

Top core web vitals recommendation for 2023

Explain to you like you are 5 years old(I tried but maybe we should fix at 22 years old)

You don't need to stay through this presentation

- If you are busy, you just need to read through these links:
 - Our top Core Web Vitals recommendations for 2023
 - Top Core Web Vitals Recommendations for 2023 — Barry Pollard — We Love Speed 2023
- If you stay and go through these next slides, I will try to answer you 3 questions: What is it?/Why should we do it?/How to do it? :
 - Largest Contentful Paint(LCP)
 - Cumulative Layout Shift(CLS)
 - First Input Delay(FID)

(Loading)

LCP

Largest Contentful Paint



LCP - What?

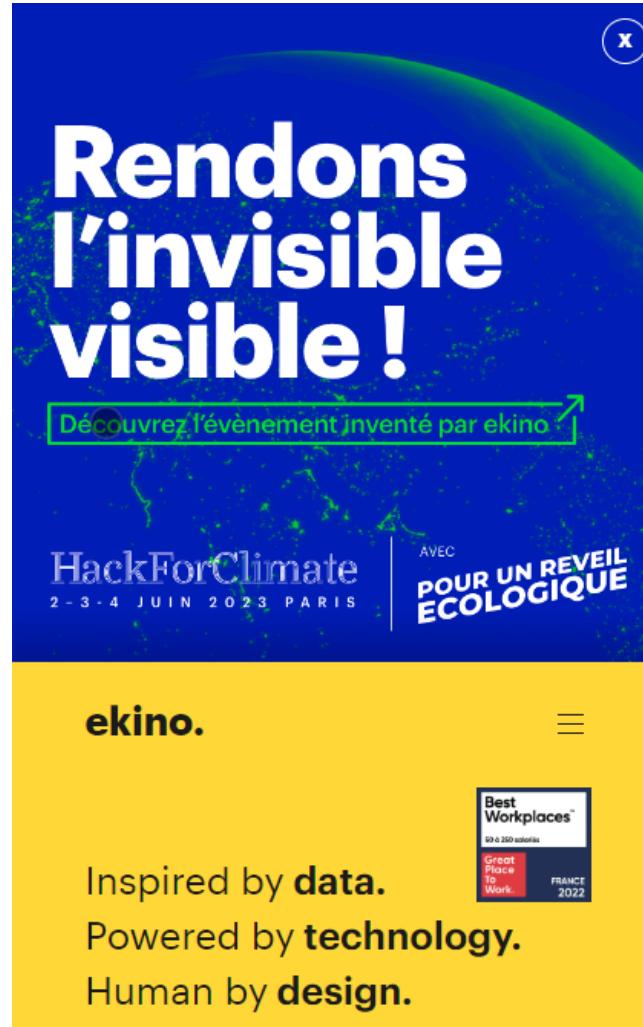
- The Largest Contentful Paint (LCP) metric reports the render time of the largest image or text block visible within the viewport, relative to when the page first started loading.
- *Simple explanation for a kid:* When you first time see your crush, the largest impression(*content*) may be his/her eyes(*image*) or his/her voice(*text*). The first impression may impact your feeling(*your user*) forever.

LCP - Example - Ekino site in mobile view

Question: Do you know what is the LCP part ?

Answer:

The blue part which is an image is considered as the current LCP.



LCP - Elements to be considered

- `img` elements
- `image` elements inside an `svg` element
- `video` elements with a poster image (the poster image load time is used)
- An element with a background image loaded via the `url()` function (as opposed to a CSS gradient)
- Block-level elements containing text nodes or other inline-level text elements children.

Not every element has been equal considered, but every element will have the opportunity to be considered

Not John F.Kennedy

LCP - Elements to be ignored

- Elements with an opacity of 0, that are invisible to the user
- Elements that cover the full viewport, that are likely considered as background rather than content
- Placeholder images or other images with a low entropy, that likely do not reflect the true content of the page

Not every element has been equal considered, but every element will have the opportunity to be considered

Not John F.Kennedy

LCP/CLS/FID - Why?

Cases studies:

- Carpe improved LCP by 52% and CLS by 41%: 10% increase in traffic, 5% increase conversion rate, and 15% increase in revenue.
- Sunday Citizen improve 25% in LCP and 61% in CLS: 4% decrease in bounce rate and over 6% increase in conversion.
- Groupe Renault improve 1 second improvement: 14% point decrease in bounce rate, and 13% increase in conversions.
- Swappie improved LCP by 55%, CLS by 91% and FID by 90%: 42% increase in mobile revenue and a 10% point increase in relative mobile conversion rate.

