* [Optimizing caching](https://developers.google.com/speed/docs/best-practices/caching) — keeping your application's data and logic *off* the network altogether
* [**Leverage browser caching**](https://developers.google.com/speed/docs/best-practices/caching#LeverageBrowserCaching)
* [**Leverage proxy caching**](https://developers.google.com/speed/docs/best-practices/caching#LeverageProxyCaching)
* [Minimizing round-trip times](https://developers.google.com/speed/docs/best-practices/rtt) — reducing the number of serial request-response cycles
* [**Minimize DNS lookups**](https://developers.google.com/speed/docs/best-practices/rtt#MinimizeDNSLookups)
* [**Minimize redirects**](https://developers.google.com/speed/docs/best-practices/rtt#AvoidRedirects)
* [**Avoid bad requests**](https://developers.google.com/speed/docs/best-practices/rtt#AvoidBadRequests)
* [**Combine external JavaScript**](https://developers.google.com/speed/docs/best-practices/rtt#CombineExternalJS)
* [**Combine external CSS**](https://developers.google.com/speed/docs/best-practices/rtt#CombineExternalCSS)
* [**Combine images using CSS sprites**](https://developers.google.com/speed/docs/best-practices/rtt#SpriteImages)
* [**Optimize the order of styles and scripts**](https://developers.google.com/speed/docs/best-practices/rtt#PutStylesBeforeScripts)
* [**Avoid document.write**](https://developers.google.com/speed/docs/best-practices/rtt#AvoidDocumentWrite)
* [**Avoid CSS @import**](https://developers.google.com/speed/docs/best-practices/rtt#AvoidCssImport)
* [**Prefer asynchronous resources**](https://developers.google.com/speed/docs/best-practices/rtt#PreferAsyncResources)
* [**Parallelize downloads across hostnames**](https://developers.google.com/speed/docs/best-practices/rtt#ParallelizeDownloads)
* [Minimizing request overhead](https://developers.google.com/speed/docs/best-practices/request) — reducing upload size
* [**Minimize request size**](https://developers.google.com/speed/docs/best-practices/request#MinimizeRequestSize)
* [**Serve static content from a cookieless domain**](https://developers.google.com/speed/docs/best-practices/request#ServeFromCookielessDomain)
* [Minimizing payload size](https://developers.google.com/speed/docs/best-practices/payload) — reducing the size of responses, downloads, and cached pages
* [**Enable compression**](https://developers.google.com/speed/docs/best-practices/payload#GzipCompression)
* [**Remove unused CSS**](https://developers.google.com/speed/docs/best-practices/payload#RemoveUnusedCSS)
* [**Minify JavaScript**](https://developers.google.com/speed/docs/best-practices/payload#MinifyJS)
* [**Minify CSS**](https://developers.google.com/speed/docs/best-practices/payload#MinifyCSS)
* [**Minify HTML**](https://developers.google.com/speed/docs/best-practices/payload#MinifyHTML)
* [**Defer loading of JavaScript**](https://developers.google.com/speed/docs/best-practices/payload#DeferLoadingJS)
* [**Optimize images**](https://developers.google.com/speed/docs/best-practices/payload#CompressImages)
* [**Serve scaled images**](https://developers.google.com/speed/docs/best-practices/payload#ScaleImages)
* [**Serve resources from a consistent URL**](https://developers.google.com/speed/docs/best-practices/payload#duplicate_resources)
* [Optimizing browser rendering](https://developers.google.com/speed/docs/best-practices/rendering) — improving the browser's layout of a page
* [**Use efficient CSS selectors**](https://developers.google.com/speed/docs/best-practices/rendering#UseEfficientCSSSelectors)
* [**Avoid CSS expressions**](https://developers.google.com/speed/docs/best-practices/rendering#AvoidCSSExpressions)
* [**Put CSS in the document head**](https://developers.google.com/speed/docs/best-practices/rendering#PutCSSInHead)
* [**Specify image dimensions**](https://developers.google.com/speed/docs/best-practices/rendering#SpecifyImageDimensions)
* [**Specify a character set**](https://developers.google.com/speed/docs/best-practices/rendering#SpecifyCharsetEarly)
* [Optimizing for mobile](https://developers.google.com/speed/docs/best-practices/mobile)New! — tuning a site for the characteristics of mobile networks and mobile device
* [**Defer parsing of JavaScript**](https://developers.google.com/speed/docs/best-practices/mobile#DeferParsingJS)
* [**Make landing page redirects cacheable**](https://developers.google.com/speed/docs/best-practices/mobile#CacheLandingPageRedirects)

Two of the major speed analysis tools are Google's Page Speed and Yahoo's YSlow. Both require Firefox and the Firebug add-on.

