

Chilled to the Core Web Vitals

Dave Smart, Tame the Bots

**CRAWL
ERROR**



GREETING, FIENDS & GHOULS

- I'm Dave Smart, pleased to eat meet you!
- I'm a technical SEO consultant & developer.
- I'm also a Google Platinum Product Expert.
- I'm here to tell you you don't need to be afraid of:

CORE WEB VITALS!

WHAT ARE CORE WEB VITALS?

Three “simple” metrics that attempt to quantify the experience a visitor has when using your website, in three key areas:

1. Speed
2. Interactivity
3. Stability

It’s an initiative, driven by Google, to get unified quality scores across the board.



LARGEST CONTENTFUL PAINT

= Speed



FIRST INPUT DELAY

= Interactivity



CUMULATIVE LAYOUT SHIFT

= Stability

LARGEST CONTENTFUL PAINT – SPEED



Largest Contentful Paint (LCP) is the time it takes for the main content is most likely visible on the page.

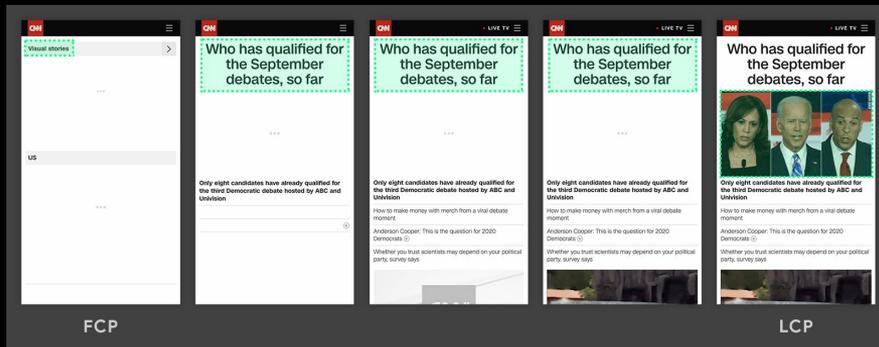
- The element must be in the viewport, all, or some of it should be in the first screen, but stuff outside the viewport is not counted.
- Not all elements are looked at currently the list is:
 - ``
 - `<image>` tags inside `<svg>`
 - `<video>` (the poster size is used)
 - Elements containing a text node

* The list of elements is subject to change.

LARGEST CONTENTFUL PAINT



As a page loads, different elements may become the LCP candidate:



As soon as the page is interacted with (touched, clicked, scrolled etc.) LCP 'stops' and the biggest score at that time is your LCP score.

LARGEST CONTENTFUL PAINT



Measured in seconds, these are the current target times:

- Good:
2.5 seconds and under
- Needs Improvement:
Between 2.5 seconds + and 4 seconds and under
- Poor:
Over 4 seconds

LARGEST CONTENTFUL PAINT



Why this metric?

It's a good proxy of the speed of the page, as your users perceive it.

The largest element in the viewport is mostly going to be the most eye-catching.

Former metrics weren't always great, FCP might be irrelevant content painted to the screen, FMP is hard to standardise and can be very different load to load. It's hard for a browser to decide what "meaningful" is.

FIRST INPUT DELAY = INTERACTIVITY



First Input Delay (FID) is a measure of how long it takes a browser to responding to an input after the user interacts with the page, such as

- Clicking a link
- Tapping a button
- Using a custom JavaScript control or other UX feature.

It's a measure of if a browser is too busy doing other stuff to listen to the users commands.

FIRST INPUT DELAY



It is not a measure of how long the events these controls trigger take to run. If clicking this:

```
<button onclick="release('the bats')">Vengeance</button>
```

The FID would be the time until the browser is ready to execute the release function, not the time the function took to set free the sky mice.



FID is measured at the point of first interaction, so the first tap or click on the screen.

