

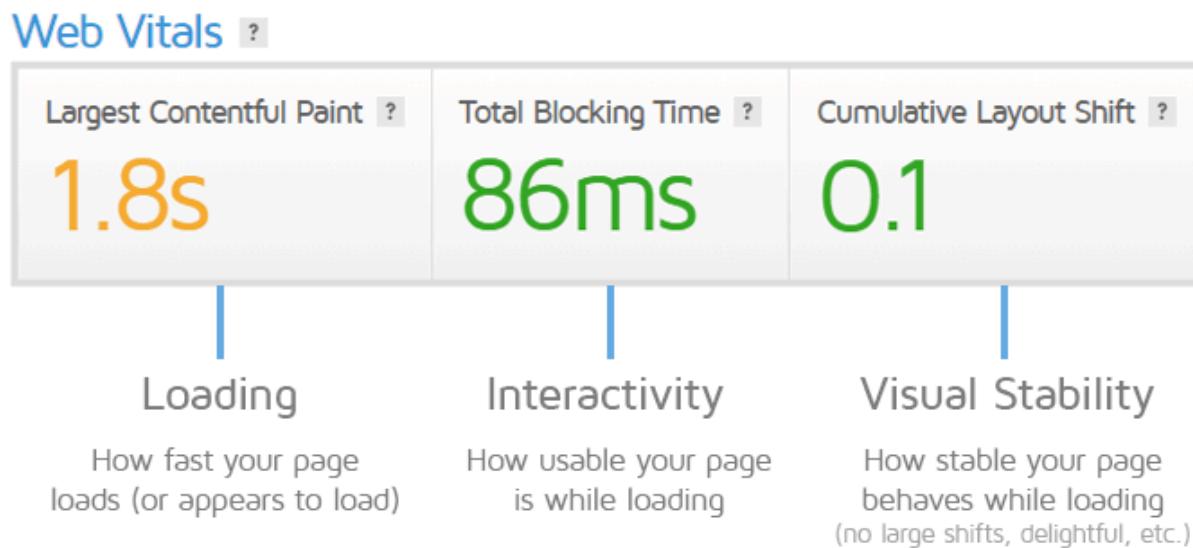
Web Optimization Metrics:

Web Vitals:

Google introduced a new concept in **2020** called "**Web Vitals**", which focuses on a small set of key metrics to assess your page experience.

Each metric represents a key facet of the page experience, namely loading, interactivity, and visual stability and includes the following metrics:

1. Largest Contentful Paint (**LCP**)
2. Total Blocking Time (**TBT**) or First Input Delay (**FID**)
 - a. Google page insight uses **FID** and GTMetrix uses **TBT**
3. Cumulative Layout Shift (**CLS**)



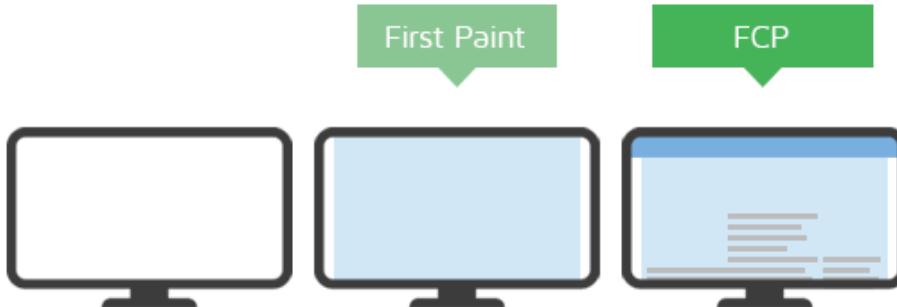
Web Vitals essentially measure your page for 3 key criteria that define a fast performing page.

Web Vitals' importance is further emphasized by the fact that they make up 80% of the total Performance Score calculation.

First Contentful Paint (FCP)

FCP is a **user-centric metric** for measuring perceived performance because it marks the point where visitors can first see any **consumable content** on the screen.

A fast FCP reassures the user that something is happening during your page load; a slow FCP implies that it's not doing anything, and thus is slow.



FCP measures how fast the first content element loads on your page

The faster your page renders consumable content (like text), the better the perceived performance your visitor experiences.

