 2022.

Visual Stability Conclusion

Visual stability of the site can have a big influence on the user experience of the page. Having text shifting around while reading or a button we were just about to click disappear from the viewport can lead to user frustration. The good news is that Cumulative Layout Shift (CLS) continued to improve in 2024. That indicates that more and more website owners are adopting good practices such as sizing images and preserving space for dynamic content, as well as optimizing for bfcache eligibility to benefit from this browser feature.

Conclusion

Web performance continued to improve in 2024, with positive trends across many key metrics. We now have a more comprehensive metric to assess website interactivity—INP—which hopefully should lead to even greater performance optimizations.

However, challenges remain. For example, there is still a significant gap in INP performance between desktop and mobile. Presentation Delay is the main contributor to poor INP, mostly caused by third-party scripts for behavior tracking, consent providers, and CDNs.

Visual stability continues to improve by the adoption of best practices like proper image sizing and preserving space for dynamic content. Additionally, with recent changes in Chrome's bfcache eligibility, more sites will benefit from faster back and forward navigation.

Overall, web performance is on a promising track, making loading times faster, interactivity smoother, and visual stability more reliable. However, the difference between mobile and desktop experiences remains large. In future Web Almanac reports, we hope to see this gap decreasing, making the web experience consistent across all devices.