CWV – What is First Contentful Paint (FCP)?

Last Modified on 10/21/2021 11:53 am EDT

The First Contentful Paint (FCP) metric measures the amount of time between the moment the user accesses the website and the moment they see something appear in their browser. It's one of the crucial elements of website performance and is also a big part of another Core Web Vital (CWV) metric, Largest Contentful Paint (LCP).

FCP addresses the user's need to perceive that the website is fast. This perception is satisfied when the first content appears on the screen. The content can be text, images, graphic elements (

What is a good FCP score?

FCP is measured in seconds, with the optimal time between zero and 1 second. The lower your FCP, the faster users see essential content.

- Great = less than 1 second
- Needs Improvement = between 1 and 3 seconds
- Poor = more than 3 seconds

Use **Freestar Analytics** to determine the FCP score of your website's pages. You can also determine the pattern of your score over time.

To dig deeper into your FCP score, use Lighthouse or PageSpeed Insights. Lighthouse's scoring is based on real website performance data, which are converted to percentiles using a logarithmic scale.

What contributes to a poor FCP score?

Anything that prevents a website from quickly displaying content leads to a lower FCP score. Such slowdowns could be caused by:

- A poorly performing server which leads to longer response times.
- An abundance of render-blocking resources such as JavaScript and CSS. Since these files are usually loaded at the top of a page, the impacts on FCP are greater.
- An improperly configured cache causes returning visitors to request content that has not changed since their previous visit.
- Lack of or inefficient compression.

How can you improve FCP on your site?

Improving FCP on a site requires reducing the number of requests the browser makes to the server, reducing the amount of data that the browser fetches, or improving server response time.

Some tactics include:

- **Keep slow rendering below the fold**. Make the top of your site a fast as possible.
- **Change your hosting provider** or **choose dedicated hosting** instead of shared hosting. Dedicated hosts run much faster than shared hosting services because your site is not sharing server resources with other sites.
- **Reduce the number of render-blocking resources** such as scripts (JavaScript) and stylesheets (CSS). Every call to a CSS or JavaScript file slows down your site. When possible, combine all JavaScript and CSS files into a single file. This reduces the number of calls to the server because one call is sufficient to get all the required functionality.
- **Reduce the size of CSS and JavaScript files**. These files often contain comments and spacing to make it easier to read for a human. However, the browser does not require this information. By removing as many extra characters from stylesheets and scripts as possible, it makes the site faster. You can achieve this using online tools like https://www.minifier.org/ or development tools.
- **Remove any unused CSS or JavaScript**. Third-party software often includes JavaScript and stylesheets designed to support as many different types of websites as possible. However, if you're only using a portion of that code, try to eliminate the unused portions to speed up the site. Use a coverage tool (like those included in Chrome) to determine which parts of the code are used.
- **Avoid multiple page redirects**. Each redirect is a new request to the server and adds time to complete the page's loading and rendering.
- **Preload critical requests where possible**. In the setup of your page, tell the browser that it needs to load multiple assets (usually CSS and JavaScript files) to render the page. Telling the browser to fetch those assets before rendering reduces the number of requests back to the server.
- **Make use of an efficient CDN** (content delivery network) to reduce the time to retrieve data from your server.
- **Avoid payloads that exceed 5,000 KB** (5 MB). This also benefits users who pay for cellular data.
- **Serve static assets with an efficient cache policy**. Efficient caching speeds up your site for returning users by caching the data locally. This removes the need to request data from the server.
- **Avoid DOM trees** (the graph representation of a webpage) **with more than 1500 nodes**, a **depth greater than 32 nodes**, or a **parent node with more than 60 nodes**. You can

achieve this by breaking a page into multiple smaller pages.
- **Use compression for webpage transfers**. At a minimum, configure the server to use gzip. If possible, use Brotli, which provides better speed and compression.
- If possible, **preload web fonts** since some browsers do not display the page unless all fonts are already loaded.

To determine your site's FCP score:

- Use Lighthouse
- Use PageSpeed Insights
- Use Google's Search Console Speed report
- Use WebPageTest
- Use Chrome DevTools

Where can I find more information?

For more information to help improve your FCP score, visit the web.dev site.

- https://web.dev/fcp/
- https://web.dev/first-contentful-paint/

For information on other Core Web Vitals (CWV), see Freestar's other CWV articles:

- CWV – What is First Input Delay (FID)?
- CWV – What is Largest Contentful Paint (LCP)?
- CWV – What is Cumulative Layout Shift (CLS)?

For industry insights and information about our product offerings, check out our blog!

Want to see our products in action? For a demo, fill out a form here.