FIRST INPUT DELAY



Measured in milliseconds, these are the current target times:

- Good:
100ms and under
- Needs Improvement:
Between 100ms + and 300ms and under
- Poor:
Over 300ms

FIRST INPUT DELAY



Why this Metric?

It's a good measure of how the user perceives the responsiveness of the site.

Many sites now are more than just text and images on a page, furiously tapping a button and nothing happening can lead to RAGE!



CUMULATIVE LAYOUT SHIFT = STABILITY



Cumulative Layout Shift (CLS) is a measure of how stable a page layout is.

- It's a measurement of how much of the viewport is affected by a shift, and how much that moves by.
- It's measured render frame by frame, although a shift can occur across more than 1 render frame.
- Cumulative means it adds up, each shift is added to the grand total of the page, this grand total is the CLS score

Unlike the other two metrics, CLS is reported for the whole page lifecycle

CUMULATIVE LAYOUT SHIFT



The formula to calculate the layout shift is:

layout shift score = impact fraction * distance fraction

Impact fraction is the union of the visible areas of elements that have moved, across the previous frame and the current one.

CUMULATIVE LAYOUT SHIFT



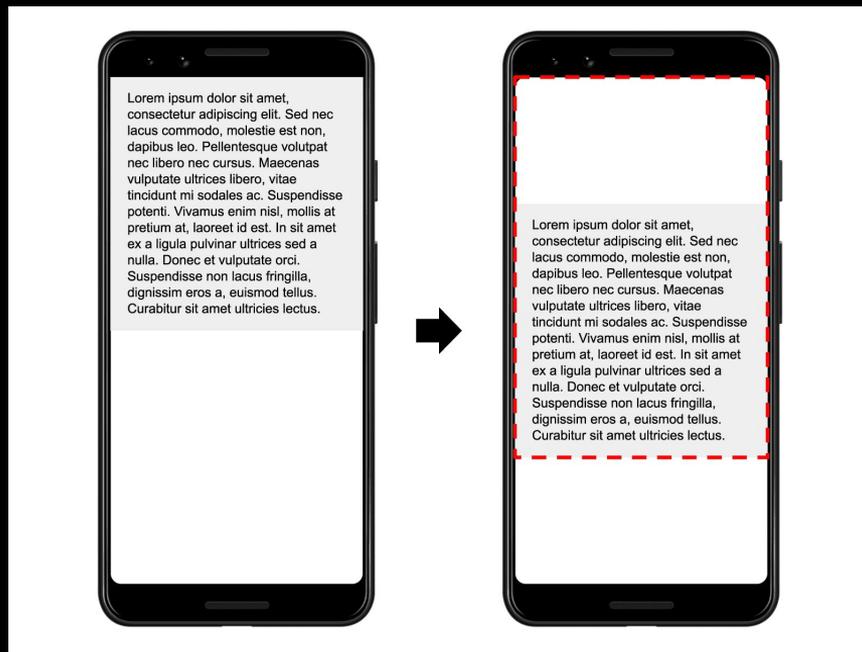
If an element that covered 50% of the screen, or pushed down from the top by 20% would mean that with the previous & current frame added together:

- 70% of the screen is affected
- So as a fraction it's 0.7

CUMULATIVE LAYOUT SHIFT



Visual example of impact fraction from <https://web.dev/cls>



CUMULATIVE LAYOUT SHIFT



The distance fraction is how far these elements moved in between the two frames, so in our example, it was 20% of the screen, so the distance fraction would be 0.2.

Using that calculation:

layout shift score = impact fraction * distance fraction

Becomes:

$$0.7 * 0.2 = 0.14$$

CUMULATIVE LAYOUT SHIFT



...And You Will Know Us by the Trail of ~~Dead~~ Shifted

A new element, like a div, or an image appearing isn't considered in CLS, rather the shift comes from the elements the newcomer displaces, be that down, left, right (or very rarely, up). The same is true for an element disappearing.

Some shifts are to be expected.

