

web vitals

retain users. convesation. user experience. LCP as KPI

light house score. metrics

**Lab data**

Lighthouse takes a URL and runs a series of audits against the page, generating a report on how well the page did. There are multiple ways to run Lighthouse, including an option to easily audit a page from within Chrome DevTools.

**Field data**

Chrome User Experience Report (CrUX) provides metrics showing how real-world Chrome users experience popular destinations on the web.

**Overview**

Web Vitals is an initiative by Google to provide unified guidance for quality signals that are essential to delivering a great user experience on the web.

**Core Web Vitals** are the subset of Web Vitals that apply to all web pages, should be measured by all site owners, and will be surfaced across all Google tools. Each of the Core Web Vitals represents a distinct facet of the user experience, is measurable in the field, and reflects the real-world experience of a critical user-centric outcome.

loading, interactivity, and visual stability

**[Largest Contentful Paint (LCP)](https://web.dev/articles/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)

**[Interaction to Next Paint (INP)](https://web.dev/articles/inp):** measures interactivity. To provide a good user experience, pages should have a INP of 200 milliseconds or less.

**[Cumulative Layout Shift (CLS)](https://web.dev/articles/cls):** measures visual stability. To provide a good user experience, pages should maintain a CLS of 0.1. or less.

**Field tools to measure Core Web Vitals**

**The Chrome User Experience Report** collects anonymized, real user measurement data for each Core Web Vital. This data enables site owners to quickly assess their performance without requiring them to manually instrument analytics on their pages, and powers tools like **PageSpeed Insights**, and Search Console's Core Web Vitals report.



**LCP**

Element timing api

LCP reports the render time of the largest image or text block visible in the viewport, relative to when the user first navigated to the page.

75th of all visits should be in 2.5 secs is good.

**INP**

INP is a metric that assesses a page's overall responsiveness to user interactions by observing the latency of all click, tap, and keyboard interactions that occur throughout the lifespan of a user's visit to a page. The final INP value is the longest interaction observed, ignoring outliers.



**INP is the successor metric to First Input Delay (FID).** While both are responsiveness metrics, FID only measured the input delay of the first interaction on a page. INP improves on FID by observing all interactions on a page, beginning from the input delay, to the time it takes to run event handlers, and finally up until the browser has painted the next frame.

CLS

CLS is a measure of the largest burst of layout shift scores for every unexpected layout shift that occurs during the entire lifecycle of a page.

A burst of layout shifts, known as a session window, is when one or more individual layout shifts occur in rapid succession with less than 1-second in between each shift and a maximum of 5 seconds for the total window duration.

Impact Fraction

percent of whole window

distance

假设有一个元素在页面加载时从顶部移动到中间：

元素移动前占据视窗顶部 25% 的区域。

元素移动后占据视窗中间 25% 的区域。

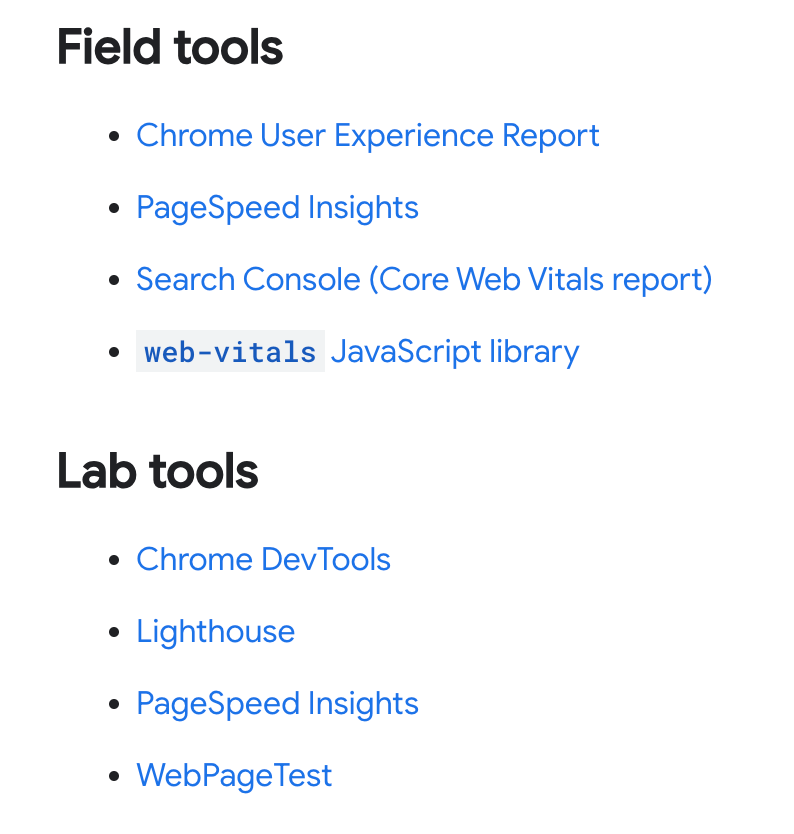
影响区域：元素在移动前后的包围盒交集面积是 25% 的视窗面积，因此影响区域比例是 0.25。