For more cases studies, read [here](#)

Photo by Alexander Mils on Unsplash

LCP - How ?

- Ensure the LCP resource is discoverable from the HTML source
- Ensure the LCP resource is prioritized
- Use a CDN to optimize document and resource TTFB



LCP - Ensure the LCP resource is discoverable from the HTML source

According to the 2022 Web Almanac by HTTP Archive, 72% of mobile pages have an image as their LCP element – [Our top Core Web Vitals recommendations for 2023](#)

Make sure your image got discoverable by user's browser

- Load the image using an `img` element with the `src` or `srcset` attribute.
- Prefer server-side rendering (SSR) over client-side rendering (CSR)
- If your image needs to be referenced from an external CSS or JS file, you can still include it in the HTML source via a `link` tag

```
<link rel="preload" href="an-image-800w.jpg" as="image">

```

LCP - Ensure the LCP resource is prioritized

Even we already applied the code in previous slide, our LCP image may not be loaded faster than other resources, so how should we make it get more prioritized

- Add `fetchpriority="high"` to the `img` tag of your LCP image.
- Never set `loading="lazy"` on the `img` tag of your LCP image.
- Defer non-critical resources when possible.

```
<link rel="preload" href="an-image-800w.jpg" as="image" fetchpriority="high">

<ul class="carousel">
  
  
  
  
</ul>
```

LCP - Use a CDN to optimize document and resource TTFB

We make the resources discoverable and prioritized. Now we need to deliver these resources as fast as possible. The browser will not do anything until it received the first byte of the initial HTML document response, or we can call it Time to First Byte (TTFB). To do it, we serve the content through the CDN:

- Serve your content as geographically close to your users as possible
- Cache that content so recently-requested content can be served again quickly.
- CDN can host not only static assets but also HTML documents(included also dynamical generated one)

(Visual Stability)

CLS

Cumulative Layout Shift



CLS - What?

- Cumulative Layout Shift (CLS) is an important, user-centric metric for measuring visual stability because it helps quantify how often users experience unexpected layout shifts—a low CLS helps ensure that the page is delightful.
- *Simple explanation for a kid:* When you throw a ball(*a user action*) to your friend but he ducks(*layout shift*) and the ball hit your crush(*wrong action*). This will not be a happy ending(*bad user experience*).

CLS - Example - Layout shift makes wrong click

A screencast illustrating how layout instability can negatively affect users.

Order confirmation

You have selected **56** items. Is this correct?

[Yes, place my order](#)

[No, go back](#)

CLS - How ?

- Set explicit sizes on any content loaded from the page
- Ensure pages are eligible for bfcache
- Avoid animations/transitions that use layout-inducing CSS properties

HOW ?



CLS - Set explicit sizes on any content loaded from the page

According to HTTP Archive, 72% of pages have at least one unsized image – *Our top Core Web Vitals recommendations for 2023*

To fix layout shifts caused by unsized images/videos/third-party ads is to explicitly set width and height attributes (or equivalent CSS properties). If we don't know the real height of the content, make a container with min-height may help to reduce a little layout shift to an acceptable tolerant.

```
  
<video width="1600" height="900"></video>  
<div class="ad-container" style="min-height: 500px">  
  Your ad goes here  
</div>
```

