

Interaction to Next Paint (INP)

explains how it works, how to measure it, and offers suggestions on how to improve it.



Jeremy Wagner

 (<https://github.com/malchata>)  (<https://www.linkedin.com/in/malchata>) 

(<https://jlwagner.net/>)

Browser Support	96	96	X	X
-----------------	----	----	---	---

[Source](https://developer.mozilla.org/docs/Web/API/PerformanceEventTiming/interactionId) (<https://developer.mozilla.org/docs/Web/API/PerformanceEventTiming/interactionId>)

Interaction to Next Paint (INP) is a [stable](#) (/articles/vitals#stable) Core Web Vital metric that assesses page responsiveness using data from the Event Timing API. INP observes the latency of all click, tap, and keyboard interactions with a page throughout its lifespan, and reports the longest duration, ignoring outliers. A low INP means the page is consistently able to respond quickly to the vast majority of user interactions.

Good responsiveness means a page responds quickly to interactions. When a page responds to an interaction, the browser presents *visual feedback* in the next frame it renders to show that the interaction was successful. For example, it can provide feedback on the following:

- Whether an item you add to an online shopping cart is actually being added.
- Whether a mobile navigation menu has opened.
- Whether a login form's contents are being authenticated by the server.

Some interactions naturally take longer than others, but for especially complex interactions, it's important to quickly present some initial visual feedback to tell the user something is

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

HIDE

The goal of INP is minimize the time from when a user initiates an interaction until the next frame is painted, for all or most interactions the user initiates.

gShoe product Q&A:

- What is gShoe?
- What technology does gShoe use?
- How much does gShoe cost? 

gShoe product Q&A:

- What is gShoe?
- What technology does gShoe use?
- How much does gShoe cost? 

Poor responsiveness

An example of poor versus good responsiveness. On the left, long tasks block the accordion from opening. This causes the user to click multiple times, thinking the experience is broken. When the main thread catches up, it processes the delayed inputs, resulting in the accordion opening and closing unexpectedly. On the right, a more responsive page opens the accordion quickly and without incident.

➤ A note on how INP is calculated

INP is calculated by observing all the interactions made with a page. For most sites, the interaction with the worst latency is reported as INP. However, for pages with large numbers of interactions, random hiccups can result in an unusually high-latency interaction on an otherwise responsive site. The more interactions, the more likely this is to happen. To counter this, and give a better measure of the actual responsiveness for those types of pages, we ignore one highest interaction for every 50 interactions. Because the vast majority of page experiences don't have over 50 interactions, the browser almost always still reports the worst interaction. The 75th percentile of all page views is then reported as usual, which further removes outliers to give a value more representative of the general user experience.

An *interaction* is a group of event handlers that fire during the same logical user gesture. For example, "tap" interactions on a touchscreen device include multiple events, such as `pointerup`, `pointerdown`, and `click`. An interaction can be driven by JavaScript, CSS, built-in browser controls like form elements, or a combination of these things.

An interaction's latency consists of the single longest duration

(<https://w3c.github.io/event-timing/#ref-for-dom-performanceentry-duration%E2%91%A1:%7E:text=The%20Event%20Timing%20API%20exposes%20a%20duration%20value>)

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

HIDE

Key point: For more details on how INP is measured, see "[What's in an interaction?](#)" (#whats_in_an_interaction).

What is a good INP score?

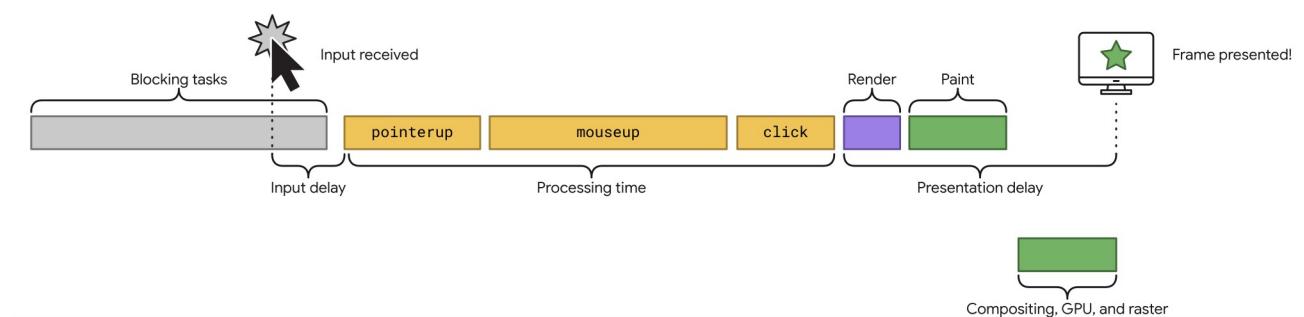
To ensure you're delivering user experiences with good responsiveness, a good threshold to measure is the **75th percentile** of page loads recorded in the field, segmented across mobile and desktop devices:

- An INP equal to or less than **200 milliseconds** means your page has **good responsiveness**.
- An INP between **200 milliseconds** and **500 milliseconds** means your page's responsiveness **needs improvement**.
- An INP greater than **500 milliseconds** means your page has **poor responsiveness**.



Good INP values are 200 ms or less. Poor values are greater than 500 ms.

What's in an interaction?



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

HIDE

interactivity through controls *not* powered by JavaScript, such as checkboxes, radio buttons, and controls powered by CSS.