移动距离：元素从顶部移动到中间，移动了 50% 的视窗高度，因此移动距离比例是 0.50。

布局偏移分数：影响区域比例（0.25） × 移动距离比例（0.50） = 0.125。

Measurement

use Lighthouse or web-vitals



**Measure in JS**

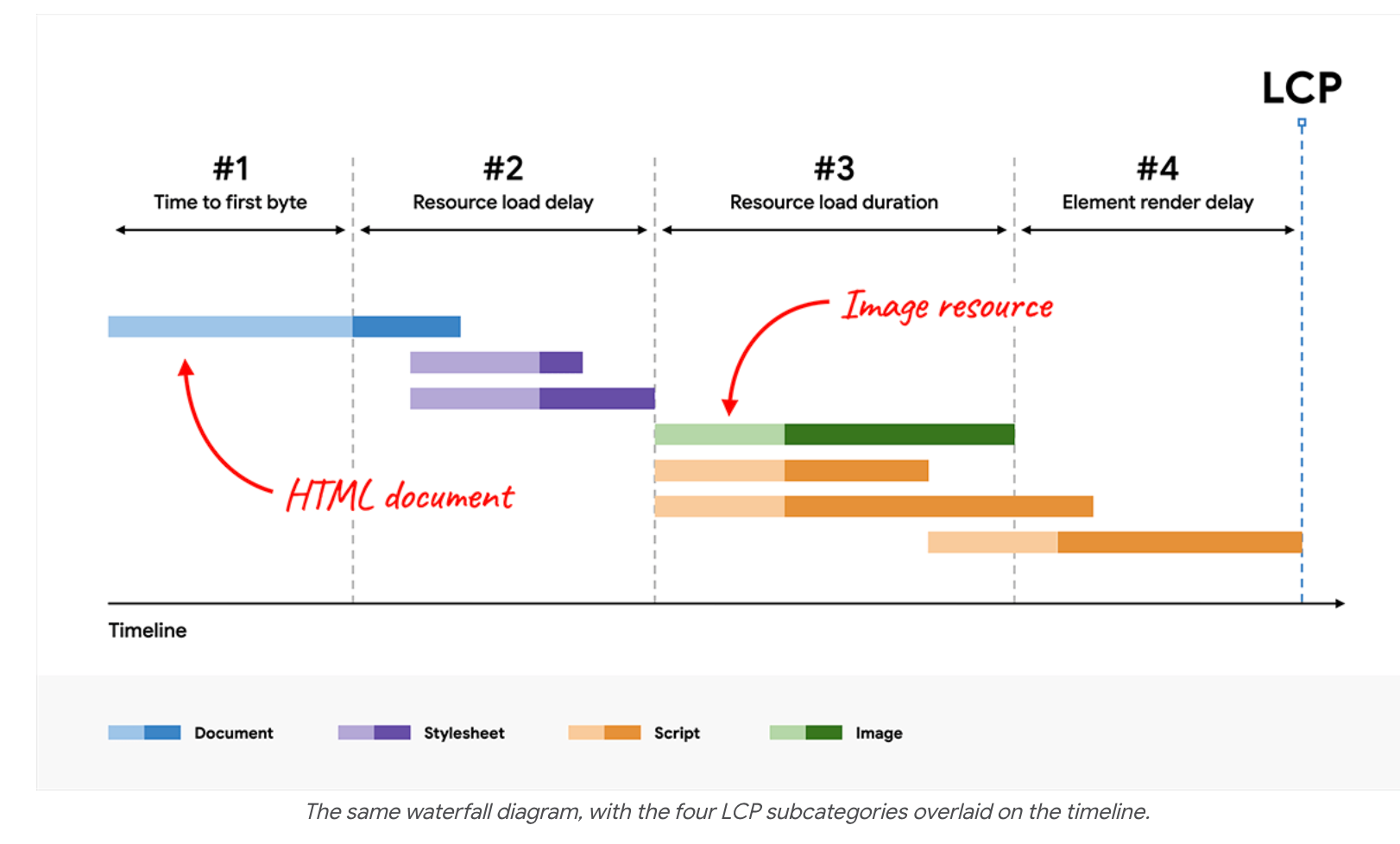
<https://github.com/GoogleChrome/web-vitals?tab=readme-ov-file#usage>

