Eliminate render-blocking JavaScript and CSS in above-the-fold content | SearchTides

Clip source: Eliminate render-blocking JavaScript and CSS in above-the-fold content | SearchTides

Eliminate render-blocking JavaScript and CSS in above-the-fold content

If you follow the Google Webmaster Central blog and you keep up with SEO trends you'll no doubt understand the importance of website speed. Google's best practices are the guidelines that website owners all over the world should follow. Implement Google's recommendations and …
March 9, 2017
Scroll Down.
If you follow the Google Webmaster Central blog and you keep up with SEO trends you'll no doubt understand the importance of website speed. If not, I suggest you check out the former resource at least. Google's best practices are the guidelines that website owners all over the world should follow. It's vitally important to keep up with the ever-changing search engine algorithm. Implement Google's recommendations and see your ranking improve!

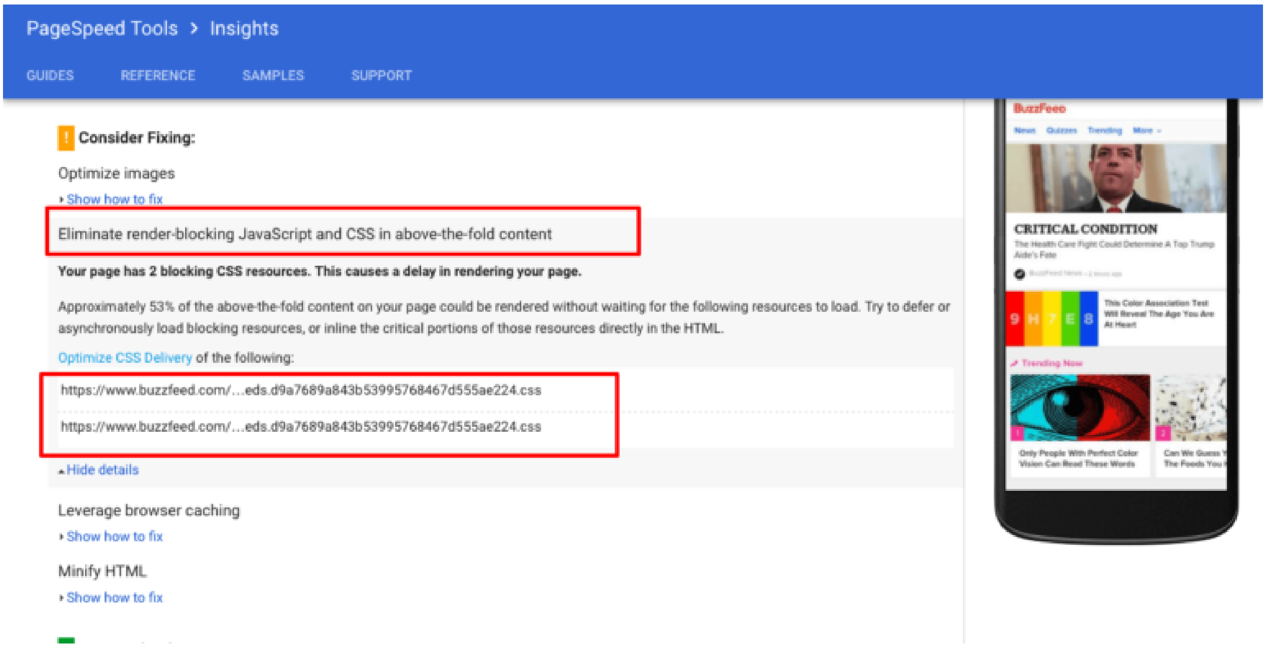
Any SEO worth his salt can tell you about the importance of having a fast loading website. Website users in 2017 will not wait for more than a couple of seconds for a page to load. It's a fast world and the internet is expected to deliver information quickly. A slow-loading website provides a poor user experience. Google punishes these websites by putting them lower in the search results. And this means less organic traffic.

What causes a web page to download slowly?

