Core Web Vitals Cheatsheet

TIP! In Dev Tools (in incognito), go to the Network tab and **change 'No throttling' to 'Slow 3G'** - this will give you a clearer picture of how your page loads on slower connections, how it shifts, and when your LCP occurs.

TIP! After adjusting the throttling setting, go to the Performance tab, **hit the Record button** and then reload the page. You will then see a timeline of your page load, with markers for when each CWV occurs.

LCP

Largest Contentful Paint

The time it takes your largest image* or text block in the viewport (above the fold) to load

Primarily affected by:

- Slow server response times
- Render blocking CSS
- Render blocking JavaScript
- Resource load times

TIP! PageSpeed Insights will show you your LCP element under the Diagnostics section

TIP! Detailed guides available on the ARC (search for the Master Page Speed guide)

Render blocking CSS

- Make sure your CSS is minified
- Inline Critical CSS in `<head>`
- Defer loading of other non-critical stylesheets (using new `media="print"` method)

Resource load times

- Optimise/compress image file sizes
- Use WebP images where possible
- If your LCP element is an image/video, `<preload>` it
- Ensure image's true dimensions aren't much bigger than the size they're being rendered at

important if your LCP element is an image

Render blocking JavaScript

- Make sure all JS is being minified
- Defer scripts where possible using `defer` attribute
- View Source and search for .js to see all scripts being included - can you remove any or replace with custom lightweight JS?
- Split any large chunks of page specific JS into separate files and only include on the pages that need them rather than site wide
- Lazy load all images and iframes below the fold
- **Don't** lazy load anything above the fold
- Don't use JS to inject any critical content (e.g. don't append hero videos with JS)

Slow server response times

- Install WP Super Cache WordPress plugin

***What elements are considered for the LCP?**

- `<img>` elements
- `<video>` elements (but the fallback poster image is used)
- Elements with a background image (unless the bg is a CSS gradient)
- Block-level elements containing only text nodes or other inline text elements children
- `<image>` elements inside an <svg> element
- This range of elements could be expanded in future (correct as of July 2021)

FID

First Input Delay

Time between user clicking on something (e.g. a button) and the page responding to this click

Primarily affected by:

- Third-party embeds/scripts
- Excessive/heavy JavaScript

TIP! FID is based on real life user interactions so is only available in the Field Data, but it correlates strongly with **Total Blocking Time** so you can use that as an indicator

TIP! Detailed guides available on the ARC (search for the Master Page Speed guide)

Third party embeds/scripts

- Look at 3rd party scripts in place - can you remove any? e.g. Mouseflow
- Lazy load all 3rd party embeds (videos, maps etc.)
- Lazy load inline Wistia videos – use their iframe embed code (don't include their EV-1.js script)
- Don't use Wistia's popover code - use iframe code and follow guide on ARC to display this in a custom modal
- Use Reviews.io API instead of their widgets - example in Modular components library (Reviews Carousel)

Excessive/heavy JavaScript

- View Source and search for .js to see all scripts being included - can you remove any or replace with custom lightweight JS?
- Defer scripts where possible using `defer` attribute
- Avoid Contact Form 7 V5.4+ (ideally use *HTML Forms* WP plugin)
- Are you using JS libraries/plugins for things like carousels, accordions, modals, tabs etc.? Could you write your own or see if someone else has? Our Modular components library also has custom lightweight solutions for some of these features
- Split any large chunks of page specific JS into separate files and only include on the pages that need them rather than site wide

CLS

Cumulative Layout Shift

How much the page unexpectedly shifts around during page load and after interactions

Primarily affected by:

- Images without size attributes
- Differences between web font and fallback font
- Dynamically injected content

TIP! PageSpeed Insights will show the elements that contribute the most to your CLS score under 'Avoid Large Layout Shifts' in the Diagnostics section

TIP! Detailed guides available on the ARC (search for the Master Page Speed guide)

Images without size attributes

- Add width & height attributes to your images (browsers will calculate an aspect ratio rather than use the fixed values)

  Caveat: won't work on images that have a different aspect ratio at different screen sizes e.g. a hero that's portrait on mobile and landscape on desktop

Differences between webfont and fallback font

- Ensure your fallback font is as close as possible to the webfont that replaces it
- If your webfont makes text drop onto two lines, you could reserve the space using **`min-height`**
- New ***CSS Font Descriptors*** are coming which will allow you to adjust the sizing of your fallback font to make it a closer match

Dynamically injected content

- Ensure any content injected with JS doesn't shift the page layout; e.g. use **`position:fixed`** to avoid layout shift
- If the dynamically injected content isn't fixed, then try and reserve the space it will occupy using **`min-height`**