我的代码例子

**Optimization**

**LCP OPT**

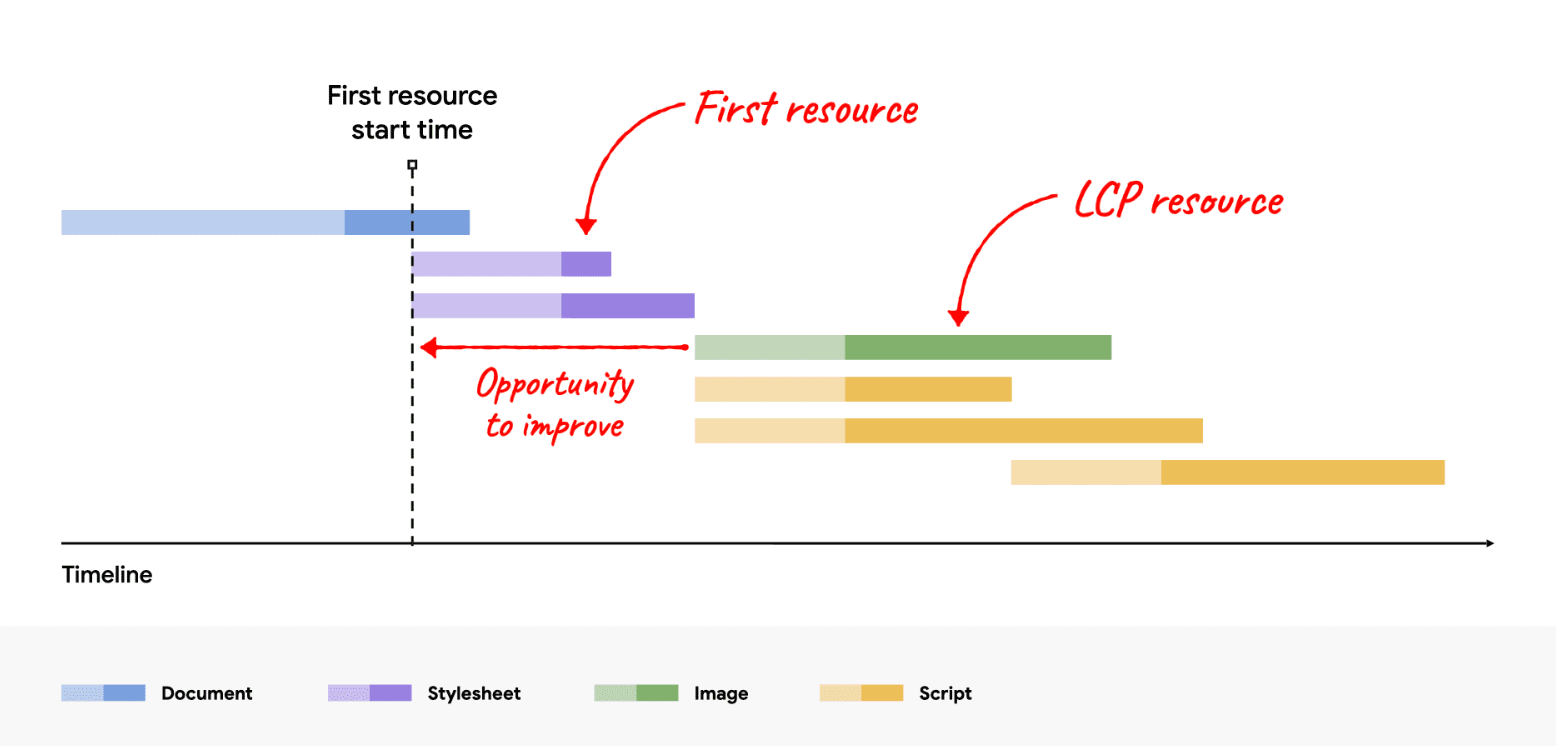
breakdown first



1. **elimate resource load delay** 对应2

if the LCP resource starts loading later than the first resource, then there's opportunity for improvement.

try to remove all in #2



1. Optimize when the resource is discovered

The LCP element is an <img> element, and its src or srcset attributes are present in the initial HTML markup.

