MOBILE

1.PERFORMANCE

1. Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption.

<https://web.dev/uses-webp-images/?utm_source=lighthouse&utm_medium=devtools>

1. Eliminate render-blocking resources

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles.

Which URLs get flagged as render-blocking resources? #

Lighthouse flags two types of render-blocking URLs: scripts and stylesheets.

A <script> tag that:

Is in the <head> of the document.

Does not have a defer attribute.

Does not have an async attribute.

A <link rel="stylesheet"> tag that:

Does not have a disabled attribute. When this attribute is present, the browser does not download the stylesheet.

Does not have a media attribute that matches the user's device specifically. media="all" is considered render-blocking.

## How to eliminate render-blocking scripts [#](https://web.dev/render-blocking-resources/?utm_source=lighthouse&utm_medium=devtools#how-to-eliminate-render-blocking-scripts)

Once you've identified critical code, move that code from the render-blocking URL to an inline script tag in your HTML page. When the page loads, it will have what it needs to handle the page's core functionality.

If there's code in a render-blocking URL that's not critical, you can keep it in the URL, and then mark the URL with async or defer attributes (see also [Adding Interactivity with JavaScript](https://web.dev/critical-rendering-path-adding-interactivity-with-javascript/)).

Hepsi de cok kritik olmadigi icin sadece defer ekledim

Graphical user interface

Description automatically generated with medium confidence

## How to eliminate render-blocking stylesheets [#](https://web.dev/render-blocking-resources/?utm_source=lighthouse&utm_medium=devtools#how-to-eliminate-render-blocking-stylesheets)

Similar to inlining code in a <script> tag, inline critical styles required for the first paint inside a <style> block at the head of the HTML page. Then load the rest of the styles asynchronously using the preload link (see [Defer unused CSS](https://web.dev/defer-non-critical-css)).

CSS files are [render-blocking resources](https://developers.google.com/web/tools/lighthouse/audits/blocking-resources): they must be loaded and processed before the browser renders the page. Web pages that contain unnecessarily large styles take longer to render.

By preloading a certain resource, you are telling the browser that you would like to fetch it sooner than the browser would otherwise discover it because you are certain that it is important for the current page.