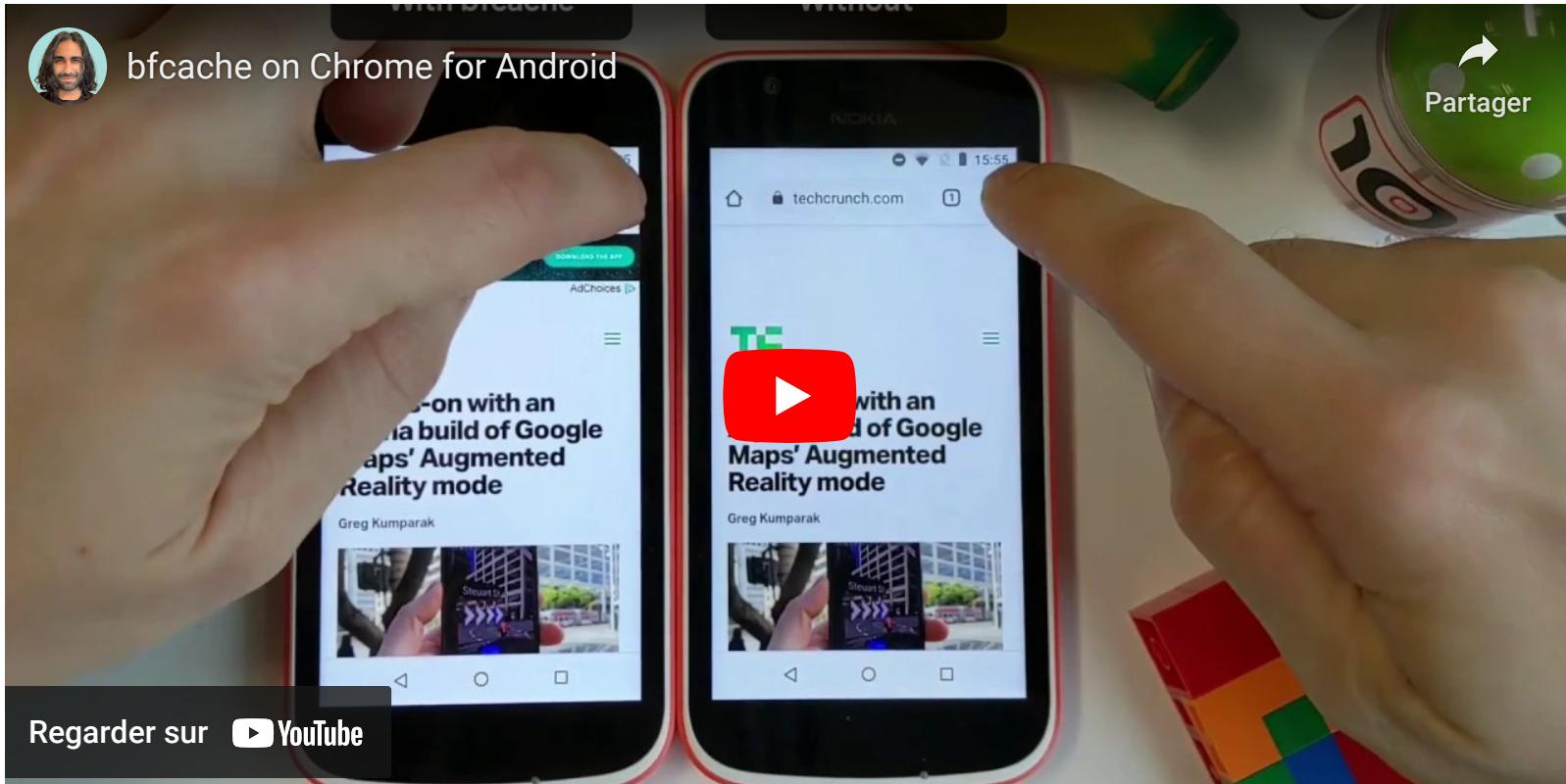
CLS - Ensure pages are eligible for bfcache

Back/forward cache (or bfcache) is a browser optimization that enables instant back and forward navigation.

bfcache is an in-memory cache that stores a complete snapshot of a page (including the JavaScript heap) as the user is navigating away.
With the entire page in memory, the browser can quickly and easily restore it if the user decides to return. – *Back/forward cache*

As a site owner, you should always check your site to be eligible for bfcache, and if not, read the reason and find a way to optimize it. Unless your site contain sensitive information(as bank information, transaction, user account), you should strive to have bfcache active.

CLS - Example between with and without bfcache



CLS - Avoid animations/transitions that use layout-inducing CSS properties

When element got animated, you may not notice but it can be a reason for layout shift. Animate or transition any CSS property that requires the browser to update the page layout will cause layout shift.

```
span {  
  margin: 10px;  
  padding: 10px;  
  position: absolute;  
  top: 10px;  
  left: 10px;  
  right: 10px;  
  bottom: 10px;  
}
```

Layout shift will also happen with element bring out of the document flow like element with `position: absolute`. Always prefer to use `transform` property.

You can read more in [Avoid non-composited animations](#)

(Interactivity)

FID

First Input Delay



FID - What?

- First Input Delay (FID) is a Core Web Vitals metric that captures a user's first impression of a site's interactivity and responsiveness. It measures the time from when a user first interacts with a page to the time when the browser is actually able to respond to that interaction.
- *Simple explanation for a kid:* When your mother(*your user*) ask your father(*your site*) a question about why did he come home late, if your father(*your site*) can't response quick enough, someone will have harsh time.