It's common and expected that things might move when you interact with a page, for that reason, any shift that occurs within .5 seconds of a click or a tap isn't added to the CLS score.

CUMULATIVE LAYOUT SHIFT



Why this Metric?

It's massively off putting when a user goes to click on a button or link, only to have that element move, so the user misses, or even clicks an action they didn't want to perform.



MEASURING YOUR VITALS



Before your site can become a living, breathing performance monster, you need to know how it performs. Fortunately the Web Vitals are all over the place!

The methods can be broken down into two types:

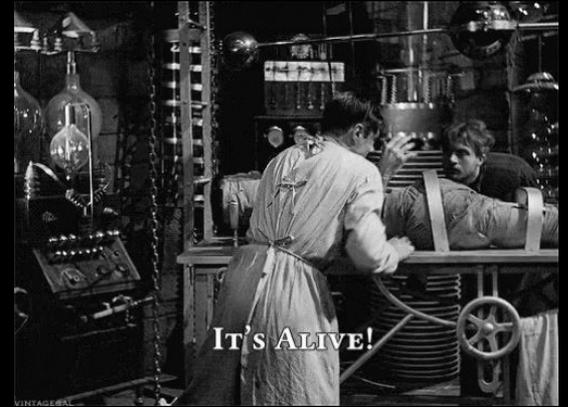
- Lab Test
- Real User Measurements

MEASURING – LAB TESTS

Lab tests are ‘simulated’ measurements, run on request that mimic the experience a user might get.

Examples are:

- Lighthouse
- PageSpeed Insights
- WebPageTest.org



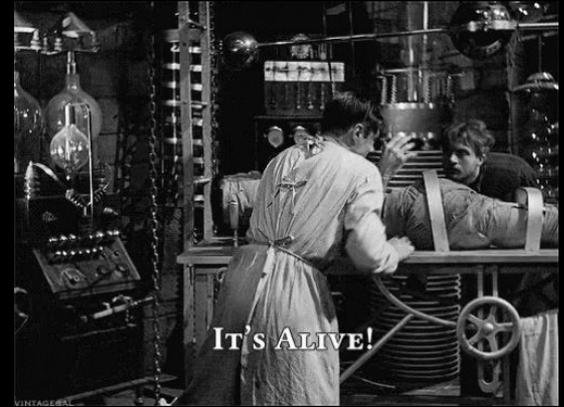
MEASURING – LAB TESTS

Pluses:

Lab tests are immediate, you can get the data there and then.

Downsides:

The connection speeds and device capabilities might not match your users, they only take the initial load into consideration, no FID



MEASURING – REAL USER MEASUREMENT

Real User Measurement (RUM) data are metrics collected from actual visitors to your site.

You can find RUM data in:

- Search Console (from CrUX)
- PageSpeed Insights field & origin scores (also from CrUX)
- Other CrUX data sources, like Data Studio.
- Collect your own.



MEASURING – REAL USER MEASUREMENT

Pluses:

This is actually what your users see!

Downsides:

RUM metrics are reactive, not proactive. They can take time to collect.

If relying on CrUX, a URL might not have enough data to show.



LAB OR RUM?



OR



?

Why not both? It actually should be both!

- Use lab tests to predict if changes you make will be positive or negative.
- Use real world measurement to see if the the changes WERE positive

MEASURING

To overcome the limitations of CrUX data collection rates, collect your own!

- Google have released a library called web-vitals:
<https://github.com/GoogleChrome/web-vitals>
- SALT.agency have created wordpress plugin to make this easier:
<https://wordpress.org/plugins/web-vitals/>
- Use the web-vitals chrome extension (search google) or the bookmarklet here:
<https://tamethebots.com/blog-n-bits/web-perf-iv-a-new-hope>
to spot check 'RUM' metrics, with the warning the user isn't quite real, it's you.
- Use Total Blocking Time (TBT) as a proxy for FID if using Lighthouse

IMPROVING YOUR VITALS

There is no one way to guarantee great scores for all sites, and slay your performance foes.