Page Speed and YSlow generally offer the same service, however there are differences in their calculations. Each service analyzes a page using a set of rules that they believe are most relevant to page speed and performance. Most of the rules overlap or are very similar to each other, but in general your scores should be comparable.

The benefit of GTmetrix is that you can analyze your page using both services, giving you different perspectives on how to optimize your code.

* [Avoid a character set in the meta tag](http://gtmetrix.com/avoid-charset-in-meta-tag.html):

# Specifying a character set in a meta tag disables the lookahead downloader in IE8. To improve resource download parallelization, move the character set to the HTTP Content-Type response header

* [Avoid bad requests](http://gtmetrix.com/avoid-bad-requests.html)

# Removing "broken links", or requests that result in 404/410 errors, avoids wasteful requests

* [Avoid CSS @import](http://gtmetrix.com/avoid-css-import.html)

# Using CSS @import in an external stylesheet can add additional delays during the loading of a web page.

* [Avoid CSS expressions (deprecated)](http://gtmetrix.com/avoid-css-expressions-pagespeed.html)

# CSS expressions degrade rendering performance; replacing them with alternatives will improve browser rendering for IE users.

* [Avoid document.write](http://gtmetrix.com/avoid-document.write.html)

# Using document.write() to fetch external resources, especially early in the document, can significantly increase the time it takes to display a web page.

* [Avoid Flash on mobile webpages](http://gtmetrix.com/avoid-flash-on-mobile.html)

# Adobe Flash is not supported on many mobile devices, so its used should be avoided on pages that are viewed on mobile devices. HTML5 is now universally supported on major mobile devices, so if you require Flash-like elements, consider switching to HTML5 alternatives.

* [Avoid landing page redirects](http://gtmetrix.com/avoid-landing-page-redirects.html)

# Redirections on landing pages add delays to the page load and while the redirections are occurring, nothing is shown to the client. In many cases, redirections can be eliminated without changing the function of a page.

* [Combine external CSS](http://gtmetrix.com/combine-external-css.html)

# Combining external stylesheets into as few files as possible cuts down on RTTs and delays in downloading other resources.

* [Combine external JavaScript](http://gtmetrix.com/combine-external-javascript.html)

# Combining external scripts into as few files as possible cuts down on RTTs and delays in downloading other resources

* [Combine images using CSS sprites](http://gtmetrix.com/combine-images-using-css-sprites.html)

# Combining images into as few files as possible using CSS sprites reduces the number of round-trips and delays in downloading other resources, reduces request overhead, and can reduce the total number of bytes downloaded by a web page.

* [Defer loading of JavaScript](http://gtmetrix.com/defer-loading-of-javascript.html)
* [Defer parsing of JavaScript](http://gtmetrix.com/defer-parsing-of-javascript.html)
* [Enable gzip compression](http://gtmetrix.com/enable-gzip-compression.html)
* [Enable Keep-Alive](http://gtmetrix.com/enable-keep-alive.html)
* [Improve server response time](http://gtmetrix.com/improve-server-response-time.html)
* [Inline small CSS](http://gtmetrix.com/inline-small-css.html)
* [Inline small JavaScript](http://gtmetrix.com/inline-small-javascript.html)
* [Leverage browser caching](http://gtmetrix.com/leverage-browser-caching.html)
* [Leverage proxy caching (deprecated)](http://gtmetrix.com/leverage-proxy-caching.html)
* [Make landing page redirects cacheable](http://gtmetrix.com/make-landing-page-redirects-cacheable.html)
* [Minify CSS](http://gtmetrix.com/minify-css.html)
* [Minify HTML](http://gtmetrix.com/minify-html.html)
* [Minify JavaScript](http://gtmetrix.com/minify-javascript.html)
* [Minimize cookie size (deprecated)](http://gtmetrix.com/minimize-cookie-size.html)
* [Minimize DNS lookups](http://gtmetrix.com/minimize-dns-lookups.html)
* [Minimize redirects](http://gtmetrix.com/minimize-redirects.html)
* [Minimize request size](http://gtmetrix.com/minimize-request-size.html)
* [Optimize images](http://gtmetrix.com/optimize-images.html)

You should perform both basic and advanced optimization on all images. Basic optimization includes cropping unnecessary space, reducing color depth to the lowest acceptable level, removing image comments, and saving the image to an appropriate format. You can perform basic optimization with any image editing program, such as [GIMP](http://www.gimp.org/).  Advanced optimization involves further (lossless) compression of JPEG and PNG files.

For JPEG, we recommend [jpegtran](http://jpegclub.org/)or [jpegoptim](http://freshmeat.net/projects/jpegoptim/) (available on Linux only; run with the --strip-all option). For PNG, we recommend [OptiPNG](http://optipng.sourceforge.net/) or [PNGOUT](http://www.advsys.net/ken/util/pngout.htm).