The LCP element requires a CSS background image, but that image is preloaded using <link rel="preload"> in the HTML markup (or using a Link header).

The LCP element is a text node that requires a web font to render, and the font is loaded using <link rel="preload"> in the HTML markup (or using a Link header).

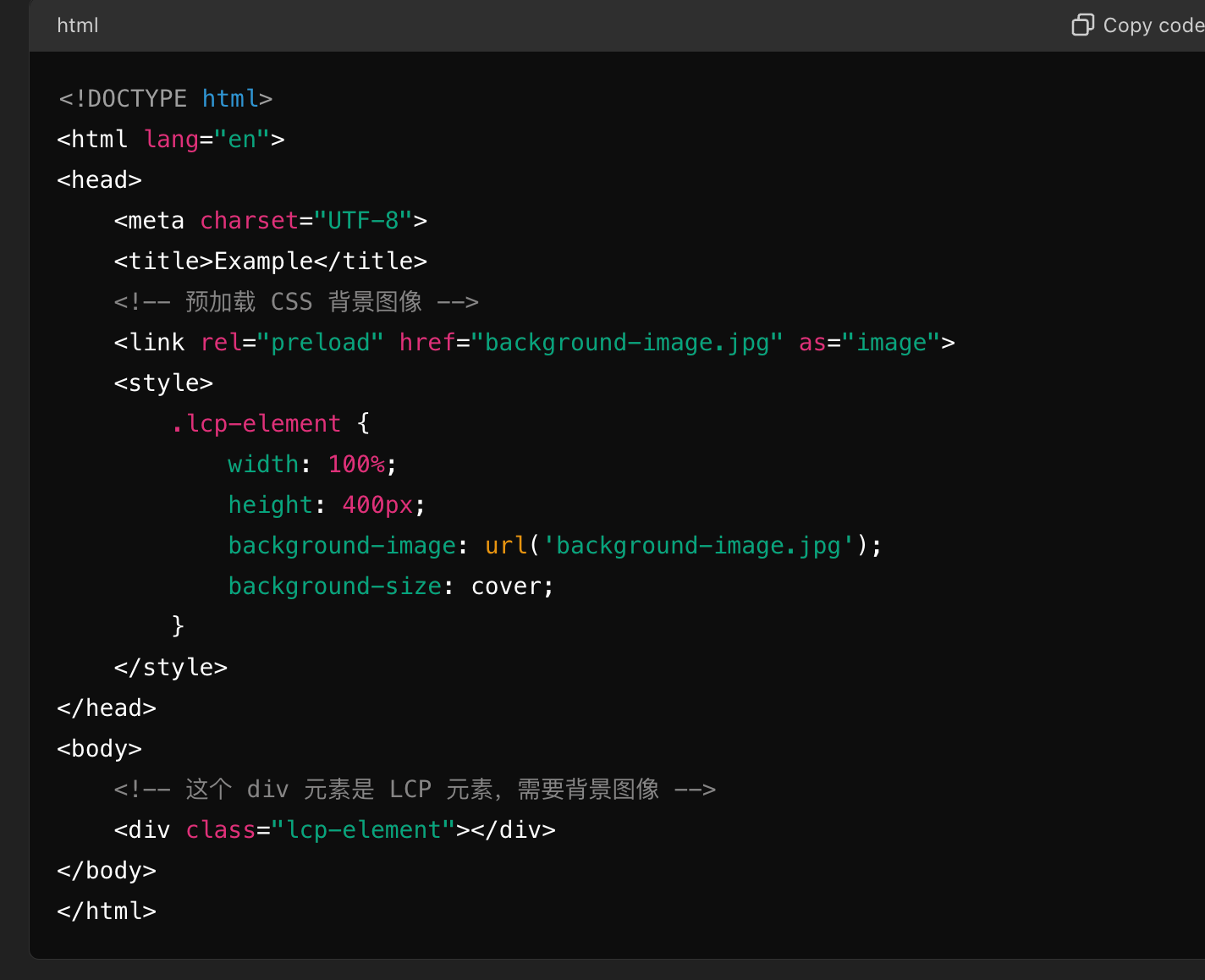
Here are some examples where the LCP resource cannot be discovered from scanning the HTML document response:

The LCP element is an <img> that is dynamically added to the page using JavaScript.

The LCP element is lazily loaded with a JavaScript library that hides its src or srcset attributes (often as data-src or data-srcset).

The LCP element requires a CSS background image.

<https://chatgpt.com/c/f6ff4c4d-76fc-4364-ad53-46568ee1ae8b>

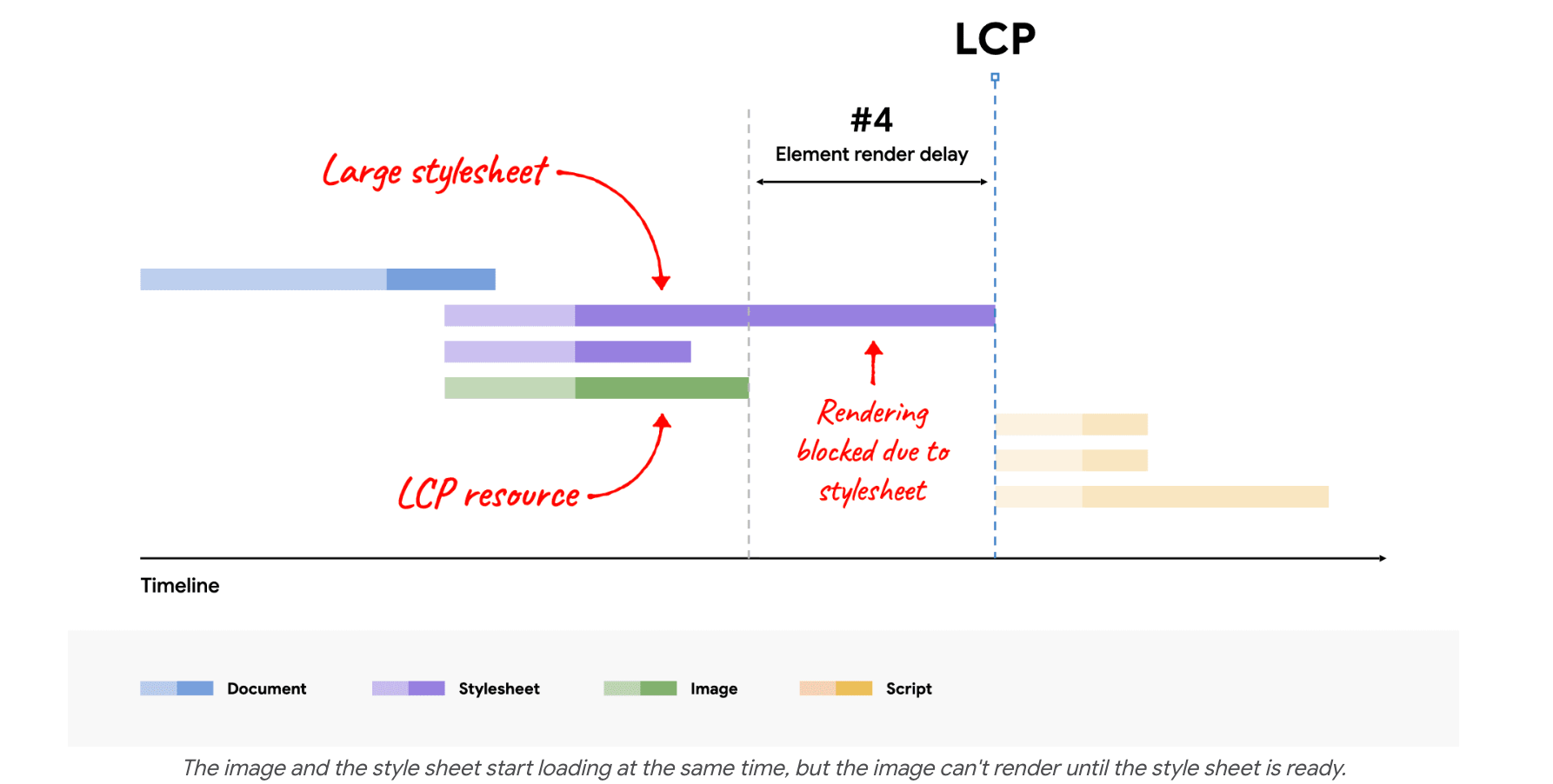


1. Optimize the priority the resource is given

Never lazy-load your LCP image, as that will always lead to unnecessary resource load delay, and will have a negative impact on LCP.

<img fetchpriority="high" src="/path/to/hero-image.webp">

1. **reduce resouce load time** 对应4
2. stylesheet too large



inline the style sheet into the HTML to avoid the additional network request; or,

reduce the size of the style sheet.

听起来是css work

1. Defer or inline render-blocking JavaScript (in <head>)
2. server side rendering
3. **reduce resouce load time**  
   Reduce the size of the resource.

Reduce the distance the resource has to travel. (geographically )

Reduce contention for network bandwidth.

Eliminate the network time entirely. (cache)

1. **reduce TTFB**

more like a network problem

**INP OPT**

1. The input delay, which starts when the user initiates an interaction with the page, and ends when the event callbacks for the interaction begin to run.

Utilize techniques like debounce and throttle to limit the frequency of expensive operations.

