**Describe how the performance of a website can be improved.**

**1. Reduce HTTP requests**

Every fetched resource with either a HTTP or HTTPS web address decreases speed, which means: the fewer the number of requests, the better your website will perform. You can speed up your website by simplifying the design and coding elements. Here are some techniques:

* Use image spriting: Image spriting combines multiple images into one single image file. It uses background positioning to display the required portion of the image.
* Connect CSS and JavaScript: Concatenating CSS and JavaScript is crucial. Creating one master file is always better than having multiple files to perform different tasks.
* Use CSS instead of images whenever possible: It is tempting to use lots of images instead of CSS because it can make a developer's job easier, but this strategy impacts performance. Instead, try to get the look you want using CSS coding instead of images.
* Don’t over-rely on JavaScript: Using too much JavaScript will cause HTTP overload and make your website harder to maintain. It’s better to use JavaScript only when it’s necessary.

**2. Use fewer redirects**

When you redirect to another webpage, it will create an additional HTTP request. This increases page loading time.  
  
For example, does your mobile redirect looks like this: yourdomain.com → www.yourdomain.com → m.yourdomain.com → m.yourdomain.com/home? If so, your website will load much more slowly because of these additional redirects.

**3. Minify resources**

“Minify” is a verb from programming jargon that means removing undesired code from a CSS or JavaScript file. Minifying code is an excellent way to improve web page performance.  
  
If your code contains extra white spaces, page breaks, and line breaks, remove these elements to enhance website performance.

**4. Enable Gzip compression**

One of the fastest ways to increase page loading speed is to use Gzip compression. It compresses HTML and CSS files before sending them to the browser. When a browser hits a web server, it checks for Gzip compression.  
  
If the server is Gzip compression-enabled, it will fetch a Gzip file, which is significantly smaller than an uncompressed version. We recommend enabling Gzip compression as a standard practice because it can reduce page size by up to 70%.

**5. Optimize images**

New to image optimization? Read [our post](http://blog.tbhcreative.com/2014/03/optimizing-images-for-your-website.html) for tips and tricks.

Many websites are heavily dependent on graphics. If your images are not compressed, it will slow down your website’s performance.  
  
Optimizing images means saving your image in a web-friendly format.  
  
According to the [HTTP Archive](http://httparchive.org/interesting.php#bytesperpage), image files contribute to around 62% of a website's page loading speed on average. This means that image optimization is essential when it comes to good website performance.

**6. Optimize CSS and JavaScript**

Writing clean and well-structured CSS code is crucial for performance because poorly written CSS will increase page loading time which leads to an inferior user experience.  
  
JavaScript is used to develop functionality-rich websites. Complex and poorly structured JavaScript can affect performance. For instance, users may encounter a blank screen if a page doesn't load correctly due to JavaScript problems.

**7. Code better**

Try writing cleaner code. Do so by reducing the lines of coding as much as you can, optimizing database queries, using design patterns, and avoiding recursion. Structured and readable code always pays off. It improves maintainability and decreases HTTP overload, all of which will enhance performance significantly.

**8. Don’t forget your server**

Make sure your server administrator tunes the server regularly so that your website can accommodate higher traffic growth as needed. Optimal server performance ensures a better user experience. Monitor server resource utilization and make changes to the server capacity.

**9. Take advantage of browser caching**

Cache mechanisms operate either on the server-side or in the user’s browser. You can speed up load time on future visits for your users if you store your website’s static files (such as your logo and banner art) locally.  
  
When your user revisits or refreshes the web page, she will already having a portion of the web page stored on her local computer, improving website performance. Since the data stored in the browser does not have to be downloaded again, fewer server requests have to be made.

**10. Consider using CDN**

A content delivery network (CDN) is a distributed system that handles the delivery of static content like HTML, CSS, JavaScript, images, and videos to users based on geographic location. CDN is a great tool to use in order to improve performance because it segregates static content away from the application server.  
  
Some prominent providers of CDN include Akamai, Google, and AWS. AWS and Google’s CDN offerings are inexpensive, but Akamai is considered the most dependable. However, its reliability comes at a steeper cost. Lesser known players include Cloudflare, MaxCDN, and Rackspace.  
  
[Every second matters.](https://blog.radware.com/applicationdelivery/wpo/2014/04/web-page-speed-affect-conversions-infographic/) While it’s good to think about functionality, content, and aesthetics when building your website, if it loads slowly, those things won’t matter because your users will leave before they experience them. That’s why website performance should always be a priority.  
  
To ensure the best experience for your users, focus on concise coding, server-side tuning and optimization, image quality and compression, and adopting technologies like CDN to serve static content.