- Good - nothing to do here = FCP time of 934 milliseconds or less.
- OK, but consider improvement = FCP time between 934 and 1205 milliseconds.
- Longer than recommended = FCP time between 1205 and 1600 milliseconds.
- Much longer than recommended = FCP time higher than 1600 milliseconds.

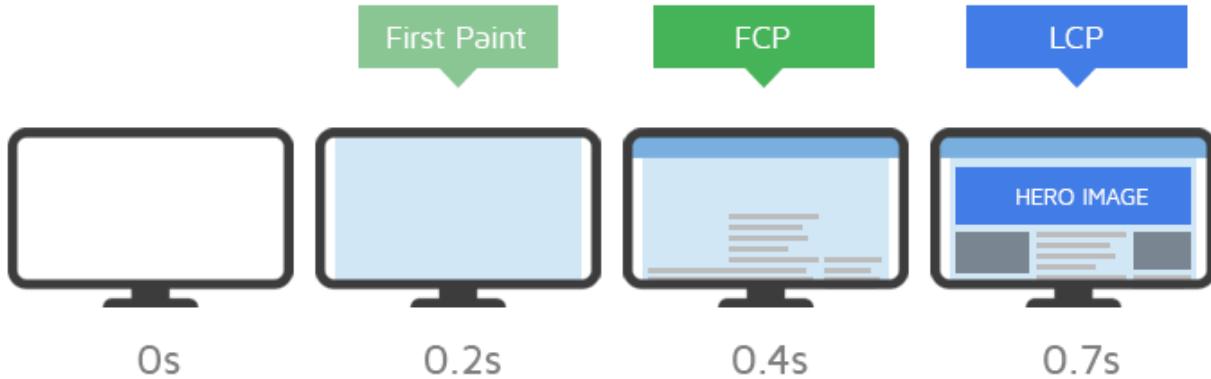
How to improve First Contentful Paint (FCP):

1. Reducing server response time:
 - a. Reducing initial Server Response Time / Time to First Byte (TTFB)
 - i. Optimizing your application code (including database queries).
 - ii. Implementing server-side caching.
 - iii. Upgrading server hardware for more CPU or memory resources.
 - b. Using a Content Delivery Network (CDN) for resources (eg: images, js, css)
 - c. Serving static assets with an efficient cache policy. ([NGINX](#))
 - d. Avoiding multiple page redirects
 - e. Establishing early connections to important third-party origins
2. Eliminating render-blocking resources:
 - a. Resources example: styles and script at the <head> of the html. Visitors will see a blank screen while render-blocking resources are processed.
3. Minimizing request counts and payload sizes:
 - a. Avoiding multiple page redirects.
 - b. Use SVG icons instead of using images as background.
 - c. Removing unused styles/script or unwanted third-party scripts.
 - d. Compressing your text resources. (Enable **GZIP** compression in server).
 - e. Using HTTP/2 or HTTP/3 instead of HTTP (Configure in server).

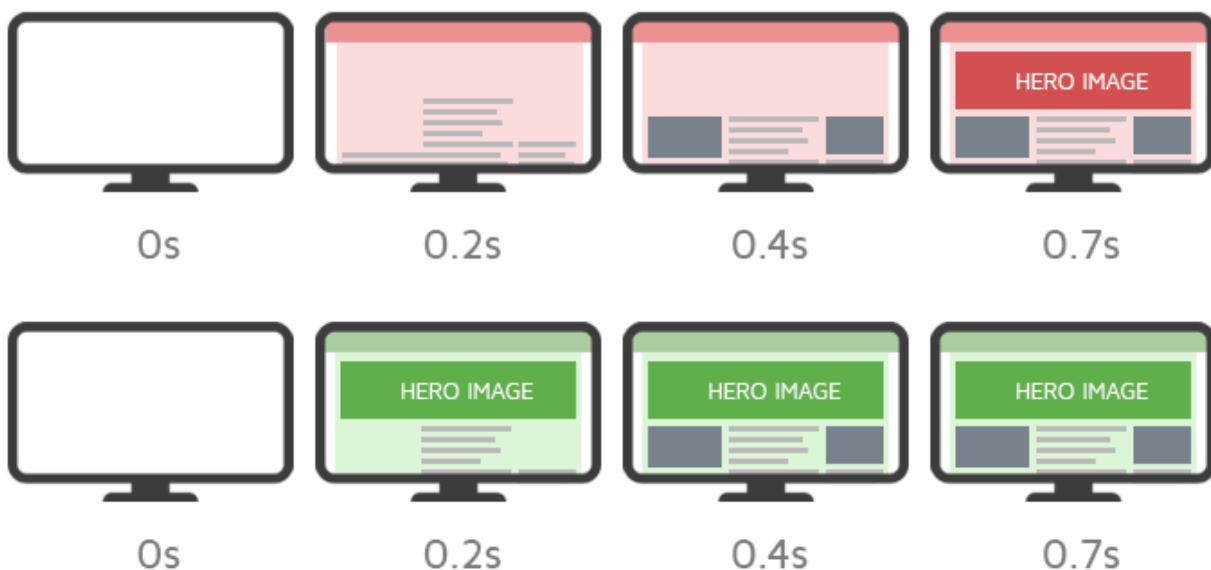
Largest Contentful Paint (LCP)