Each site is it's own movie script, but like those famous monsters of the silver screen, there are often some common tropes!



IMPROVING LARGEST CONTENTFUL PAINT

The first port of call is to make sure that the time to first byte is as good as it gets, this is the time it takes for a request for a resource to start being returned to the browser after it asks for it.

Look At:

- Optimising your servers performance
- Cache assets wherever possible
- Think about using a CDN to serve content from a location closer to the user
- Use pre-connect to help the browser fetch third party resources faster.

IMPROVING LARGEST CONTENTFUL PAINT

The second part is to make sure the browser can render the elements as quickly as possible.

Look At:

- Reducing render blocking CSS, by minifying, inlining CSS that's needed to build the layout in the initial viewport, and defer as much as the rest as you can.
- Reduce render blocking (and in fact) as much JavaScript as you can, minify it, serve only what you need and defer as much of that as you can.
- Make sure images and video are as well compressed and encoded as you can.

IMPROVING FIRST INPUT DELAY

The main cause of poor FID is overloading the main browser thread with heavy, long running JavaScript tasks.

Look At:

- Break long running tasks into smaller, shorter tasks and run them asynchronously.
- If you are processing large data sets in your scripts, make them as small as possible.
- If you really need large data sets, with a lot of post processing, move them off the main thread using web-workers.
- As with LCP, ship only the JavaScript you actually need for the page.

IMPROVING CUMULATIVE LAYOUT SHIFT

The main causes of CLS include lazy-loaded images and resources, and ads.

Look At:

- Including sizes on images with the height & width attributes. Modern browsers will use these to calculate aspect ratios automatically, so the space is reserved before they load in. This is still compatible with CSS methods to make images responsive.
- For ads, reserve a space for them before they load (i.e. load them in a div sized sufficiently to contain them)
- Only add new content above other elements on interaction.
- Use transform animations, and beware that some CSS properties changing can cause layout shift.

IMPROVING – FURTHER READING

Take a look at the guidance on the web.dev site:

- Optimise LCP
<https://web.dev/optimize-lcp/>
- Optimise FID
<https://web.dev/optimize-fid/>
- Optimise CLS
<https://web.dev/optimize-cls/>



BLOOPER REEL – SOME ODDITIES

There are a few oddities, gotchas and strange stuff you might want to look out for!

- **Single Page Applications:**
Because these, as far as the browser is concerned, are just one page, the metrics are a little confused. Metrics may still be reported, but assigned to the first page the user lands on. OK(ish) for LCP & FID, bad news for CLS
- Sometimes things aren't considered an interaction, like selecting an option in a `<select>` element, so if that causes something to move, CLS is affected, where a button click that does the same doesn't.



BLOOPER REEL – SOME ODDITIES

The APIs in are new, and changing, look at your site in Chrome Canary and see if there are differences in the way the metrics are measured, that might alter your scores for better or worse.

The chromium team are working hard to improve, stabilize and standardise things.

There is a wider push by W3C to standardise these tests further, and hopefully this will break the metrics out of the chromium world and into the whole web ecosystem.

SOME FINAL THOUGHTS

- Although many have discovered Core Web Vitals because of its future use as a ranking signal, performance is first a foremost a user satisfaction thing.
- Its 'power' as a ranking signal is likely to be quite small, and the focus is likely stay mostly on content relevance and quality. This does not make performance any less important, your users delight > arcane ranking signals.
- Which metric is most important depends on your site. FID is very important to an interactive web app, less important to a news article.
- Don't kill features users love just to meet scores. There's a balance to be found to give your users the best features, in the quickest, most responsive, stable way possible.

I've been Dave Smart, you can reach me at:

{Tame**the**Bots}

<https://tamethebots.com/>

@davewsmart on twitter

<https://twitter.com/davewsmart>



Fin.