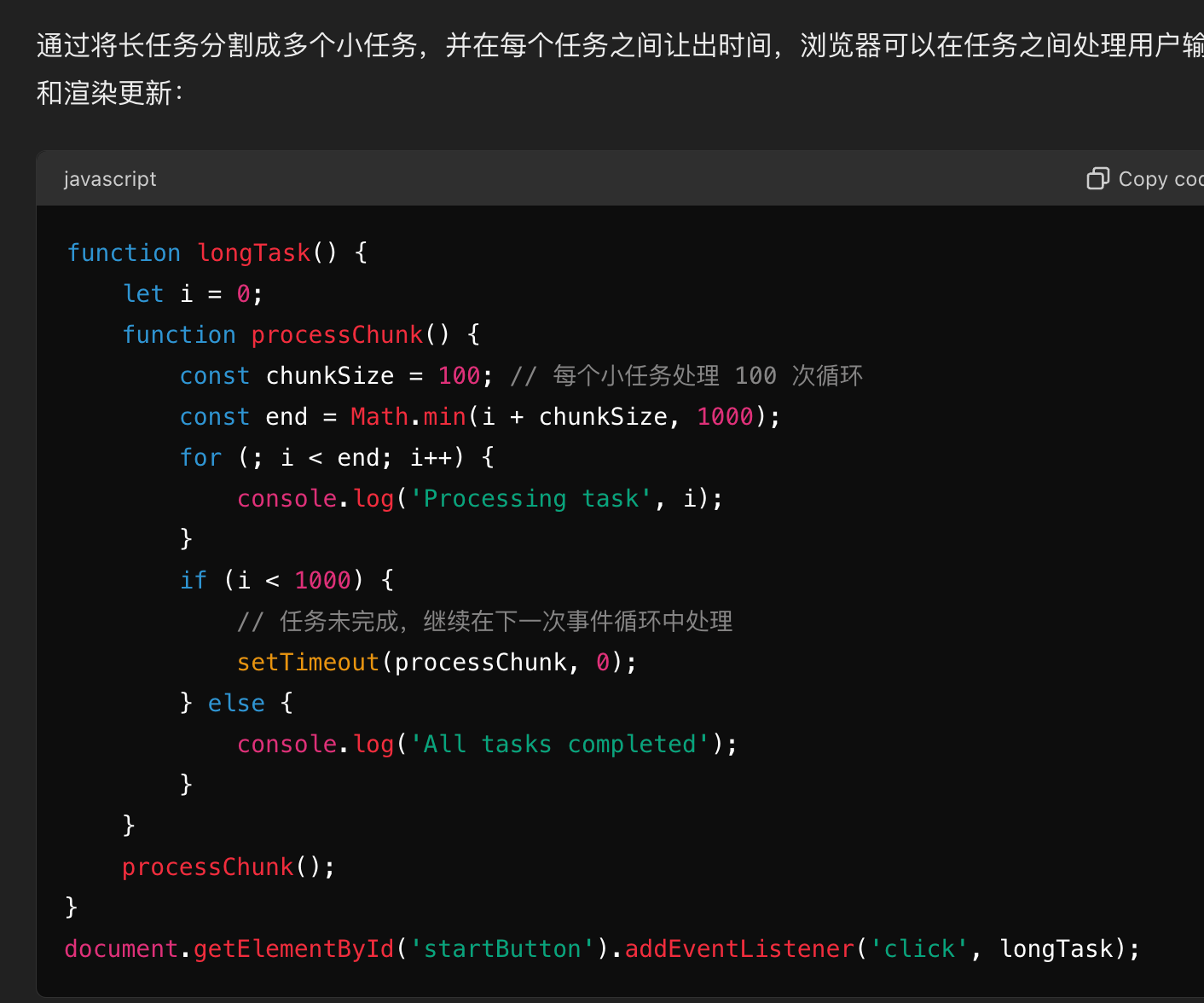
1. The processing duration, which consists of the time it takes for event callbacks to run to completion.