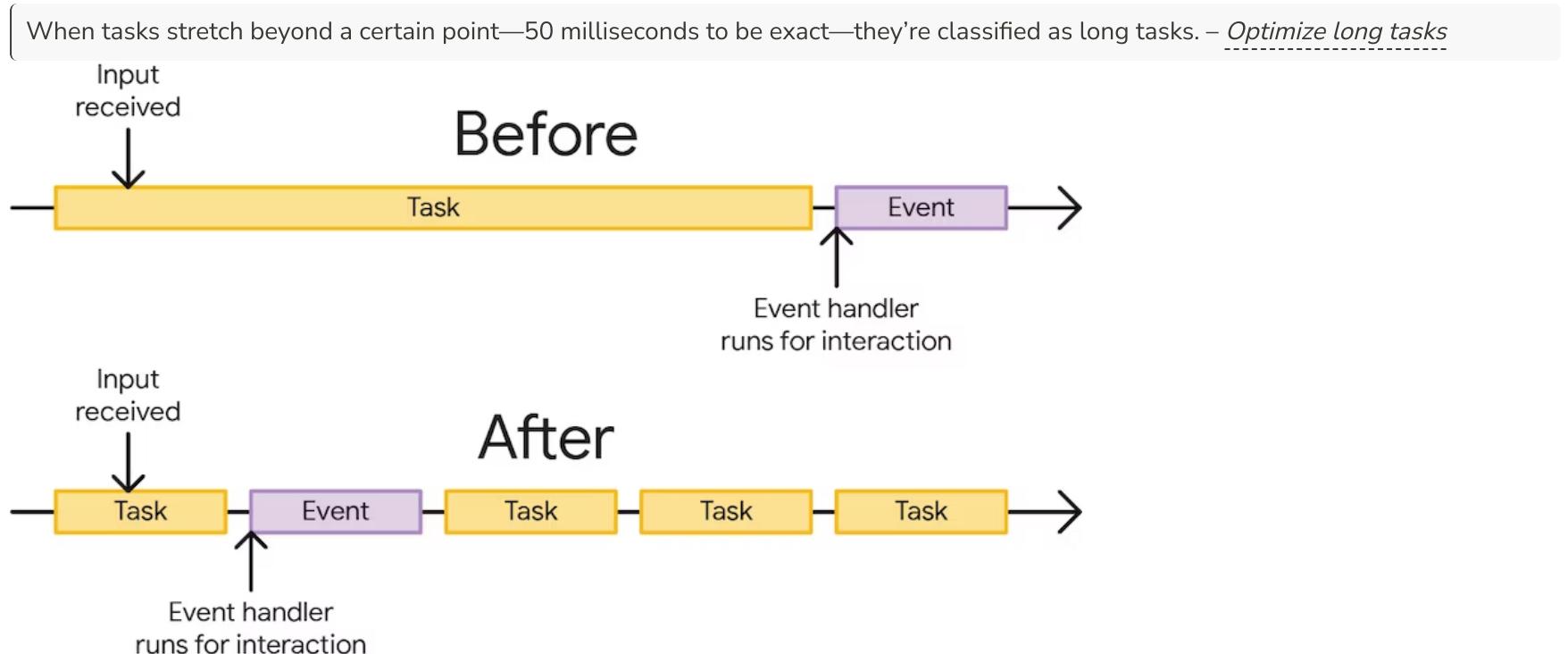
FID - How?

- Avoid or break up long tasks
- Avoid unnecessary JavaScript
- Avoid large rendering updates



FID - Avoid or break up long tasks

Don't block the main thread. And if you can't avoid long task, it is better to break it up to many smaller tasks.



By breaking up into smaller tasks, you stop blocking the main thread so the user can interact normally.

FID - Avoid unnecessary JavaScript

A large JS file may have a lot of unused function, or wait to be used. How should we deliver a better file?

- Use any coverage tool to find out unused code, remove it to save time
- The unused code may be not really "unused" or we can mark it as waiting to be used, use code splitting into several bundles to require only when we need

FID - Avoid large rendering updates

When large rendering updates happen, they can interfere with your website's ability to respond to user inputs.

