




Interaction to Next Paint (INP) becomes a Core Web Vital on March 12. Start making your websites more responsive to user input today. [Learn how.](#)

(https://web.dev/blog/inp-cwv-march-12?utm_source=web.dev&utm_medium=banner&utm_campaign=inp-cwv)

User-centric performance metrics



Philip Walton

 (<https://twitter.com/philwalton>)  (<https://github.com/philipwalton>)  (<https://philipwalton.com/>)

We've all heard how important performance is. But when we talk about performance—and about making websites "fast"—what specifically do we mean?

The truth is performance is relative:

- A site might be fast for one user (on a fast network with a powerful device) but slow for another user (on a slow network with a low-end device).
- Two sites may finish loading in the exact same amount of time, yet one may *seem* to load faster (if it loads content progressively rather than waiting until the end to display anything).
- A site might *appear* to load quickly but then respond slowly (or not at all) to user interaction.

So when talking about performance, it's important to be precise and to refer to performance in terms of objective criteria that can be quantitatively measured. These criteria are known as *metrics*.

But just because a metric is based on objective criteria and can be quantitatively measured, it doesn't necessarily mean those measurements are useful.

Defining metrics

web.dev uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic. [Learn more.](#)

HIDE

fetching content and displaying anything on the page until several seconds after the `load` event fires. While such a page might technically have a fast load time, that time would not correspond to how a user actually experiences the page loading.

Over the past few years, members of the Chrome team—in collaboration with the [W3C Web Performance Working Group](https://www.w3.org/webperf/) (<https://www.w3.org/webperf/>)—have been working to standardize a set of new APIs and metrics that more accurately measure how users experience the performance of a web page.

To help ensure the metrics are relevant to users, we frame them around a few key questions:

Is it happening?	Did the navigation start successfully? Has the server responded?
Is it useful?	Has enough content rendered that users can engage with it?
Is it usable?	Can users interact with the page, or is it busy?
Is it delightful?	Are the interactions smooth and natural, free of lag and jank?

How metrics are measured

Performance metrics are generally measured in one of two ways:

- **In the lab:** using tools to simulate a page load in a consistent, controlled environment
- **In the field:** on real users actually loading and interacting with the page

Neither of these options is necessarily better or worse than the other—in fact you generally want to use both to ensure good performance.

In the lab

Testing performance in the lab is essential when developing new features. Before features are released in production, it's impossible to measure their performance characteristics on real users, so testing them in the lab before the feature is released is the best way to prevent performance regressions.

In the field

web.dev uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic. [Learn more.](#)

HIDE

Moreover, page loads may not be deterministic. For example, sites that load personalized content or ads may experience vastly different performance characteristics from user to user. A lab test will not capture those differences.

The only way to truly know how your site performs for your users is to actually measure its performance as those users are loading and interacting with it. This type of measurement is commonly referred to as Real User Monitoring (https://en.wikipedia.org/wiki/Real_user_monitoring)—or RUM for short.

Types of metrics

There are several other types of metrics that are relevant to how users perceive performance.

- **Perceived load speed:** how quickly a page can load and render all of its visual elements to the screen.
- **Load responsiveness:** how quickly a page can load and execute any required JavaScript code in order for components to respond quickly to user interaction
- **Runtime responsiveness:** after page load, how quickly can the page respond to user interaction.
- **Visual stability:** do elements on the page shift in ways that users don't expect and potentially interfere with their interactions?
- **Smoothness:** do transitions and animations render at a consistent frame rate and flow fluidly from one state to the next?

Given all the above types of performance metrics, it's hopefully clear that no single metric is sufficient to capture all the performance characteristics of a page.

Important metrics to measure

- **First Contentful Paint (FCP)** (</articles/fcp>): measures the time from when the page starts loading to when any part of the page's content is rendered on the screen. (*lab* (#in_the_lab), *field* (#in_the_field))
- **Largest Contentful Paint (LCP)** (</articles/lcp>): measures the time from when the page starts loading to when the largest text block or image element is rendered on the screen. (*lab*

web.dev uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic. [Learn more](#).

HIDE

- **Interaction to Next Paint (INP)** (/articles/inp): measures the latency of every tap, click, or keyboard interaction made with the page, and—based on the number of interactions—selects the worst interaction latency of the page (or close to the highest) as a single, representative value to describe a page's overall responsiveness. (*lab* (#in_the_lab), *field* (#in_the_field))
- **Total Blocking Time (TBT)** (/articles/tbt): measures the total amount of time between FCP and TTI where the main thread was blocked for long enough to prevent input responsiveness. (*lab* (#in_the_lab))
- **Cumulative Layout Shift (CLS)** (/articles/cls): measures the cumulative score of all unexpected layout shifts that occur between when the page starts loading and when its lifecycle state (<https://developer.chrome.com/blog/page-lifecycle-api/>) changes to hidden. (*lab* (#in_the_lab), *field* (#in_the_field))
- **Time to First Byte (TTFB)** (/articles/ttfb): measures the time it takes for the network to respond to a user request with the first byte of a resource. (*lab* (#in_the_lab), *field* (#in_the_field))

While this list includes metrics measuring many of the various aspects of performance relevant to users, it does not include everything. For example, runtime responsiveness and smoothness are not currently covered.

In some cases, new metrics will be introduced to cover missing areas, but in other cases the best metrics are ones specifically tailored to your site.

Custom metrics

The performance metrics listed above are good for getting a general understanding of the performance characteristics of most sites on the web. They're also good for having a common set of metrics for sites to compare their performance against their competitors.

However, there may be times when a specific site is unique in some way that requires additional metrics to capture the full performance picture. For example, the LCP metric is intended to measure when a page's main content has finished loading, but there could be cases where the largest element is not part of the page's main content and thus LCP may not be relevant.

To address such cases, the Web Performance Working Group has also standardized lower-level APIs that can be useful for implementing your own custom metrics:

web.dev uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic. [Learn more.](#)

HIDE

- [Resource Timing API](https://w3c.github.io/resource-timing/) (https://w3c.github.io/resource-timing/)
- [Server timing](https://w3c.github.io/server-timing/) (https://w3c.github.io/server-timing/)

Refer to the guide on [Custom Metrics](/articles/custom-metrics) (/articles/custom-metrics) to learn how to use these APIs to measure performance characteristics specific to your site.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (https://developers.google.com/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2019-11-08 UTC.

web.dev uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic. [Learn more](#).

HIDE