Many things, but the one I see most frequently has to do with the loading of JavaScript and CSS on your page. 90% of the website's I've tested suffer from poor performance because of "render-blocking JavaScript and CSS in above-the-fold content". If you can fix this problem then you'll be one step ahead of the competition. What does this mean?

1. Render-blocking – Something is preventing the website from loading quickly.
2. Javascript – A scripting language that is commonly used on modern websites.
3. CSS – The styling code that makes plain HTML look pretty.

4. Above-the-fold content – The part of the website that appears on screen without requiring the user to scroll down.



PageSpeed Tools > Insights

GUIDES REFERENCE SAMPLES SUPPORT

! Consider Fixing:

Optimize images

▸ Show how to fix

Eliminate render-blocking JavaScript and CSS in above-the-fold content

Your page has 2 blocking CSS resources. This causes a delay in rendering your page.

Approximately 53% of the above-the-fold content on your page could be rendered without waiting for the following resources to load. Try to defer or asynchronously load blocking resources, or inline the critical portions of those resources directly in the HTML.

Optimize CSS Delivery of the following:

https://www.buzzfeed.com/…eds.d9a7689a843b53995768467d555ae224.css

https://www.buzzfeed.com/…eds.d9a7689a843b53995768467d555ae224.css

▴ Hide details

Leverage browser caching

▸ Show how to fix

Minify HTML

▸ Show how to fix

BuzzFeed
News Quizzes Trending More ⌄
CRITICAL CONDITION
The Health Care Fight Could Determine A Top Trump Aide's Fate
9 H 7 E 8
This Color Association Test Will Reveal The Age You Are At Heart
Trending Now
1
Only People With Perfect Color Vision Can Read These Words
2
Can We Guess Y
The Foods You

In summary: some of the code required to show the initial screen of the website is blocking the entire site from loading quickly. The remedy? Optimize or remove that code.

How do I find out if I have a Javascript and CSS rendering problem on my website?

Go to Google's PageSpeed Insights tool or the excellent GTMetrix. Both of these services highlight potential issues which affect the speed of your website.

I'm going to show you how to address "eliminate render-blocking javascript and css in above-the-fold content" for WordPress websites. I will give two different solutions. The first is a solution for developers and people competent in editing code. The second is for the rest of us. Both work pretty much the same way but developers will be able to tweak the code a little bit more to their own requirements.

For most websites, Javascript will be contributing more to the problem so let's start there

Solution 1 (for developers)

Run Google’s PageSpeed Insights test. If the results determine that JavaScript is causing the site to load slowly, copy the JavaScript-related URLs to a spreadsheet or text document.

Open the *footer.php* file with your favorite code editor via FTP (or access via Appearance > Editor)

Add the following code before </body>:

```
<script type=”text/javascript”>
 function parseJSAtOnload() {
 var links = [“javascript1.js”, “javascript2.js”, “javascript3.js”],
 headElement = document.getElementsByTagName(“head”)[0],
 linkElement, i;
 for (i = 0; i < links.length; i++) {
 linkElement = document.createElement(“script”);
 linkElement.src = links[i];
 headElement.appendChild(linkElement);
 }
 }
 if (window.addEventListener)
 window.addEventListener(“load”, parseJSAtOnload, false);
 else if (window.attachEvent)
 window.attachEvent(“onload”, parseJSAtOnload);
 else window.onload = parseJSAtOnload;
 </script>
```

Each javascript.js file in line 3 should be replaced with the scripts that Google PageSpeed Insights points out as ‘render-blocking’. Be careful to copy the link that appears as you hover over the URL. Copying the link directly from the results page will give you an incorrectly formed URL as PageSpeed Insights inserts some ellipsis (…) into the link in order to shorten them.

The *javascript.js* URLs will be something like this one http://yourdomain.com/wp-includes/js/jquery/jquery.js?ver=1.12.4

To add more files simply add another comma followed by the URL surrounded by quotation marks.

Once you’ve replaced the placeholder URLs with the real ones, save the file and run the PageSpeed test again.