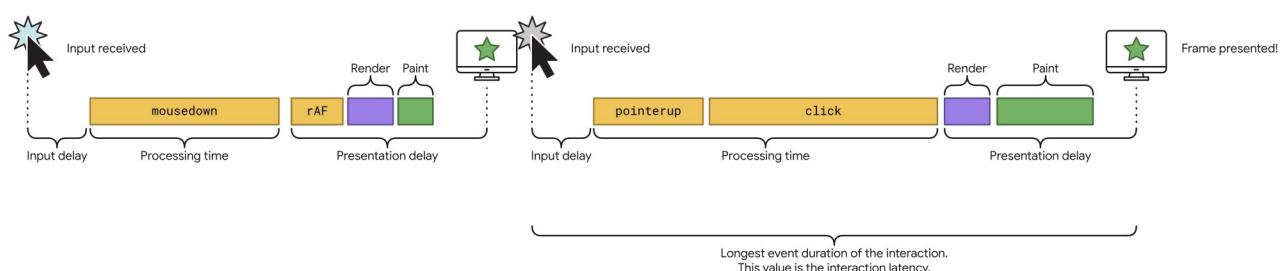
For the purposes of INP, **only the following interaction types are observed:**

- Clicking with a mouse.
- Tapping on a device with a touchscreen.
- Pressing a key on either a physical or onscreen keyboard.

Key point: Hovering and scrolling does not factor into INP. However, scrolling with the keyboard (space bar, page up, page down, and so forth) involves a keystroke, which can trigger other events that INP *does* measure. Any resulting scrolling isn't factored into the INP calculation.

Interactions happen in the main document or in iframes embedded in the document—for example, clicking play on an embedded video. Because end users aren't aware of which parts of the page are in iframes, you need to measure INP within iframes to accurately measure it for the whole page. However, JavaScript Web APIs don't have access to the iframe contents, and might not be able to measure INP within an iframe. This shows as a [difference between CrUX and RUM](#) (/articles/crux-and-rum-differences#iframes).

Interactions can consist of multiple events. For example, a keystroke includes the `keydown`, `keypress`, and `keyup` events, and tap interactions include `pointerup` and `pointerdown` events. The event with the longest duration within the interaction is chosen as the interaction's latency.



When the user presses the mouse button to click, the browser handles the interaction and presents a frame of visual feedback. When the user releases the mouse button, another series of event handlers runs before the browser can present another frame.

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

HIDE

How is INP different from First Input Delay (FID)?

INP is the successor metric to First Input Delay (FID). While both are responsiveness metrics, FID only measured the input delay of the *first* interaction on a page. INP improves on FID by considering *all* page interactions, beginning from the input delay, to the time it takes to run event handlers, and finally up until the browser has painted the next frame.

These differences mean that both INP and FID are different types of responsiveness metrics. Where FID was a load responsiveness metric

(/articles/user-centric-performance-metrics#types_of_metrics) designed to assess the page's first impression on the user, INP is a more reliable indicator of overall responsiveness, regardless of when in the life of a page interactions occur.

What if no INP value is reported?

It's possible for a page to return no INP value. This can happen for a number of reasons, including the following:

- The page loaded, but the user never interacted with it.
- The page loaded, but the user interacted with it using gestures that aren't measured, such as scrolling or hovering over elements.
- The page is being accessed by a bot, such as a search crawler or headless browser, that hasn't been scripted to interact with the page.

How to measure INP

INP can be measured both in the field (/articles/lab-and-field-data-differences#field_data) and in the lab (/articles/lab-and-field-data-differences#lab_data) using a variety of tools.

Key point: The best way to measure your website's INP is by gathering metrics from actual users in the field. If you're accustomed to relying on lab data for assessing performance, we recommend reading Why lab and field data can be different (and what to do about it) (/articles/lab-and-field-data-differences).

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

HIDE

interaction occurred during or after page load, the type of interaction (click, keypress, or tap), and other valuable information.

Read to learn more: [Find slow interactions in the field](/articles/find-slow-interactions-in-the-field) (/articles/find-slow-interactions-in-the-field).

If your website qualifies for inclusion in the [Chrome User Experience Report \(CrUX\)](#) (<https://developer.chrome.com/docs/crux>), you can quickly get INP field data through [CrUX in PageSpeed Insights](#) (/articles/find-slow-interactions-in-the-field#get_field_data_quickly_with_crux), alongside data on other Core Web Vitals. At a minimum, you can get an origin-level picture of your website's INP, but in some cases, you can also get page-level data.

However, while CrUX can tell you that there *is* a problem at a high level, it often doesn't provide enough detail to help fully understand what the problem is. A RUM solution can help you uncover more details about the pages, users, or user interactions that experience slow interactions. Being able to attribute INP to individual interactions helps avoid guesswork and wasted effort.

In the lab

Optimally, you'll want to start testing in the lab after you have field data that suggests you have slow interactions. In the absence of field data, however, there are some strategies for reproducing slow interactions in the lab. Strategies include following common user flows and testing interactions along the way, as well as interacting with the page during load, when the main thread is often busiest, in order to expose slow interactions during that crucial part of the user experience.

Read to learn more: [Manually diagnose slow interactions in the lab](#) (/articles/manually-diagnose-slow-interactions-in-the-lab).

How to improve INP

See our [collection of guides on optimizing INP](#) (/articles/optimize-inp) for guidance on identifying slow interactions in the field and using lab data to optimize them.

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

HIDE

the definitions of the metrics themselves. As a result, changes must sometimes be made, and these changes can show up as improvements or regressions in your internal reports and dashboards.

To help you manage this, all changes to either the implementation or definition of these metrics are surfaced in this [Changelog](http://bit.ly/chrome-speed-metrics-changelog) (<http://bit.ly/chrome-speed-metrics-changelog>).

If you have feedback for these metrics, provide it in the [web-vitals-feedback Google group](https://groups.google.com/g/web-vitals-feedback) (<https://groups.google.com/g/web-vitals-feedback>).

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 2024-03-12 UTC.

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

HIDE