LCP is a **user-centric metric** that measures how quickly the largest **above-the-fold** content element is painted onto your page, like a carousel or hero image.

A faster LCP gives the impression to the visitors that your page loaded faster.



LCP measures how fast the largest content element loads on your page.



The green website looks more visually complete at 0.2s than the red website.

What this means is, optimizing for LCP can make the biggest improvement in web performance for the site, both, in terms of **Performance Score** and your visitors' **perception** of performance.

- **Good - nothing to do here** = LCP of 1200 milliseconds or less.
- **OK, but consider improvement** = LCP between 1200 and 1666 milliseconds.
- **Longer than recommended** = LCP between 1666 and 2400 milliseconds.
- **Much longer than recommended** = LCP higher than 2400 milliseconds.

How to improve Largest Contentful Paint (LCP)

1. Reducing server response time:
 - a. Reducing initial Server Response Time / Time to First Byte (TTFB)
 - i. Optimizing your application code (including database queries)
 - ii. Implementing server-side caching.
 - iii. Upgrading server hardware for more CPU or memory resources
 - b. Using a Content Delivery Network (CDN) for resources (eg: images, js, css)

- c. Serving static assets with an efficient cache policy ([NGINX](#))
 - d. Avoiding multiple page redirects
 - e. Establishing early connections to important third-party origins
2. Eliminating render-blocking resources:
- a. Resources example: styles and script at the <head> of the html. Visitors will see a blank screen while render-blocking resources are processed.
3. Optimizing images and videos:
- a. Properly size images: Images served on your page should be appropriately sized based on the dimensions they will be displayed at.
 - b. Efficiently encoding images: Use [tinypng](#) for easy lossless compression of images.
 - c. Serve images in next-gen formats: **WebP** (recommended), **JPEG 2000**, or **JPEG XR**
 - d. Do not use gifs images for animated content. Instead convert GIFs to videos using tools like [ezgif](#), [FFmpeg](#).

Cumulative Layout Shift (CLS)

CLS is a **user-centric metric** for measuring visual stability because it helps quantify how often users experience unexpected layout shifts - a low CLS helps ensure that the page is delightful.

Reducing CLS is crucial as pages that move around can result in a negative **user experience** (particularly on mobile devices).



Good CLS



It is important to note that CLS is a score - not a timing in milliseconds or seconds.

- **Good - nothing to do here** = CLS of 0.1 or less.
- **OK, but consider improvement** = CLS between 0.1 and 0.15.
- **Longer than recommended** = CLS between 0.15 and 0.25.
- **Much longer than recommended** = CLS of 0.25 or higher .

How to improve Cumulative Layout Shift

1. Specifying image dimensions:
 - a. Always specify, both, width and height for your website's image and video elements so that the correct spacing is used for images/videos.
2. Preventing the **Flash of Invisible Text (FOIT)**
 - a. Preloading web fonts
 - b. Using font-display
 - c. For Google web fonts, we can add font-display functionality by adding the "display=swap" parameter at the end of the web font URL

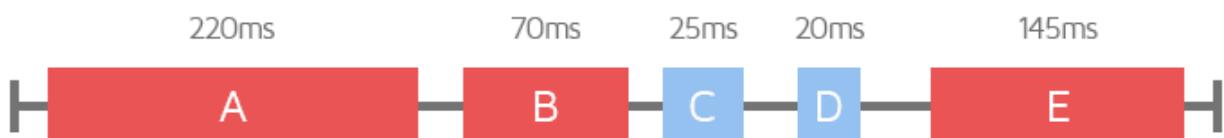
```
<link href="googlewebfonturl&display=swap" rel="stylesheet">
```

Total Blocking Time (TBT)

Total Blocking Time (**TBT**) measures the total time between First Contentful Paint (**FCP**) and Time to Interactive (**TTI**) where the main-thread was blocked for long enough to prevent input responsiveness.