So that’s the javascript part. The CSS part is much trickier to pull off on WordPress than on say, a straight HTML website. WordPress uses PHP to ‘enqueue’ the css scripts. And because of the requirement (from a UX point of view) to have CSS in place before users see any content, it is difficult to optimize CSS files without the use of a plugin. Technically, all CSS files should be added to the header in WordPress but we’ll need to break some rules here.

The best method I’ve found to do this is the following:

- Locate the style function that loads the styles in your theme’s *php* file.
- Comment out the function code and save the stylesheet URLs. (Eg. http://fonts.googleapis.com/css?family=Roboto:400,100,300,500,700,900)
- Create a new function which loads the styles in the footer. Example:

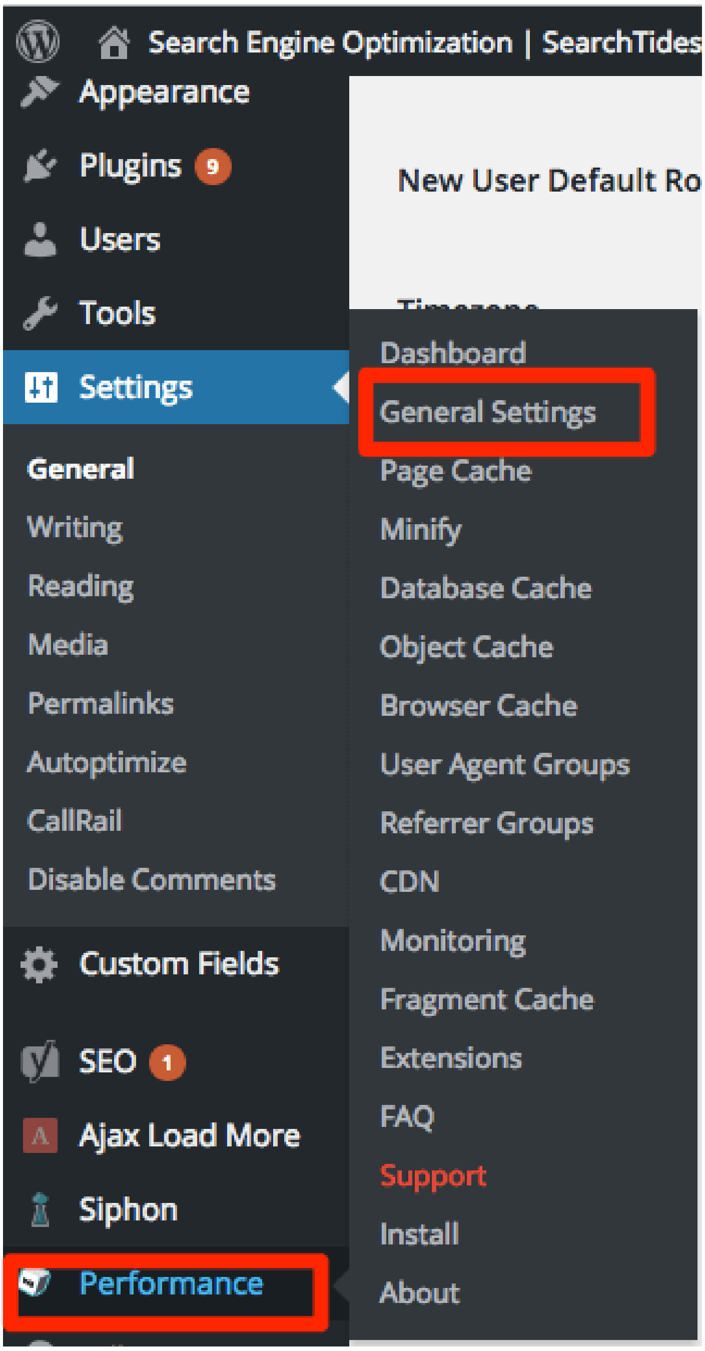
```
function styles_in_wp_foot() {
 echo ‘<link rel=”stylesheet” href=”http://fonts.googleapis.com/css?family=Roboto:400,100,300,500,700,900″ type=”text/css” media=”all” />’;
 echo ‘<link rel=”stylesheet” href=”‘ . get_stylesheet_uri() .'” type=”text/css” media=”all” />’;
 }
 add_action( ‘wp_footer’, ‘styles_in_wp_foot’ ); */
```

