SEO

!!! use gtmetrics.com to gide you !!!

I)Serve scaled images

Serving appropriately-sized images can save many bytes of data and improve the performance of your webpage, especially on low-powered (eg. mobile) devices.
Read More
How does your site score on this recommendation?

But my site is responsive...
It can be tempting to serve high-resolution images to all users when catering for displays of varying pixel density. The compressive images technique often yields smaller file sizes with better perceived quality, so that option should definitely be evaluated, however just be aware that higher resolution images will always require more memory and take longer to decode, so it's important to also consider low-powered devices with standard-resolution displays. For this reason, we recommend making use of the srcset attribute. Browser support is very good, and lagging browsers can be helped along with Scott Jehl's picturefill. Take at look at his example usages when determining the most appropriate implementation for your site.
Further details from Google
Sometimes you may want to display the same image in various sizes, so you will serve a single image resource and use HTML or CSS in the containing page to scale it.
For example, you may have a 10 x 10 thumbnail version of a larger 250 x 250 image, and rather than forcing the user to download two separate files, you use markup to resize the thumbnail version. This makes sense if the actual image size matches at least one - the largest - of the instances in the page, in this case 250 x 250 pixels. However, if you serve an image that is larger than the dimensions used in all of the markup instances, you are sending unnecessary bytes over the wire. You should use an image editor to scale images to match the largest size needed in your page, and make sure that you specify those dimensions in the page as well.

II) Laveage borwser caching

Page load times can be significantly improved by asking visitors to save and reuse the files included in your website.

- Reduces page load times for repeat visitors
- Particularly effective on websites where users regularly re-visit the same

areas of the website
- Benefit-cost ratio: high
- Access needed

What is browser caching?

Every time a browser loads a webpage it has to download all the web files to properly display the page. This includes all the HTML, CSS, JavaScript and images.

Some pages might only consist of a few files and be small in size – maybe a couple of kilobytes. For others however there may be a lot of files, and these may add up to be several megabytes large. Twitter.com for example is 3 MB+.

The issue is two fold.

1. These large files take longer to load and can be especially painful if you're on a slow internet connection (or a mobile device).
2. Each file makes a separate request to the server. The more requests your server gets simultaneously the more work it needs to do, only further reducing your page speed.

Browser caching can help by storing some of these files locally in the user's browser. Their first visit to your site will take the same time to load, however when that user revisits your website, refreshes the page, or even moves to a different page of your site, they already have some of the files they need locally.

This means the amount of data the user's browser has to download is less, and fewer requests need to be made to your server. The result? Decreased page load times.

How does it work?

Browser caching works by marking certain pages, or parts of pages, as being needed to be updated at different intervals. Your logo on your website, for instance, is unlikely to change from day to day. By caching this logo image, we can tell the user's browser to only download this image once a week. Every visit that user makes within a week would not require another download of the logo image.

By the web server telling the browser to store these files and not download them when you come back saves your users time and your web server bandwidth.

Why is it important?

The main reason why browser caching is important is because it reduces the load on your web server, which ultimately reduces the load time for your users.

How to leverage browser caching

To enable browser caching you need to edit your HTTP headers to set expiry times for certain types of files.

Configuring Apache to serve the appropriate headers

Find your .htaccess file in the root of your domain. This file is a hidden file but should show up in FTP clients like FileZilla or CORE. You can edit the .htaccess file with notepad or any form of basic text editor.

In this file we will set our caching parameters to tell the browser what types of files to cache.
EXPIRES CACHING
<IfModule mod_expires.c>
ExpiresActive On
ExpiresByType image/jpg "access plus 1 year"
ExpiresByType image/jpeg "access plus 1 year"
ExpiresByType image/gif "access plus 1 year"
ExpiresByType image/png "access plus 1 year"
ExpiresByType text/css "access plus 1 month"
ExpiresByType application/pdf "access plus 1 month"
ExpiresByType text/x-javascript "access plus 1 month"
ExpiresByType application/x-shockwave-flash "access plus 1 month"
ExpiresByType image/x-icon "access plus 1 year"
ExpiresDefault "access plus 2 days"
</IfModule>
EXPIRES CACHING
Depending on your website's files you can set different expiry times. If certain types of files are updated more frequently, you would set an earlier expiry time on them (ie. css files)
When you are done save the file as is and not as a .txt file.
If you are using any form of CMS, cache extensions or plugins might be available.

Recommendations

- Be aggressive with your caching for all static resources
- Expiry at a minimum of one month (recommended: access plus 1 year)
- Don't set your caching more than a year in advance!

Be careful

You want to be careful when enabling browser caching as if you set the parameters too long on certain files, users might not be getting the fresh version of your website after updates.
This is particularly relevant if you are working with a designer to make changes to your website – they might have made the changes but you can't see them yet because the elements that have been changed are cached on your browser.

in timtumb.proxy.php add:

+header('Expires: '.gmdate('D, d M Y H:i:s \G\M\T', time() + (60 * 60 * 24 * 5))); // 5 days

III)Defer parsing of JavaScript

In order to load a page, the browser must parse the contents of all <script> tags, which adds additional time to the page load. By minimizing the amount of

JavaScript needed to render the page, and deferring parsing of unneeded JavaScript until it needs to be executed, you can reduce the initial load time of your page.
Read More
How does your site score on this recommendation?