TTI measures the earliest time after First Contentful Paint (**FCP**) when the page is reliably ready for user interactivity.

Browser main-thread timeline (task durations)



Any task longer than 50 ms is considered a Long Task

In the above example, there are 5 tasks on the main-thread, 3 of which are Long Tasks, as their individual durations exceed 50 ms. The blocking times for each Long Task are, as follows:

- Task A - 170 ms
- Task B - 20 ms
- Task E - 95 ms

The Total Blocking Time (TBT), in this scenario, is **285ms**.

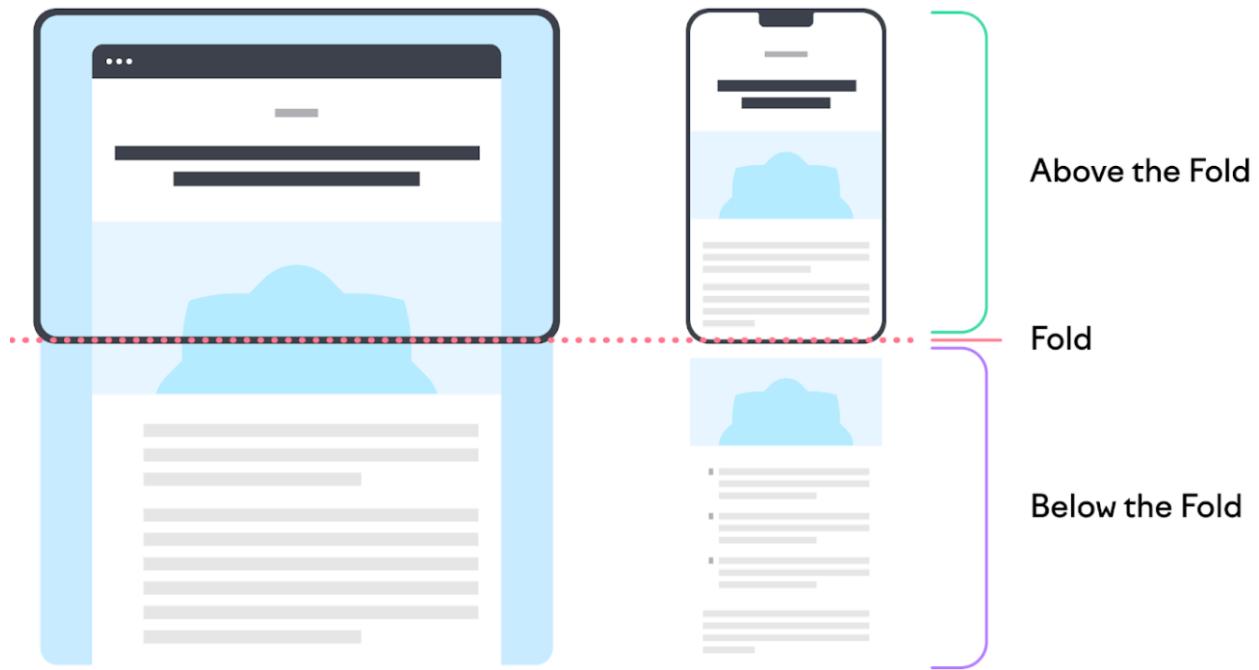
How to improve Total Blocking Time?

Total Blocking Time is highly correlated with JavaScript performance, and any improvements to JavaScript execution (in general, optimizations that improve TTI) are likely to reduce TBT.