- Test for each CSS file and rerun the Speed Test

Solution 2 (for non-developers)

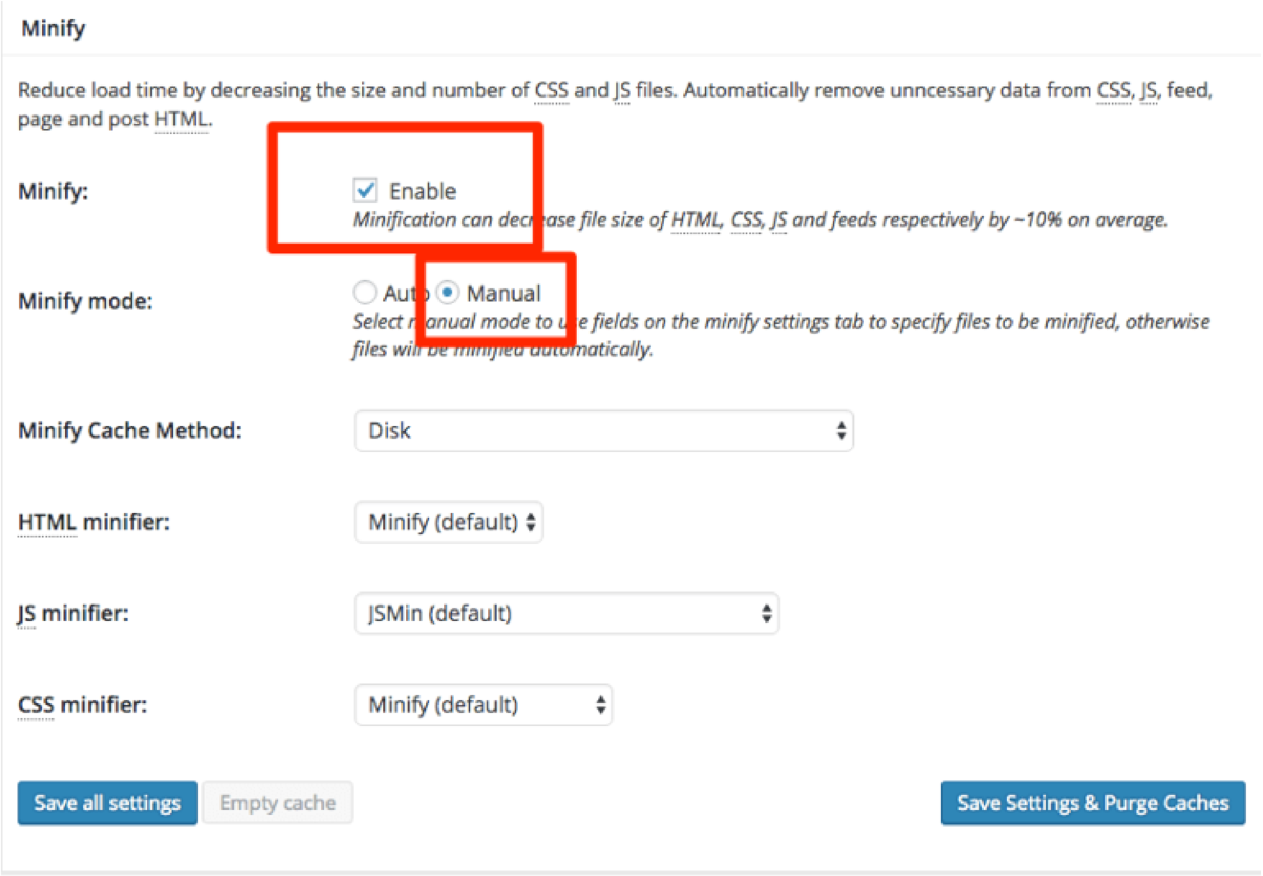
Download and install the plugin W3 Total Cache.