Some advices:

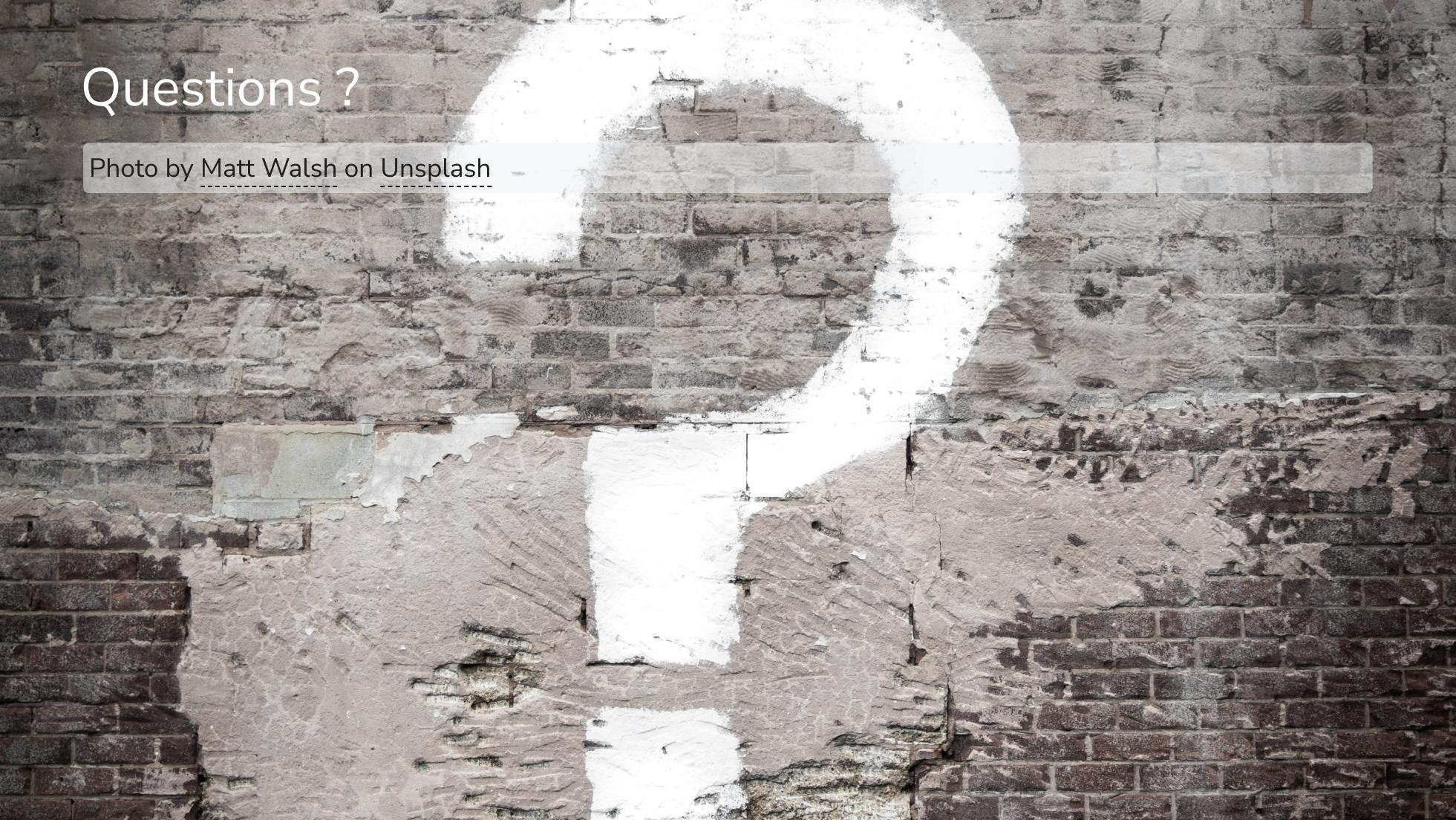
- Avoid using `requestAnimationFrame()` for doing any non-visual work.
- Keep your DOM size small.
- Use CSS containment.

```
<article>
  <h2>Heading of a nice article</h2>
  <p>Content here.</p>
</article>
<article>
  <h2>Another heading of another article</h2>
  <p>More content here.</p>
</article>

article {
  contain: content;
}
```

Questions ?

Photo by Matt Walsh on [Unsplash](#)



Thank you for your attention !

Photo by Wilhelm Gunkel on Unsplash

DANKE !
THANK YOU !

MERCI !

GRAZIE !

GRACIAS !

DANK JE WEL !



Preferences

- Our top Core Web Vitals recommendations for 2023
- Optimize Largest Contentful Paint
- Optimizing resource loading with the Fetch Priority API
- Cumulative Layout Shift (CLS)
- Optimize Cumulative Layout Shift
- Back/forward cache
- Optimize First Input Delay
- Optimize long tasks
- CSS containment
- Web Vitals
- Dynamic LCP Priority: Learning from Past Visits
- Cases studies