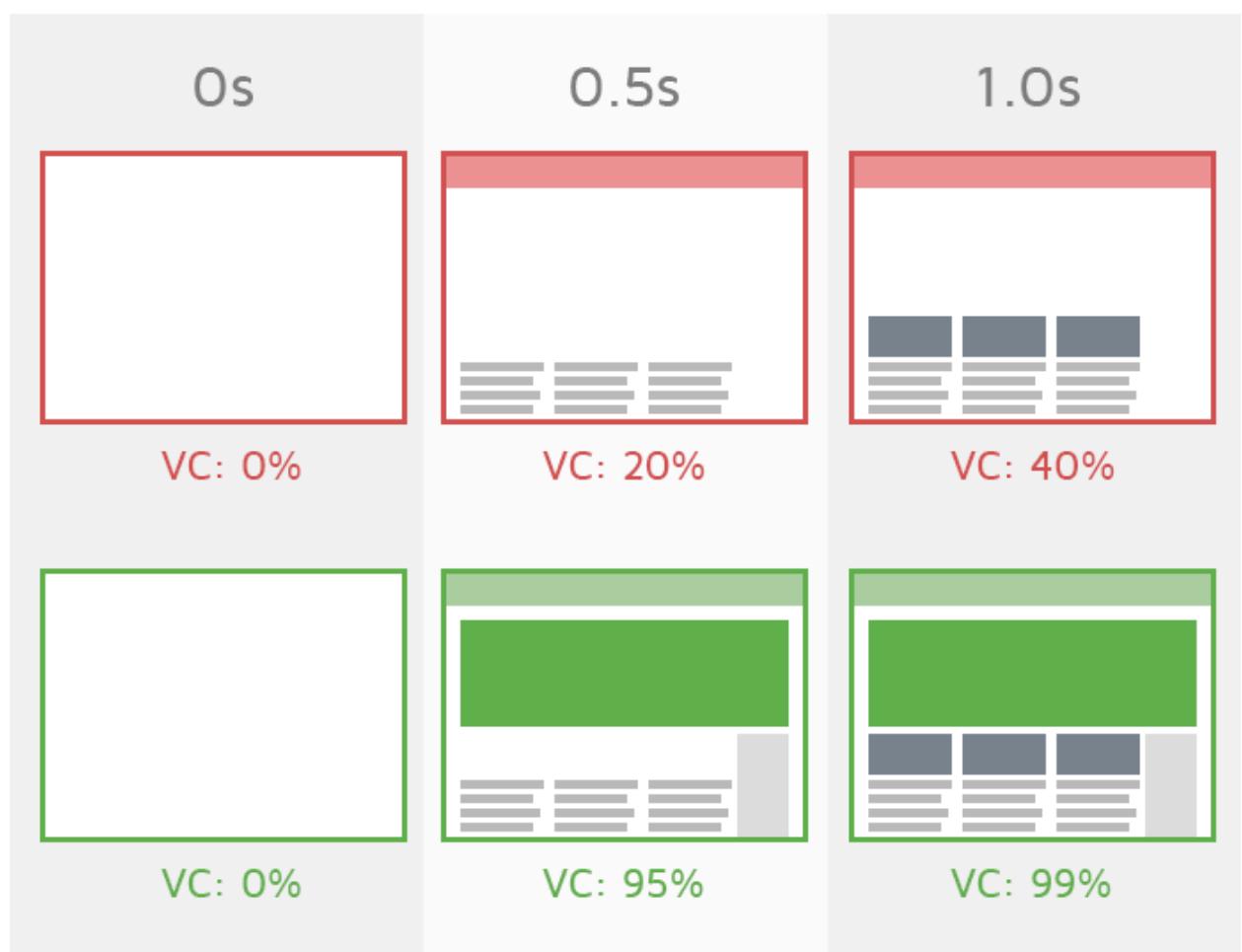
1. Reducing JavaScript execution time
 - a. Code-splitting
 - b. Removing unused code
 - c. Async / Defer scripts
 - d. Minifying and compressing JavaScript code
2. Minimizing main-thread work
 - a. Optimizing third-party JavaScript
 - b. Debouncing your input handlers (Use passive listeners to improve scrolling performance)
`document.addEventListener('touchstart', onTouchStart, {passive: true});`
 - c. Reducing CSS parsing time (by minifying, or deferring non-critical CSS, or removing unused CSS.)
3. Removing unused JavaScript
 - a. Dead code elimination
 - b. Dead imported code elimination
4. Reducing the impact of third-party code
5. Replacing large JavaScript libraries with smaller alternatives

Speed Index (SI)

Speed Index (SI) is a performance metric that measures how quickly your page is visually complete above-the-fold.



How Speed Index is calculated



Online Reference:

- [GTMetrix.com](https://gtmetrix.com)
- [Web.dev](https://web.dev)