Live Core Web Vitals Measurements

Core Web Vitals give developers and performance specialist with different facet of user experience. These core web vitals metrics can be measured in the wild and reflect real-world users experience with your content.

The current metrics in the Core Web Vitals palette are:

- Largest Contentful Paint (LCP): measures loading performance. To provide a good user experience, LCP should occur within 2.5 seconds of when the page first starts loading.
- First Input Delay (FID): measures interactivity. To provide a good user experience, pages should have a FID of 100 milliseconds or less.
- Cumulative Layout Shift (CLS): measures visual stability. To provide a good user experience, pages should maintain a CLS of 0.1. or less.

While it’s true that these performance metrics are applicable to all pages, it is not the same as to say that all tools will reflect data for all origins and web sites. Neither the HTTPArchive dataset and the Chrome User Experience Report guarantee that your origin (website) will appear in the results since your site may not meet the HTTPArchive’s popularity threshold or not enough people who have agreed to provide data to CRuX have visited your site.

One way to gather metrics that are specific to your site is to use the web vitals library and your own way to collect the data

Rather than rely on NPM and bundlers, we will work with the Unpkg CDN and its ability to deliver modules and `type="module"` scripts.

The first version is good while you’re doing local development is to log the results for each metric to the DevTools console.

This is a two-step process: First we import the functions from the Unpkg CDN and make sure we get the module version of the package.

We then call the functions and log the results to the console.

```
<script type="module">
  import {
    getCLS,
    getFID,
    getLCP
    } from 'https://unpkg.com/web-vitals?module';

  getCLS(console.log);
  getFID(console.log);
  getLCP(console.log);
</script>
```

This is good for local development but it’s not ideal for production. We have no way to aggregate the data and use it to get a better understanding of the siite’s pefromance.

If you’re using Google Analytics, you can send the Web Vitals metrics to your Analytics account to do further analysis.

The example assumes you’ve already set up and initialized your global `gtag()` function; see: [Add gtag.js to your site]() for more information on how to set it up.

Once `gtag` is set up we can use it to send the metrics to Google Analytics. To do so, we create a function that holds the data we want to send.

Google Analytics metrics must be integers, so the value is rounded. For CLS the value is first multiplied by 1000 for greater precision.

Then we use the function as the parametter for each metric instead of logging the results to the console.

```
<script type="module">
  import {
    getCLS,
    getFID,
    getLCP
    } from 'https://unpkg.com/web-vitals?module';
  function sendToGoogleAnalytics({name, delta, id}) {
    gtag('event', name, {
      event_category: 'Web Vitals',
      event_label: id,
      value: Math.round(name === 'CLS' ? delta * 1000 : delta),
      non_interaction: true,
    });
  }

  getCLS(sendToGoogleAnalytics);
  getFID(sendToGoogleAnalytics);
  getLCP(sendToGoogleAnalytics);
</script>
```

With this script you can gather data from people actually using your site rather than synthetic data.