* Reduce the white space around images - some developers use whitespace for padding which is a big no no. Crop your images to remove any whitespace around the image and use CSS to provide padding.
* Use proper file formats. If you have icons, bullets or any graphics that don't have too many colours use a format such as GIF and save the file with lower amounts of colours. If you have more detailed graphics then use JPG file format to save your images and reduce the quality.
* Save your images in the proper dimensions. If you are having to use HTML or CSS to resize your images, stop right there. Save the image in the desired size to reduce the file size.

To resize your images you will have to use some form of program. For basic compression you can use a simple editing program such as GIMP. For more advanced optimization you will have to save specific files in Photoshop, Illustrator or Fireworks

Should give the dimensions in html img tag

Sprite images

Scaled images: use the image of desired size, avoid resizing images from html and css.

Progressive jpeg

Types of images

* [Optimize the order of styles and scripts](http://gtmetrix.com/optimize-the-order-of-styles-and-scripts.html)

##### Put external scripts after external stylesheets if possible

##### Put inline scripts after other resources if possible

* [Parallelize downloads across hostnames](http://gtmetrix.com/parallelize-downloads-across-hostnames.html)

# Serving resources from four or five different hostnames increases parallelization of downloads.

##### Balance parallelizable resources across hostnames

##### Prevent external JS from blocking parallel downloads

##### Always serve a resource from the same hostname

* [Prefer asynchronous resources](http://gtmetrix.com/prefer-asynchronous-resources.html)

##### When a browser parses a traditional script tag, it must wait for the script to download, parse, and execute before rendering any HTML that comes after it. With an asynchronous script, however, the browser can continue parsing and rendering HTML that comes after the async script, without waiting for that script to complete. When a script is loaded asynchronously, it is fetched as soon as possible, but its execution is deferred until the browser's UI thread is not busy doing something else, such as rendering the web page.

* [Put CSS in the document head](http://gtmetrix.com/put-css-in-the-document-head.html)
* [Remove query strings from static resources](http://gtmetrix.com/remove-query-strings-from-static-resources.html)
* [Remove unused CSS](http://gtmetrix.com/remove-unused-css.html)
* [Serve resources from a consistent URL](http://gtmetrix.com/serve-resources-from-a-consistent-url.html)
* [Serve scaled images](http://gtmetrix.com/serve-scaled-images.html)
* [Serve static content from a cookieless domain](http://gtmetrix.com/serve-static-content-from-a-cookieless-domain.html)

##### Enable proxy caching

For resources that rarely change, [set caching headers for browsers and proxies](http://code.google.com/speed/page-speed/docs/caching.html). Because cookies will not be sent for these resources, there is no risk that proxy caches will cache user-specific content.

##### Don't serve early loaded external JS files from the cookieless domain

For JavaScript referenced in the head of the document and needed for page startup, it should be served from the same hostname as the main document. Because most browsers block other downloads and rendering until all JavaScript files have been downloaded, parsed and executed, it's better to avoid the risk of an additional DNS lookup at this point of processing.

* [Specify a cache validator](http://gtmetrix.com/specify-a-cache-validator.html)

##### All static resources should have either a Last-Modified or ETag header. This will allow browsers to take advantage of the full benefit

* [Specify a character set early](http://gtmetrix.com/specify-a-character-set-early.html)

##### Prefer HTTP over meta tag parameters

##### Specify the meta tag at the top of the head section

##### Always specify a content type

##### Be sure to specify the correct character encoding

* [Specify a Vary: Accept-Encoding header](http://gtmetrix.com/specify-a-vary-accept-encoding-header.html)

##### Bugs in some public proxies may lead to compressed versions of your resources being served to users that don't support compression. Specifying the Vary: Accept-Encoding header instructs the proxy to store both a compressed and uncompressed version of the resource.

* [Specify a viewport for mobile browsers](http://gtmetrix.com/specify-a-viewport-for-mobile.html)
* <meta name="viewport"
* content="width=device-width, initial-scale=1, maximum-scale=1"
* [Specify image dimensions](http://gtmetrix.com/specify-image-dimensions.html)

##### Specify dimensions that match those of the images themselves

##### Be sure to specify dimensions on the image element or block-level parent

* [Use an application cache](http://gtmetrix.com/use-an-application-cache.html)

##### Enabling an application cache allows a web page to be used immediately, then updated in the background while the web page is displayed

* [Use efficient CSS selectors](http://gtmetrix.com/use-efficient-css-selectors.html)

##### Avoid a universal key selector

##### Make your rules as specific as possible

##### Avoid using descendant selectors, especially those that specify redundant ancestors

##### Use class selectors instead of descendant selectors

##### Avoid the :hover pseudo-selector for non-link elements for IE clients