Details from Google

There are several techniques that can be used to defer parsing of JavaScript. The simplest and preferred technique is to simply Defer loading of JavaScript until it is needed. A second technique is to use the <script async> attribute where appropriate, which prevents parsing from blocking the initial page load by deferring it until the browser's UI thread is not busy doing something else. If neither of these techniques is suitable, there are some additional techniques commonly used in mobile applications, described below:
When building mobile applications, it can be necessary to load all of the JavaScript needed by an application up front, so the application can continue to work when the user is offline. In this case, some applications, such as mobile Gmail, find it useful to load JavaScript in comments and later eval() the JavaScript when it is needed. This approach guarantees that all JavaScript is loaded during the initial page load, while not requiring that JavaScript to be parsed.
An alternative to storing code in comments is storing code in JavaScript string literals. When using this technique, the JavaScript is only parsed when needed, again by calling eval() on the string literal. This technique also allows an application to load JavaScript early, but defer parsing until it is needed.
Note that moving your <script> tags to the end of the page is sub-optimal, since the browser will continue to show a busy indicator until the page has finished parsing this JavaScript. Users may wait until the page load indicator shows that the page load is complete before interacting with the page, so it is important to load JavaScript in a way that minimizes the time it takes for the browser to indicate that the page load is complete.
In our own tests conducted in early 2011, we found that on modern mobile devices, each additional kilobyte of JavaScript adds about 1ms of parse time to the overall page load time. So 100kB of JavaScript included in the initial page load would add 100ms of load time for your users. Because JavaScript must be parsed on every visit to a page, this added load time will be part of every page load, whether loaded from the network, via the browser cache, or in HTML5 offline mode.

IV)**Optimize images**

Reduce the load times of pages by loading appropriately sized images.

- Reduce file sizes based on where images will be displayed
- Resize image files themeselves instead of via CSS
- Save files in appropriate format depending on usage

- Cost benefit ratio: high value
- Access needed

How does your site score on this recommendation?

What is optimizing images for the web?

The images you create in programs like Photoshop and Illustrator look amazing but often the file sizes are very large. This is because the images are made in a format which makes them easier to manipulate in different ways.
With file sizes upwards of a couple of megabytes per image, if you put these files on your website it would be very slow to load.
Optimizing your images for the web means saving or compiling your images in a web friendly format depending on what the image contains.
Images hold data other than just the pixels we see on the screen. This data can add unnecessary size to the image which leads to longer load times as the user waits for the image to download.
In terms of cost versus benefit optimizing your images should be near the top of your page speed optimizations if you don't have them optimized already.

How does it work?

In simple terms optimizing your image works by removing all the unnecessary data that is saved within the image to reduce the file size of the image based on where it is being used in your website.
Optimizing images for the web can reduce your total page load size by up to 80%.
There are two forms of compression that we need to understand, Lossy and Lossless.
Images saved in a lossy format will look slightly different than the original image when uncompressed. Keep in mind that this is only visible at a very close look.
Lossy compression is good for web, because images use small amount of memory, but can be sufficiently like the original image.
Images saved in lossless format retain all the information needed to produce the original image. For this reason these images carry a lot more data and in return are a much large file size.
We also can optimize images for the web by saving them as the appropriate dimensions. Resizing the image on the webpage itself using CSS is helpful but the issue is the web browser will still download the entire original file, then resize it and display it.
Can you imagine take a poster size image and using it as a thumbnail? The little 20px by 20px image would take as long to load as the original poster, when we could just be loading a 20px image the whole time.

Why is it important?

The main reason why it's so important to optimize your images is because 90% of most websites are graphics dependent and therefore there are a lot of image files. Leaving these images uncompressed and in the wrong format can drastically slow down your web page load times.

How to Optimize Your Images

Full optimization of images can be quite an art to perfect as there are such a wide variety of images you might be dealing with. Here are the most common ways to optimize your images for the web.

- 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.

Which tools does PageSpeed use to test this recommendation?

V)**Specify image dimensions**

Specifying a width and height for all images allows for faster rendering by eliminating the need for unnecessary reflows and repaints.

How does your site score on this recommendation?

Page Speed currently only detects image dimensions that are specified via the image attributes. If you are specifying the dimensions via CSS, then you can safely ignore this recommendation.

Details from Google

When the browser lays out the page, it needs to be able to flow around replaceable elements such as images. It can begin to render a page even before images are downloaded, provided that it knows the dimensions to wrap non-replaceable elements around. If no dimensions are specified in the containing document, or if the dimensions specified don't match those of the actual images, the browser will require a reflow and repaint once the images are downloaded. To prevent reflows, specify the width and height of all images, either in the HTML <img> tag, or in CSS.

VI) **No pop-ups**

VII) Images will be named like :

notre-dame-paris.png because word press will use it if you do not specify an alt or title to your image

The alt of the <img /> tag should be very descriptive, not very long, containing at least a keyword of the poral

---

Google and fb conversion rate :
Google conversion with php is better (not js) because if the user closes page we lose that conversion - is already done because we use google script.

Conversion for facebook integration takes some time because we need access to advertiser account - after that we will see.

Blank pages when testing on 3G - signal lost, very bad signal

no additional queries to the data base when we including different layouts, for example no additional query for the slider
fraud and attacks top ten mobile vulnerabilities OWASP Mobile Top 10 risks

malware on android - adduri compromise

for stealing whatsApp db: upload
 http:

1.weak server side controls
2.Insecure Data storage no on SD card, but on Data storage where the access on that folder is exclusive for that up
3. Inssuficient Transport Layer Protection
4. Unintended Data Leakage (including data storage)
5. social engineering

VIII) Title atribute for all links (search for href= in your site)

When creating your link titles, optimize for keyword phrases you’re targeting on the linked to page (just as you would with anchor text). Search engines only use them in consideration to the page being linked to, not the page the link is on.