User-centric Performance Metrics

Older Metrics (< 2019)

Metric	Description	Remarks
First Paint (FP)	Time when the browser first renders after navigation	Only tells users that the server has started to respond, nothing else.
First Contentful Paint (FCP)	Time when browser first renders any content from the document object model (DOM)	An early measure that doesn't capture page reactivity and user experience in later stages (e.g, scrolling while the page is loading). Even if FCP is really short, it won't matter to the users if their desired content takes longer to load.
DOMContentLoaded	Time when the initial HTML document has been completely loaded without waiting for images, and subframes to finish loading.	Doesn't capture the interactivity of the page. For some sites, users may only be interested in images for which this metric is not suitable. Moreover, it doesn't necessarily correspond to what users see on their screen.
Onload	Time when the page and its dependent resources have finished loading	Doesn't capture user experience and page interactivity while the page is loading. Most users start interacting with pages while it's loading.
First Input Delay (FID)	Time from when a user first interacts with your site (i.e. click a link, tap a button) to the time when the browser is actually able to respond to that interaction.	Captures "initial impression" of a page but varies depending on the page element user decides to interact with first. If the user's desired element provides a slow response, users might close the webpage and not engage with it even if other elements provide fast responses with less FID. Moreover, scrolling and zooming experience are not included in this metric.
Max Potential FID (mFID)	Measures the worst-case First Input Delay that your users might experience	Captures the worst possible FID on a page. However, many users might not experience the worst responsiveness depending on the element of their interest in the website that they like to interact with. It only gives an upper bound of page responsiveness.
Time to First Byte (TTFB)	Duration from the user or client making an HTTP request to the first byte of the page being received by the client's browser.	Indicates responsiveness of a web server or network resources, doesn't capture responsiveness of the page itself and its interactivity. Not really captures much user experience.
Time to Interactive (TTI)	Time from when the page starts loading to when it's visually rendered, its initial scripts have loaded, and it's capable of reliably responding to user input quickly.	TTI mostly makes sense in a "lab" scenario (using tools to simulate a page load in a consistent, controlled environment). In real-world scenarios , users interact with pages randomly as they load so that messes with the metric. Moreover, users also have varying internet speeds so TTI would vary.

Speed Index (SI)	It is a page load performance metric that measures how quickly content is visually displayed during page load. So, unlike other metrics, It is not another milestone on the timeline of web page's loading.	SI does not give you any information about the user's ability to interact with the interface of a webpage. It only takes into account directly visible parts of the page because of which it is highly dependent on the viewport size. Moreover,it is complex and hard to explain.

New Metrics (> 2019)

Metric	Description	Thoughts
Cumulative Layout Shift (CLS)	Measures the sum of the individual layout shift scores for each unexpected layout shift that occurs during the page load. Unexpected movement of page content usually happens because resources are loaded asynchronously or DOM elements get dynamically added to the page above existing content.	Captures abrupt changes on the page while its loading due to which users accidently click on unintended elements. It won't be helpful for AMP pages since they size all resources statically and thus scale and format properly while loading.
Total Blocking Time (TBT)	Measures the total amount of time between FCP and TTI where the main thread was blocked for long enough to prevent input responsiveness.	Doesn't measure page performance and interactivity after it has become responsive (e.g, slow scrolling)
Largest Contentful Paint (LCP)	Measures the render time of the largest content element visible in the viewport.	It marks the point in the page load timeline when the page's main content has likely loaded. However, it assumes that the largest content is the main content of the page which users are interested in. Users may be interested in a smaller element instead.

AMP (Accelerated Mobile Pages)

- AMP is an open-source framework for creating fast-loading mobile web pages.
- AMP pages are lightweight HTML pages designed to load quickly on mobile devices.
- Brands that opt for AMP can also leverage Google's content delivery network (CDN) for delivery of pages and images.
- AMP pages are discovered by search engines and other platforms through the HTML tag. You can choose to have both a non-AMP version and an AMP version of your page, or just an AMP version.
- **Design Goal:** Predictable performance by reducing the time until the content of a page can be consumed / used by the user. This means that:
  - HTTP requests necessary to render and fully layout the document should be minimized.
  - Resources such as images or ads should only be downloaded if they are likely to be seen by the user.
  - Browsers should be able to calculate the space needed by every resource on the page without fetching that resource.
- It achieves reliable performance by restricting some parts of HTML, CSS and JavaScript. To make up for those limitations AMP HTML defines a set of custom elements for content beyond basic HTML.
- It does not require the development of new rendering engines: existing user agents can render AMP HTML just like all other HTML.
- **AMP HTML Custom Elements:** Though most tags in an AMP HTML page are regular HTML tags, some HTML tags are replaced with AMP-specific tags. AMP HTML uses a set of contributed but centrally managed and hosted custom elements to implement advanced functionality (such as image galleries) that might be found in an AMP HTML document. While it does allow styling the document using custom CSS, it does not allow author written JavaScript beyond what is provided through the custom elements to reach its performance goals.
- **AMP Runtime:** It is a piece of JavaScript that runs inside every AMP document. It provides implementations for AMP custom elements, manages resource loading and prioritization and optionally includes a runtime validator for AMP HTML for use during development.
- **Statically sized resources:** AMP JavaScript library determines the dimensions of every page element before the browser renders them. Pages thus scale and format properly while loading, unlike with HTML, where the page layout is in flux as the page loads. External resources such as images, ads or iframes must state their size in the HTML so that AMP can determine each element’s size and position before resources are downloaded. AMP loads the layout of the page without waiting for any resources to download.
- **Asynchronous JavaScript**: To keep JavaScript from delaying page rendering, AMP allows only asynchronous JavaScript.
- **Prevent extensions from blocking rendering:** AMP doesn’t let extension mechanisms block page rendering. AMP supports extensions for things like lightboxes, instagram embeds, tweets, etc. While these require additional HTTP requests, those requests do not block page layout and rendering.
- **Restrict Third-party JavaScript:** Third-party JS likes to use synchronous JS loading. AMP pages allow third-party JavaScript but only in sandboxed iframes. By restricting them to iframes, they can’t block the execution of the main page. Even if they trigger multiple style re-calculations, their tiny iframes have very little DOM.
- **Inline and size-bound CSS:** CSS blocks all rendering, it blocks page load, and it tends to get bloated. In AMP HTML pages, only inline styles are allowed.
- **Prioritize resource loading:** AMP controls all resource downloads. It prioritizes resource loading, loading only what’s needed, and prefetches lazy-loaded resources.
- **Minimize style recalculations**: In AMP pages, all DOM reads happen first before all the writes. This ensures there’s a maximum of only one recalculation of styles per frame.