Go to Performance > General Settings



Scroll down to the Minify section and tick the checkbox marked ‘enable’.

Where it says ‘minify mode’ make sure to select ‘manual’.



Minify

Reduce load time by decreasing the size and number of CSS and JS files. Automatically remove unncessary data from CSS, JS, feed, page and post HTML.

Minify: ☑ Enable
Minification can decrease file size of HTML, CSS, JS and feeds respectively by ~10% on average.

Minify mode: ○ Auto ● Manual
Select manual mode to use fields on the minify settings tab to specify files to be minified, otherwise files will be minified automatically.

Minify Cache Method: Disk

HTML minifier: Minify (default)

JS minifier: JSMin (default)

CSS minifier: Minify (default)

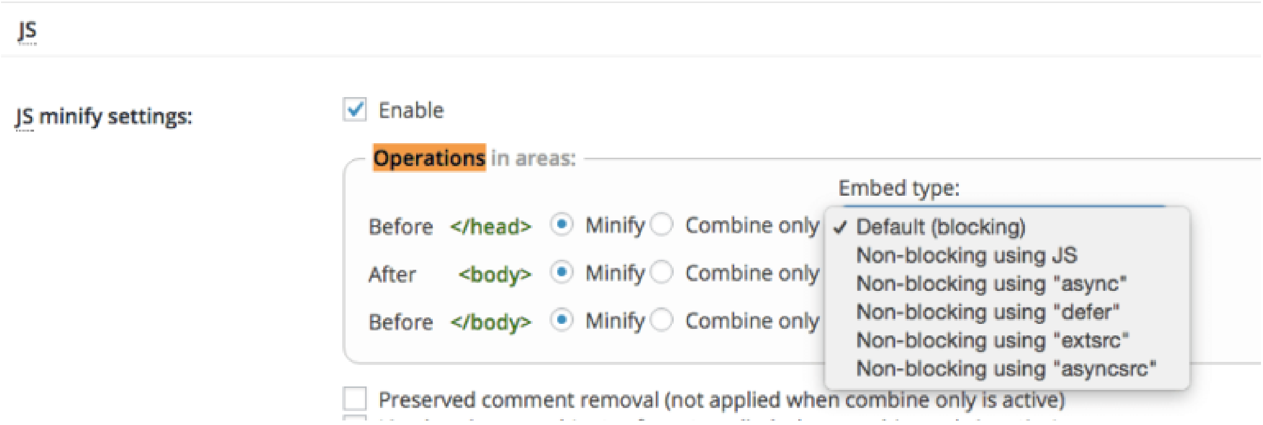
Click **Save Settings & Purge Caches**.

Now head to the Minify page by clicking Minify in the sidebar under Performance.

Scroll down to ‘JS’ and tick the box marked ‘enable’.

In the ‘Operations in Areas’ section we only need to change the first option “Before </Head>”

Select Non-Blocking using “async” in the Embed Type: pull-down list.



JS

JS minify settings: ☑ Enable

Operations in areas:

Before </head> ● Minify ○ Combine only
After <body> ● Minify ○ Combine only
Before </body> ● Minify ○ Combine only

Embed type:

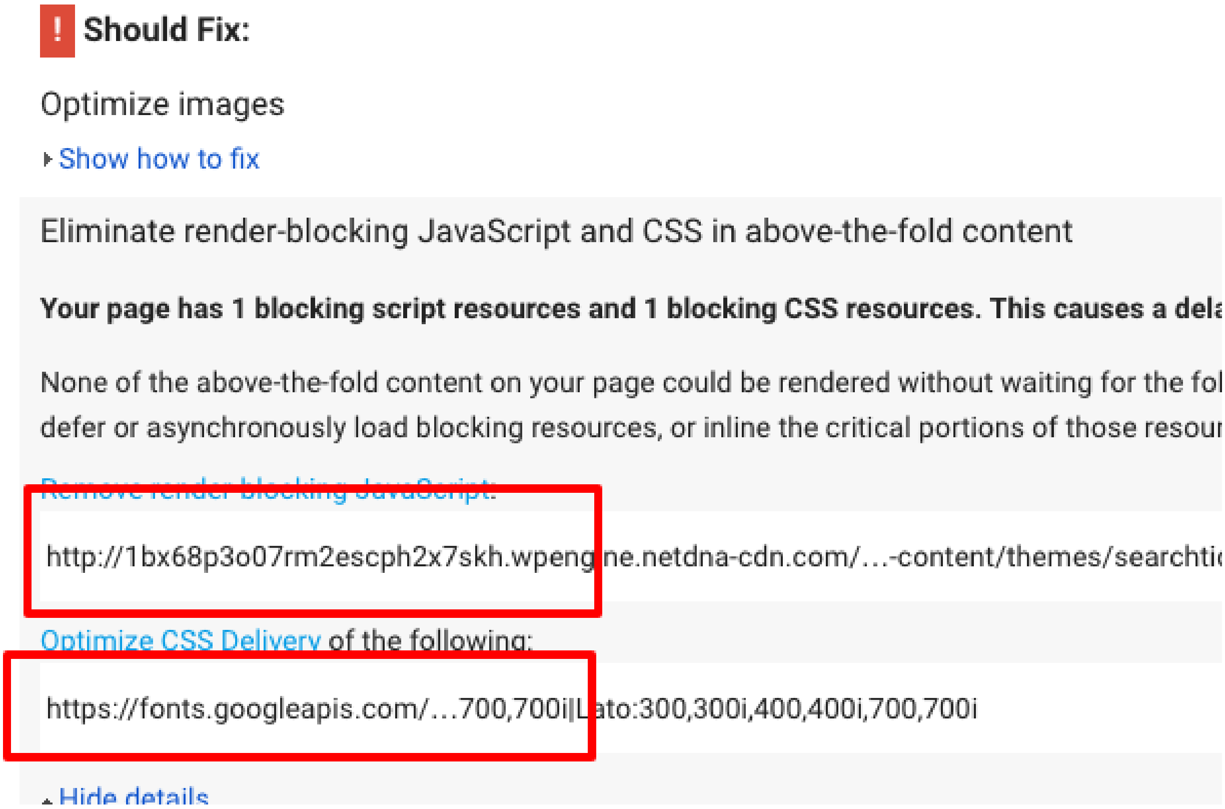
✓ Default (blocking)
Non-blocking using JS
Non-blocking using "async"
Non-blocking using "defer"
Non-blocking using "extsrc"
Non-blocking using "asyncsrc"

□ Preserved comment removal (not applied when combine only is active)

□ Line break removal (not safe, not applied when combine only is active)

You will be presented with some new options called JS File Management. This section allows you to specify the JavaScript files that you want to load asynchronously.

Copy each of the URLs which represent JS files that Google PageSpeed has marked as render-blocking.



Should Fix:

Optimize images

Show how to fix

Eliminate render-blocking JavaScript and CSS in above-the-fold content

Your page has 1 blocking script resources and 1 blocking CSS resources. This causes a dela

None of the above-the-fold content on your page could be rendered without waiting for the fol defer or asynchronously load blocking resources, or inline the critical portions of those resour

Remove render-blocking JavaScript:

http://1bx68p3o07rm2escph2x7skh.wpengine.netdna-cdn.com/…-content/themes/searchti

Optimize CSS Delivery of the following:

https://fonts.googleapis.com/…700,700i|Lato:300,300i,400,400i,700,700i

Hide details

Paste each URL into the File URI: box and select “Embed in <head>” as the Embed Location.

Click **Save Settings & Purge Cache**.

So that’s the Javascript part. We still have to tackle the CSS that’s preventing the page from loading quickly.