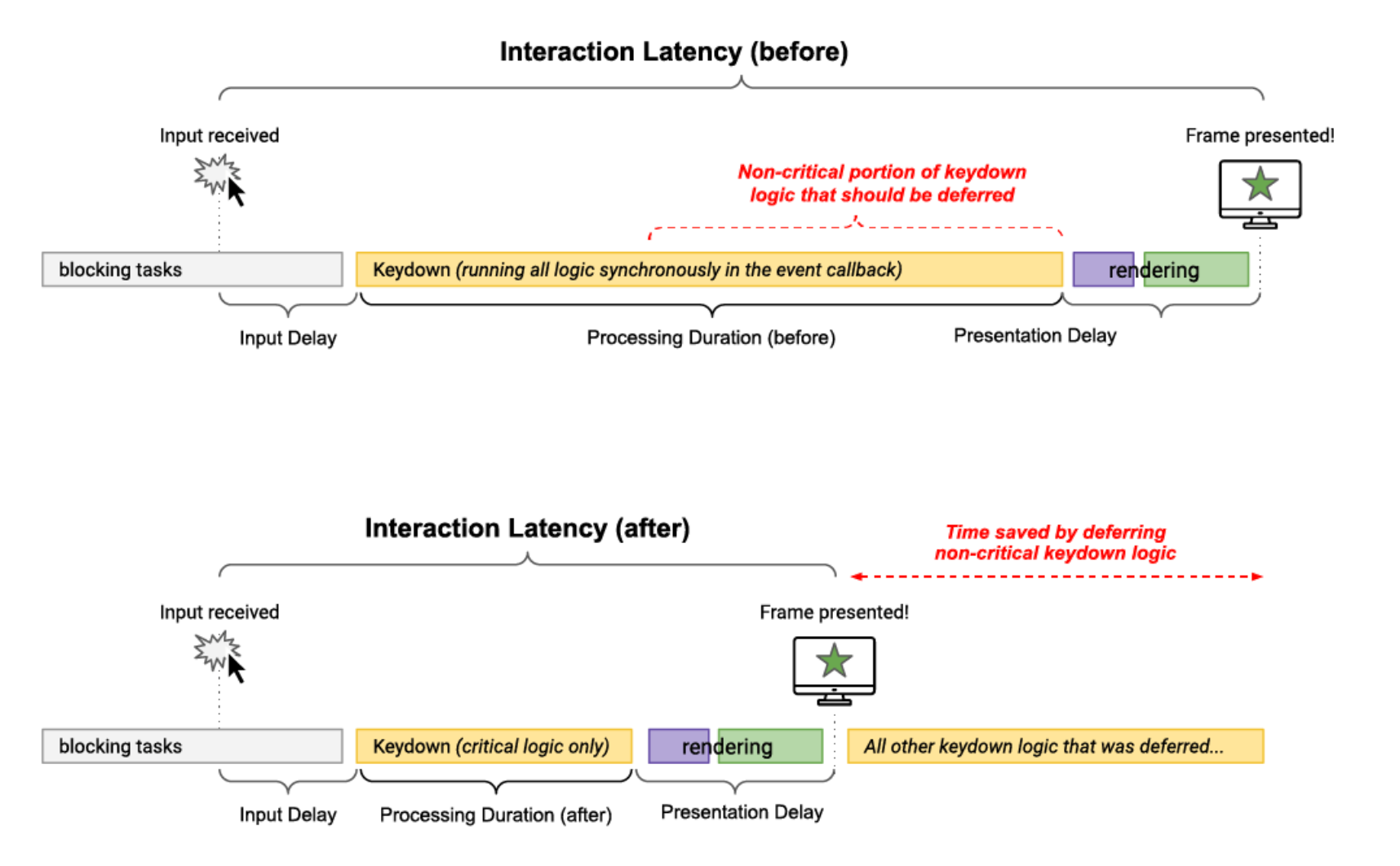
Yield

the next thing you can try is to break up the work in event callbacks into separate tasks. This prevents the collective work from becoming a long task that blocks the main thread

**setTimeout** is one way to break up tasks, because the callback passed to it runs in a new task

like some save to database steps





1. The presentation delay, which is the time it takes for the browser to present the next frame which contains the visual result of the interaction.
2. Minimize DOM size
3. Use content-visibility to lazily render off-screen elements

content-visibility: auto;

1. Be aware of performance costs when rendering HTML using JavaScript

**CLS OPT**

The most common causes of a poor CLS are:

1. Images without dimensions.

Always include width and height size attributes on your images and video elements. Alternatively, reserve the required space with CSS aspect-ratio or similar. This approach ensures that the browser can allocate the correct amount of space in the document while the image is loading.

1. Ads, embeds, and iframes without dimensions.

Reserve space for late-loading content

Place late-loading content lower in the viewport

Avoid inserting new content without a user interaction

1. Web fonts.
2. fallback font
3. invisible. Even if the text is invisible, it's still laid out using the fallback font

font-display: optional can avoid a re-layout as the web font is only used if it is available by the time of initial layout.

Ensure the appropriate fallback font is used. For example, using font-family: "Google Sans", sans-serif; will ensure the browser's sans-serif fallback font is used while "Google Sans" is loaded. Not specifying a fallback font using just font-family: "Google Sans" will mean the default font is used, which on Chrome is "Times"—a serif font which is a worse match than the default sans-serif font.

Minimize the size differences between the fallback font and the web font using the new size-adjust, ascent-override, descent-override, and line-gap-override APIs as detailed in the Improved font fallbacks post.

The Font Loading API can reduce the time it takes to get necessary fonts.

Load critical web fonts as early as possible using <link rel=preload>. A preloaded font will have a higher chance to meet the first paint, in which case there's no layout shifting.

use lighthouse