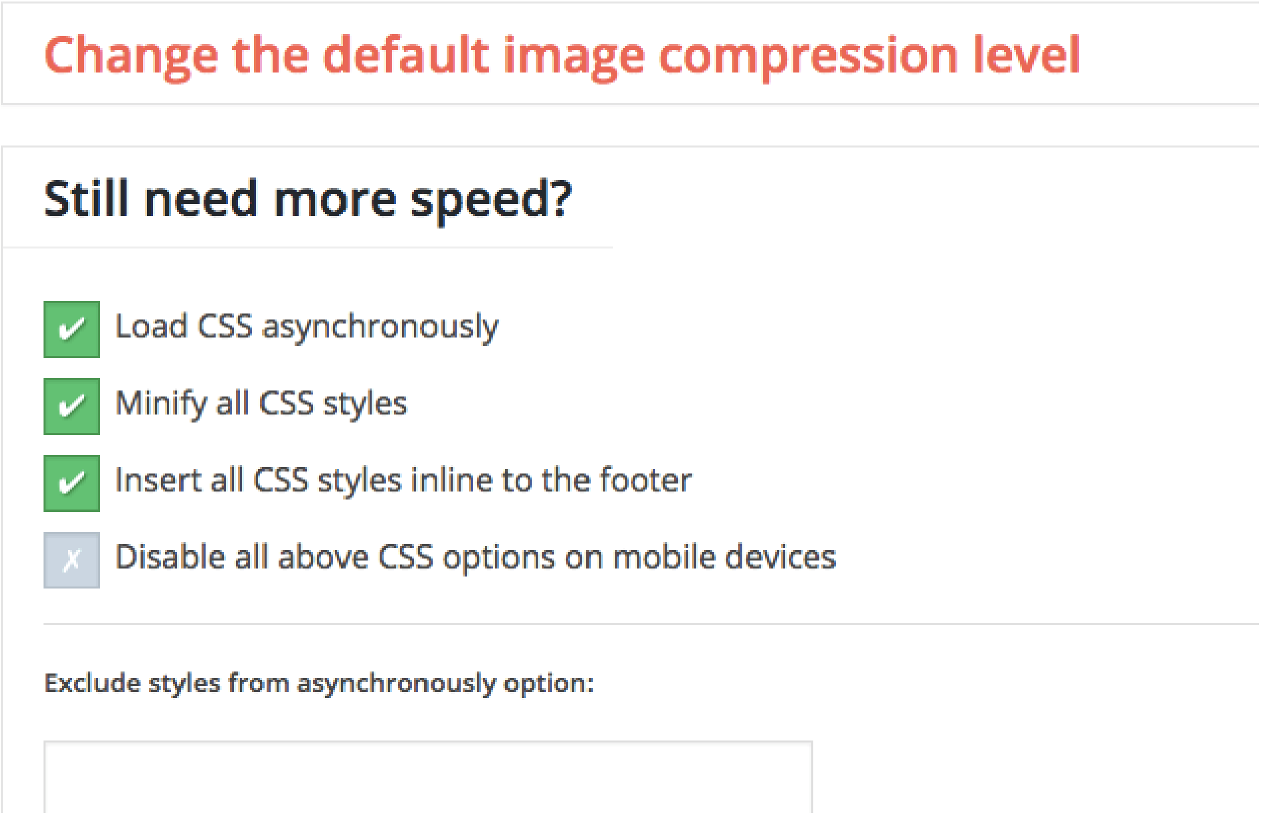
The developers of W3 Total Cache made some changes recently to the plugin and unfortunately if you want to eliminate render blocking CSS in your website you will need to upgrade from the free version. If this is not an option there is another solution.

Install Speed Booster Pack using the Plugins area of the WordPress backend.

Leave the general options area untouched as we've already taken care of the JavaScript. Click on the drop down menu that says "Still need more speed?"

Select the first three options.

- Load CSS asynchronously
- Minify all CSS styles
- Insert all CSS styles inline to the footer



Change the default image compression level

Still need more speed?

☑ Load CSS asynchronously

☑ Minify all CSS styles

☑ Insert all CSS styles inline to the footer

☒ Disable all above CSS options on mobile devices

Exclude styles from asynchronously option: