SEO

Search engine optimization (SEO) is often about making small modifications to parts of your website. When viewed individually, these changes might seem like incremental improvements, but when combined with other optimizations, they could have a noticeable impact on your site's user experience and performance in organic search results. You're likely already familiar with many of the topics in this guide, because they're essential ingredients for any web page, but you may not be making the most out of them.

# Inhalt

[1 Inhalt 1](#_Toc126080247)

[2 SEO 4](#_Toc126080248)

[2.1 Useful Terms 4](#_Toc126080249)

[2.1.1 Crawl 4](#_Toc126080250)

[2.1.2 Crawler 4](#_Toc126080251)

[2.1.3 Google Hummingbird 4](#_Toc126080252)

[2.1.4 Google My Business 4](#_Toc126080253)

[2.1.5 Google business profile 4](#_Toc126080254)

[2.1.6 Google images 4](#_Toc126080255)

[2.1.7 Google lens 4](#_Toc126080256)

[2.1.8 Google maps 4](#_Toc126080257)

[2.1.9 Google merchant center 4](#_Toc126080258)

[2.1.10 Google search 4](#_Toc126080259)

[2.1.11 Google shopping tab 4](#_Toc126080260)

[2.1.12 Googlebot 4](#_Toc126080261)

[2.1.13 Index 4](#_Toc126080262)

[2.1.14 Keyword mapping 4](#_Toc126080263)

[2.1.15 Keyword stuffing 4](#_Toc126080264)

[2.1.16 Branded Keywords 4](#_Toc126080265)

[2.1.17 Backlinks 4](#_Toc126080266)

[2.1.18 Robots.txt 4](#_Toc126080267)

[2.2 Best Practices 4](#_Toc126080268)

[2.2.1 Keywords 4](#_Toc126080269)

[2.2.2 *Related Keywords (LSI)* 4](#_Toc126080270)

[2.2.3 Tags 4](#_Toc126080271)

[2.2.4 Product Description 4](#_Toc126080272)

[2.2.5 Links 4](#_Toc126080273)

[2.2.6 Pogosticking 4](#_Toc126080274)

[2.2.7 Sitemap 4](#_Toc126080275)

[2.2.8 Structured Data 4](#_Toc126080276)

[2.2.9 Featured Snippets 4](#_Toc126080277)

[2.2.10 Product Reviews 4](#_Toc126080278)

[2.2.11 “You might also like” 4](#_Toc126080279)

[2.2.12 FAQ/Q&A 4](#_Toc126080280)

[2.2.13 Blogging 4](#_Toc126080281)

[3 Performance 4](#_Toc126080282)

[3.1 Userful Terms 4](#_Toc126080283)

[3.1.1 Cumulative Layout Shift 4](#_Toc126080284)

[3.1.2 First Contentful Paint 4](#_Toc126080285)

[3.1.3 Largest Contentful Paint 4](#_Toc126080286)

[3.1.4 Speed Index 4](#_Toc126080287)

[3.1.5 Time to Interactive 4](#_Toc126080288)

[3.1.6 Total Blocking Time 4](#_Toc126080289)

[3.1.7 Apache PageSpeed 4](#_Toc126080290)

[3.2 Best Practices 4](#_Toc126080291)

[3.2.1 HTTP 5](#_Toc126080292)

[3.2.2 Images 6](#_Toc126080293)

[3.2.3 JavaScript 8](#_Toc126080294)

[3.2.4 CSS 10](#_Toc126080295)

[4 Accessibility 12](#_Toc126080296)

[4.1 Best Practices 12](#_Toc126080297)

[4.1.1 Site structure 12](#_Toc126080298)

[4.1.2 Old Pages 12](#_Toc126080299)

[4.1.3 Error Pages 12](#_Toc126080300)

[4.1.4 Breadcrumb Navigation 12](#_Toc126080301)

[4.1.5 Form 13](#_Toc126080302)

[4.1.6 Lists 13](#_Toc126080303)

[4.1.7 Links 13](#_Toc126080304)

[4.1.8 Don't place block-level element within inline elements 13](#_Toc126080305)

[4.1.9 Avoid using <b> and <i> to bold and italicize texts on a web page 13](#_Toc126080306)

[4.1.10 Headings 13](#_Toc126080307)

[4.1.11 Visual Content 14](#_Toc126080308)

[4.1.12 Layout Shifts 14](#_Toc126080309)

[5 Tools 15](#_Toc126080310)

[5.1 Google Optimize 15](#_Toc126080311)

[5.2 Google Search Console 15](#_Toc126080312)

[5.3 Google Shopping 15](#_Toc126080313)

[5.4 Google Mobile Friendly 15](#_Toc126080314)

[5.5 AHREFS 15](#_Toc126080315)

[5.6 Screaming Frog 15](#_Toc126080316)

[5.7 Lighthouse 15](#_Toc126080317)

# SEO

## Useful Terms

### Crawl

The process of looking for new or updated web pages. Google discovers URLs by following links, by reading sitemaps, and by many other means. *Google crawls the web, looking for new pages, then indexes them (when appropriate).*

### Crawler

Automated software that crawls (fetches) pages from the web and indexes them.

<https://developers.google.com/search/docs/crawling-indexing/overview-google-crawlers>

### Google Hummingbird

Unlike previous search algorithms, which would focus on each individual word in the search query, [Hummingbird](https://en.wikipedia.org/wiki/Google_Hummingbird) considers the context of the different words together, with the goal that pages matching the meaning do better, rather than pages matching just a few words.

### Google My Business

[Google My Business](https://www.google.com/business/) makes it easy for local businesses to promote their products and services online by increasing visibility and improving their chances of showing up in search results.

### Google business profile

### Google images

### Google lens

### Google maps

### Google merchant center

### Google search

### Google shopping tab

### Googlebot

The generic name of [Google's crawler](https://developers.google.com/search/docs/advanced/crawling/googlebot). Googlebot crawls the web constantly.

### Index

Google stores all web pages that it knows about in its *index*. The index entry for each page describes the content and location (URL) of that page. *To index* is when Google fetches a page, reads it, and adds it to the index: *Google indexed several pages on my site today*.

### Keyword mapping

Keyword mapping is a strategy that occurs across a website. Once you have a list of keywords relevant to the website’s content, you can then analyse each keyword map out which page on the website each keyword should go to.

The idea behind keyword mapping is that it prevents you from loading all the keywords on the front page only or using identical sets of keywords for each page.

### Keyword stuffing

Keyword stuffing is the practice of loading an article with keywords but at the expense of the content. It’s important to avoid doing this.

### Branded Keywords

Branded keywords are phrases directly associated with your brand, products, and services.

### Backlinks

Backlinks are links on websites other than your own that go back to a page on your website.

### Robots.txt

A robots.txt file tells search engines whether they can access and therefore crawl parts of your site. This file, which must be named robots.txt, is placed in the root directory of your site. It is possible that pages blocked by robots.txt can still be crawled, so for sensitive pages, use a more secure method.

## Best Practices

### Keywords

Keywords are the words and phrases that people type into search engines to find what they're looking for.

Including keywords in titles, headers, subheaders, title tags, meta descriptions, image tags, and alt text are often good practices.

Keywords should also be sprinkled throughout the body of the content. Including primary keywords near the top will usually rank a page higher.

Keywords **must** appear in the:

* <title> tag
* URL
* First 100 Words of your content

#### Long Tail Keywords

Long tail keywords are search terms with relatively low search volume and competition levels. Also, long tail terms tend to be longer in length (3+ words) than most other keyword types.

To optimize content for the right people, it's important to be specific. "Sales training" is good, but "B2B sales training" is better, and "books on B2B sales training" beats them all. Longtail phrases also tend to have lower competition in search results, increasing the odds of higher page ranks in Google.

#### Google SERP – Related searchesRelated Keywords (LSI)

LSI keywords are search terms related to the main keyword you are targeting.

* To rank in [Google today](#_Google_Hummingbird), you also need to add synonyms and other related keywords to your content.
* Add a few “Searches related to…” terms to your content.

### Tags

Your page title and meta description might be the first thing your audience ever sees. Users might decide whether to click or skip your page based on these alone. That’s a lot of power for so few words, so make sure they’re impactful. The tags should be unique for every page.



#### Title tag

Ensure they’re unique and of the appropriate length (30 to 60 characters). Make sure you include details, like:

* The brand of the product
* The name of the product
* The model number
* Other important information

#### Heading tag

Ensure your page has properly formatted headings and subheadings (H1, H2, etc.) in HTML instead of headers in bolded regular text. Search engines give special weight to words that appear in header tags when determining what your site is about.

#### Rel Canonical

A canonical link element is an HTML element that helps webmasters prevent duplicate content issues in search engine optimization by specifying the "canonical" or "preferred" version of a web page.

#### Meta tag

The <meta> tag defines metadata about an HTML document.

##### Description <meta name=”description”>

These are short descriptions (100 to 155 characters) that show up below your webpage in search results. The meta description should be unique, concise and paint a clear picture of your page’s content.

##### Keywords <meta name=”keywords”>

##### Author <meta name=”author”>

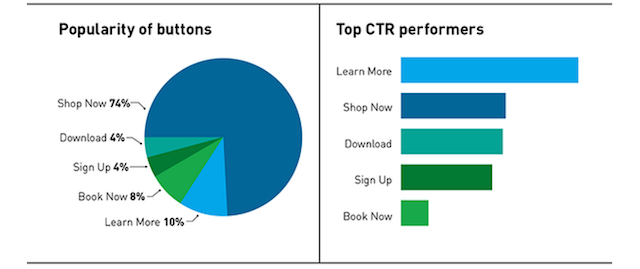
### Product Description

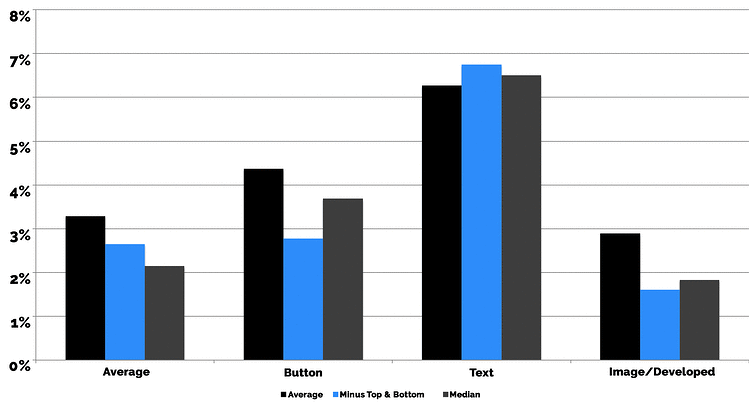
A product description should be unique, never copied from the original seller and 1000+ Words.

### Links

#### Call To Action (CTA)

Call to action is a marketing term for any design to prompt an immediate response or encourage an immediate sale. A CTA most often refers to the use of words or phrases that can be incorporated into sales scripts, advertising messages, or web pages, which compel an audience to act in a specific way.



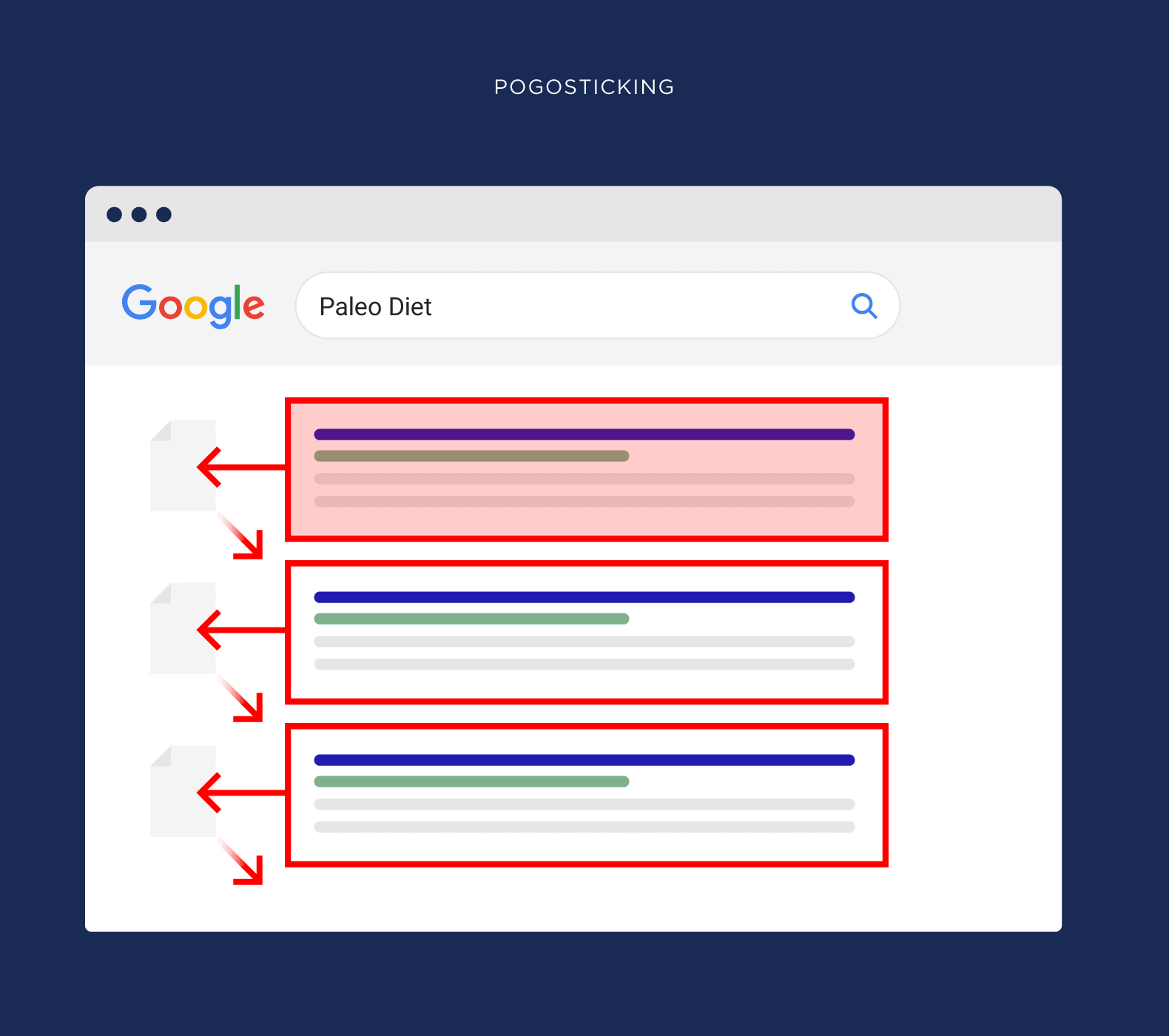


#### Internal linking

The more links to something you have, the more Google thinks that the article is of high quality. This applies to both internal and external links

### Pogosticking

Pogosticking is when a Google user clicks on your site, then “pogosticks” back to the search results to find something that actually helps them.



* Pogosticking can break your google rankings
* How do you keep users on your site longer? Use lots of bullets and subheadings. When your content is easy to read, people will spend more time on your site.

### Sitemap

A *sitemap* is a file where you provide information about the pages, videos, and other files on your site, and the relationships between them. Search engines like Google read this file to crawl your site more efficiently. A sitemap can improve the crawling of larger or more complex sites, or more specialized files. Regular updates are advised.

<https://developers.google.com/search/docs/crawling-indexing/sitemaps/image-sitemaps>

There are four main types of sitemaps:

* Normal XML Sitemap: This by far the most common type of sitemap. It’s usually in the form of an XML Sitemap that links to different pages on your website.
* Video Sitemap: Used specifically to help Google understand video content on your page.
* News Sitemap: Helps Google find content on sites that are approved for Google News.
* Image Sitemap: Helps Google find all of the images hosted on your site.

### Structured Data

[Structured data](https://developers.google.com/search/docs/advanced/structured-data/intro-structured-data) is a standardized format for providing information about a page and classifying the page content; for example, on a recipe page, what are the ingredients, the cooking time and temperature, the calories, and so on.

<schema.org>

#### Formats

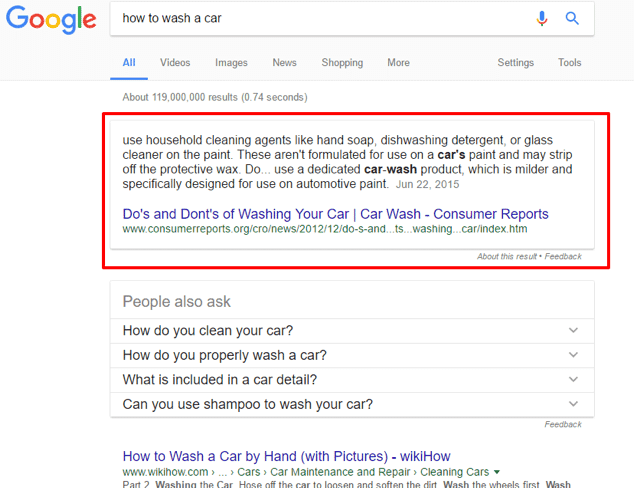
* [JSON-LD](https://json-ld.org/) (Recommended)
* [Microdata](https://html.spec.whatwg.org/multipage/microdata.html#microdata)
* [RDFa](https://rdfa.info/)

#### Ecommerce-specific Structured Data

* LocalBusiness
* Product
* Review
* HowTo
* FAQPage
* BreadcrumbList
* WebSite
* VideoObject

<https://developers.google.com/search/docs/specialty/ecommerce/include-structured-data-relevant-to-ecommerce>

### Featured Snippets

Featured snippets are highlighted excerpts of text that appear at the top of a Google search results page in what is known as 'Position 0'. They provide users with a quick answer to their search query.

Featured snippets can drive more organic traffic to a website since the CTR is much higher than ranking at a lower position than 0.

If you already have a good ranking for a keyword (positions 1 – 5), your website may be eligible to show in the featured snippet box.

To increase your chances, provide a direct answer to a question or definition by summarizing it in a couple of lines but within a single paragraph (<p>) tag. The higher you place this paragraph in your content, the better.

Having an image with a relevant ALT text also helps.

The easiest and fastest way to check if your website has any featured snippet rankings is through SEMRush.

You can opt-out of featured snippets by adding a meta tag to the head of your page.

### Product Reviews

Structured Data

### “You might also like”

Internal linking

### FAQ/Q&A

FAQ/Q&A means Frequently asked questions/Questions and answers. Sections like these in your website help it to appear in “Featured Snippets”. It’s also a possibility to include internal linking. There exists specific schema markup for Questions and Answers sections.

* Customer service - Check in with your customer support/sales teams and simply ask them about common questions customers keep asking
* Site search - See if your site has an internal search function; if so, check what kind of things people are searching for
* Google Search Console - Look at GSC queries to see what question-based phrases are getting clicks
* People Also Ask - Check related [PAA boxes](https://ahrefs.com/blog/people-also-ask/) on the SERPs
* Quora and Reddit - See what common questions are being discussed in online communities in your niche

### Blogging

Internal linking

# Performance

## Userful Terms

### Cumulative Layout Shift

CLS indicates how much layout shift is experienced by visitors as your page loads. Aim for a score of 0.1 or less.

### First Contentful Paint

How quickly content like text or images are painted onto your page. Aim for 0.9s or less.

### Largest Contentful Paint

LCP measures how long it takes for the **largest content element** (e.g. a hero image or heading text) on your page to become visible within your visitors' viewport. Aim for 1.2s or less

### Speed Index

How quickly the contents of your page are visibly populated. Aim for 1.3s or less

### Time to Interactive

How long it takes for your page to become fully interactive. Aim for 2.5s or less.

### Total Blocking Time

TBT tells you how much time is blocked by scripts during your page loading process. Aim for 150ms or less.

### Apache PageSpeed

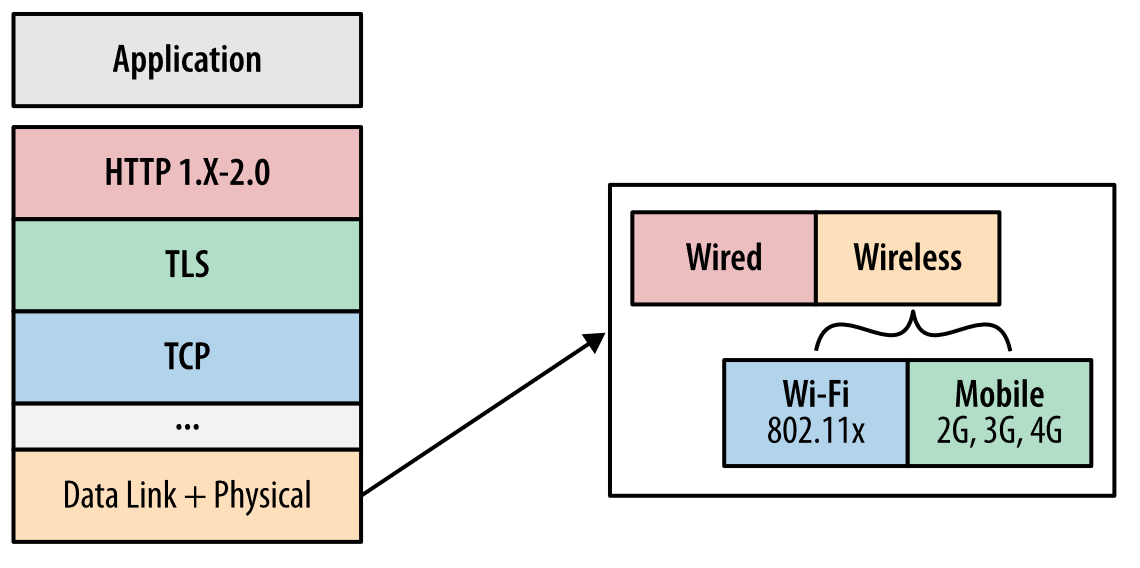
The PageSpeed Modules, [mod\_pagespeed](https://github.com/apache/incubator-pagespeed-mod) and [ngx\_pagespeed](https://github.com/apache/incubator-pagespeed-ngx), are open-source webserver modules that [optimize your site automatically](https://www.modpagespeed.com/doc/filters).

## Best Practices

<https://hpbn.co/>

<https://developer.mozilla.org/en-US/docs/Learn/Performance/Web_Performance_Basics>

<https://web.dev/image-cdns/>

High-performance browser networking relies on a host of networking technologies and the overall performance of our applications is the sum total of each of their parts.

We cannot control the network weather between the client and server, nor the client hardware or the configuration of their device, but the rest is in our hands: TCP and TLS optimizations on the server, and dozens of application optimizations to account for the peculiarities of the different physical layers, versions of HTTP protocol in use, as well as general application best practices.

### HTTP

<https://hpbn.co/optimizing-application-delivery/>

#### Reduce unnecessary network latency and minimize the number of transferred bytes

Regardless of the type of network or the type or version of the networking protocols in use, all applications should always seek to eliminate or reduce unnecessary network latency and minimize the number of transferred bytes. These two simple rules are the foundation for all of the evergreen performance best practices:

##### Reduce DNS lookups

Every hostname resolution requires a network roundtrip, imposing latency on the request and blocking the request while the lookup is in progress.

##### Reuse TCP connections

Leverage connection keepalive whenever possible to eliminate the TCP handshake and slow-start latency overhead; see [Slow-Start](https://hpbn.co/building-blocks-of-tcp/#slow-start).

##### Minimize number of HTTP redirects

HTTP redirects impose high latency overhead—e.g., a single redirect to a different origin can result in DNS, TCP, TLS, and request-response roundtrips that can add hundreds to thousands of milliseconds of delay. The optimal number of redirects is zero.

##### Reduce roundtrip times

Locating servers closer to the user improves protocol performance by reducing roundtrip times (e.g., faster TCP and TLS handshakes), and improves the transfer throughput of static and dynamic content; see [Uncached Origin Fetch](https://hpbn.co/transport-layer-security-tls/" \l "uncached-origin-fetch).

##### Eliminate unnecessary resources

No request is faster than a request not made. Be vigilant about auditing and removing unnecessary resources.

#### Caching and Compression

By this point, all of these recommendations should require no explanation: latency is the bottleneck, and the fastest byte is a byte not sent. However, HTTP provides some additional mechanisms, such as caching and compression, as well as its set of version-specific performance quirks:

##### Cache resources on the client

Application resources should be cached to avoid re-requesting the same bytes each time the resources are required.

##### Compress assets during transfer

Application resources should be transferred with the minimum number of bytes: always apply the best compression method for each transferred asset.

##### Eliminate unnecessary request bytes

Reducing the transferred HTTP header data (e.g., HTTP cookies) can save entire roundtrips of network latency.

##### Parallelize request and response processing

Request and response queuing latency, both on the client and server, often goes unnoticed, but contributes significant and unnecessary latency delays.

##### Apply protocol-specific optimizations

HTTP/1.x offers limited parallelism, which requires that we bundle resources, split delivery across domains, and more. By contrast, HTTP/2 performs best when a single connection is used, and HTTP/1.x specific optimizations are removed.

### Images

Set an explicit width and height on image elements to reduce layout shifts.

#### CDNs

[Using image CDNs](https://web.dev/image-cdns/)

Image content delivery networks (CDNs) are excellent at optimizing images. Switching to an image CDN can yield a 40–80% savings in image file size.

For images loaded from an image CDN, an image URL indicates not only which image to load, but also parameters like size, format, and quality. This makes it easy to create variations of an image for different use cases.

#### Compression

[Compressing images](https://web.dev/use-imagemin-to-compress-images/)

* AVIF is a solid first choice if lossy, low-fidelity compression is acceptable and saving bandwidth is the number one priority. Assuming encode/decode speeds meet your needs.
* WebP is more widely supported and may be used for rendering regular images where advanced features like wide color gamut or text overlays are not required.
* AVIF may not be able to compress non-photographic images as well as PNG or lossless WebP. Compression savings from WebP may be lower than JPEG for high-fidelity lossy compression.

[Using WebP images](https://web.dev/serve-images-webp/)

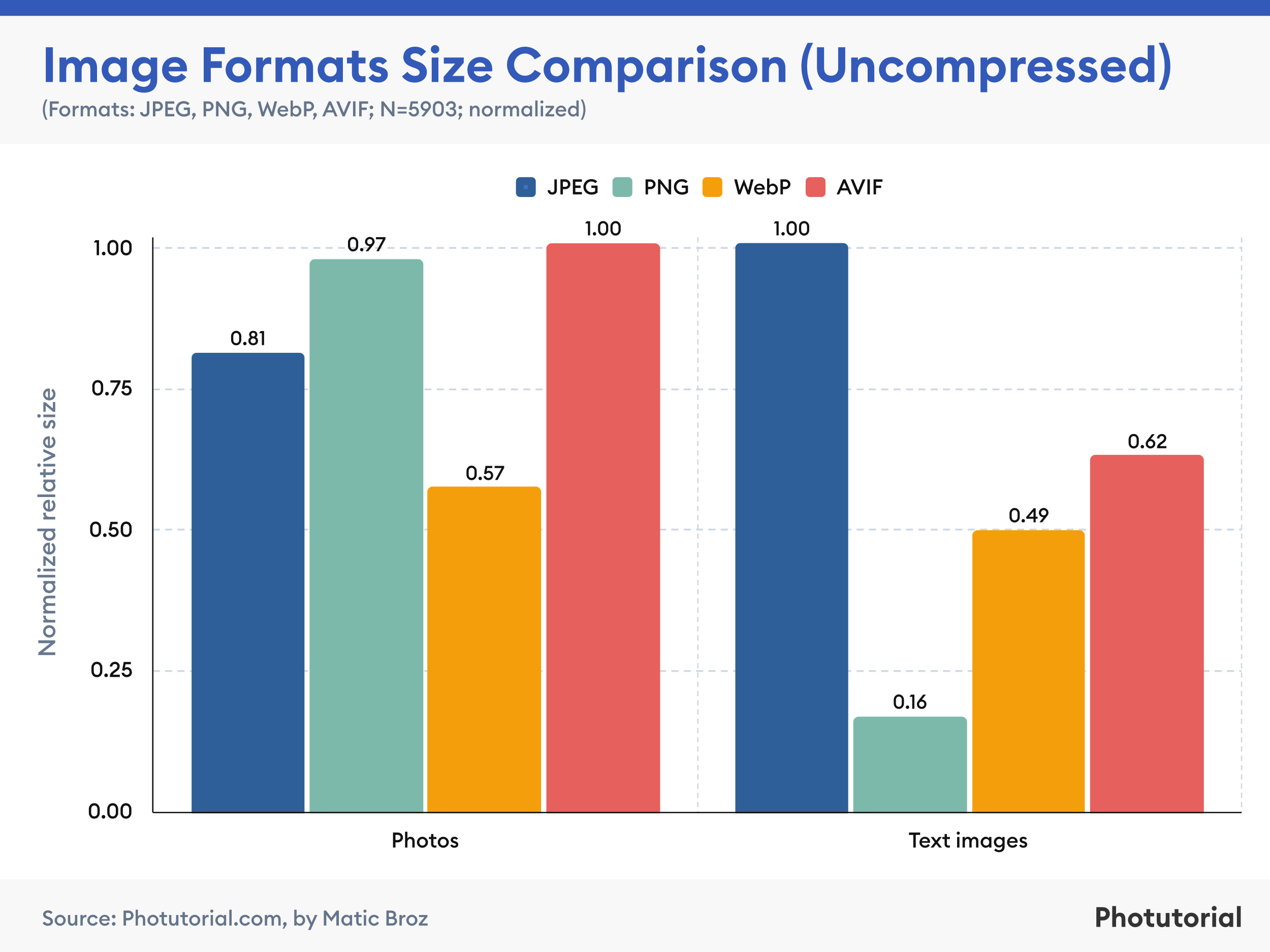
WebP images are smaller than their JPEG and PNG counterparts—usually on the magnitude of a 25–35% reduction in filesize. This decreases page sizes and improves performance.

Before:

<img src="flower.jpg" alt="">

After:

<picture>  
 <source type="image/webp" srcset="flower.webp">  
 <source type="image/jpeg" srcset="flower.jpg">  
 <img src="flower.jpg" alt="">  
</picture>



#### Lazy loading

[Lazy loading images](https://web.dev/use-lazysizes-to-lazyload-images/)

You can use the “loading” attribute to completely defer the loading of offscreen images that can be reached by scrolling:

<img src="image.png" loading="lazy" alt="…" width="200" height="200">

#### Responsive

[Serving responsive images](https://web.dev/serve-responsive-images/)

Serving desktop-sized images to mobile devices can use 2–4x more data than needed. Instead of a "one-size-fits-all" approach to images, serve different image sizes to different devices.

Specify multiple image versions and the browser will choose the best one to use:

Before:

<img src="flower-large.jpg">

After:

<img src="flower-large.jpg" srcset="flower-small.jpg 480w, flower-large.jpg 1080w" sizes="50vw">

[Serving images with correct dimensions](https://web.dev/serve-images-with-correct-dimensions/)

#### Preload

<https://web.dev/preload-responsive-images/>

### JavaScript

By default, references to external JavaScript files will block the page from rendering while they are fetched and executed. Often, these files can be loaded in a different manner, freeing up the page to visually render sooner.

In modern websites, scripts are often “heavier” than HTML: their download size is larger, and processing time is also longer.

When the browser loads HTML and comes across a <script>...</script> tag, it can’t continue building the DOM. It must execute the script right now. The same happens for external scripts <script src="..."></script>: the browser must wait for the script to download, execute the downloaded script, and only then can it process the rest of the page.

That leads to two important issues:

* Scripts can’t see DOM elements below them, so they can’t add handlers etc.
* If there’s a bulky script at the top of the page, it “blocks the page”. Users can’t see the page content till it downloads and runs:

#### Minify

#### Defer

A script that will be downloaded in parallel to parsing the page, and executed after the page has finished parsing. Deferred scripts still execute in the order they are defined in source.

<script src="demo\_defer.js" defer></script>

#### Async

Similar to “defer” but Async scripts are not guaranteed to execute in the order they are defined in source.

#### Split

#### Inline

This experiment embeds the contents of specified external scripts directly into the HTML within a script element. This increases the size of the HTML, but can often allow page page to display sooner by avoiding server round trips.Example implementation:

<script>/\* contents from /.resources/shop/js/ecommerce.js here...\*/</script>

#### Avoid chaining critical requests

[Critical request chains](https://web.dev/critical-rendering-path/) are series of dependent network requests important for page rendering. The greater the length of the chains and the larger the download sizes, the more significant the impact on page load performance. These Critical Request Chains show you what resources are loaded with a high priority. Consider reducing the length of chains, reducing the download size of resources, or deferring the download of unnecessary resources to improve page load.

* Minimize the number of critical resources: eliminate them, defer their download, mark them as async, and so on.
* Optimize the number of critical bytes to reduce the download time (number of round trips).
* Optimize the order in which the remaining critical resources are loaded: download all critical assets as early as possible to shorten the critical path length.

#### External Scripts

If a third-party script is slowing down your page load, you have several options to improve performance:

* Load the script using the async or defer attribute to avoid blocking document parsing.
* Consider self-hosting the script if the third-party server is slow.
* Consider removing the script if it doesn't add clear value to your site.
* Consider [Resource Hints](https://developer.chrome.com/docs/lighthouse/performance/uses-rel-preconnect/) like <link rel=preconnect> or <link rel=dns-prefetch> to perform a DNS lookup for domains hosting third-party scripts.
* "Sandbox" scripts with an iframe

### CSS

By default, references to external CSS files will block the page from rendering while they are fetched and executed. Sometimes these files should block rendering, but can be inlined to avoid additional round-trips while the page is waiting to render. Sometimes, such as with stylesheets that are only used for loading custom fonts, inline or async CSS can greatly improve perceived performance.

The [contain](https://developer.mozilla.org/en-US/docs/Web/CSS/contain) CSS property allows an author to indicate that an element and its contents are, as much as possible, *independent* of the rest of the document tree. This allows the browser to recalculate layout, style, paint, size, or any combination of them for a limited area of the DOM and not the entire page.

#### Minify

#### Inline external CSS

This experiment embeds the contents of specified external stylesheets directly into the HTML within a <style> element. This increases the size of the HTML, but can often allow page page to display sooner by avoiding server round trips

#### Async

To load CSS Files asynchronously in both Chrome and Firefox, we can use “preload” browser hint and “media=’print'” attribute along with onload event feature in a ordered way.

<link rel="preload" href="style.css" as="style" onload="this.rel='stylesheet'">

<link rel="stylesheet" href="style.css" media="print" onload="this.media='all'">

Move the not-immediately used styles into separate file

<link rel="stylesheet" href="print.css" media="screen and (max-width: 480px)" />

If you're going to load CSS asynchronously, it's generally recommended that you inline [critical CSS](https://www.smashingmagazine.com/2015/08/understanding-critical-css/), since CSS is a render-blocking resource [for a reason](https://developers.google.com/web/fundamentals/performance/critical-rendering-path/render-blocking-css).

[Credit to filament group](https://www.filamentgroup.com/lab/load-css-simpler/) for their many async CSS solutions.

This approach may not work with content security policy enabled.

#### Preload

<https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/rel/preload>

The preload value of the [<link>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link) element's [rel](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link#attr-rel) attribute lets you declare fetch requests in the HTML's [<head>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/head), specifying resources that your page will need very soon, which you want to start loading early in the page lifecycle, before browsers' main rendering machinery kicks in. This ensures they are available earlier and are less likely to block the page's render, improving performance. Even though the name contains the term *load*, it doesn't load and execute the script but only schedules it to be downloaded and cached with a higher priority.

<link rel="stylesheet" href="styles/main.css" />

#### Fonts

When fonts are loaded with default display settings, like font-display="block", browsers will hide text entirely for several seconds instead of showing text with a fallback font.

Applied to the [@font-face](https://developer.mozilla.org/en-US/docs/Web/CSS/@font-face) rule, the [font-display](https://developer.mozilla.org/en-US/docs/Web/CSS/@font-face/font-display) property defines how font files are loaded and displayed by the browser, allowing text to appear with a fallback font while a font loads, or fails to load. This improves performance by making the text visible instead of having a blank screen, with a trade-off being a flash of unstyled text.

@font-face {

font-family: someFont;

src: url(/path/to/fonts/someFont.woff) format("woff");

font-weight: 400;

font-style: normal;

font-display: fallback;

}

EOT and TTF formats are not compressed by default. Apply compression such as GZIP or Brotli for these file types. Use WOFF and WOFF2. These formats have compression built in.

#### Animations

To improve performance, the node being animated can be moved off the main thread and onto the GPU. When an element is promoted as a layer, also known as composited, animating transform properties is done in the GPU, resulting in improved performance, especially on mobile.

The CSS [will-change](https://developer.mozilla.org/en-US/docs/Web/CSS/will-change) property hints to browsers how an element is expected to change. Browsers may set up optimizations before an element is actually changed. These kinds of optimizations can increase the responsiveness of a page by doing potentially expensive work before it is actually required.

.element {

will-change: opacity, transform;

}

# Accessibility

## Best Practices

### Site structure

#### URL

Your URL should contain a keyword that you want your page to rank for.

Use hyphens as “word separators” in your URL. Not underscores.

Example: https://backlinko.com/seo-site-audit

Your URLs should be short and follow a simple structure.

Bad example



Good Example



### HTML Elements

#### Don't place block-level element within inline elements

<a href="#" >

<p> Visit freecodecamp </p>

</a>

<p>

Visit <a href="www.freecodecamp. org" target="\_blank">FreecodeCamp</a>

to learn Javascript

</p>

#### Headings

Use only one <h1> element per page

Do not skip heading levels

#### Form

Form elements must have labels.

#### Lists

<Ul> And <Ol> Must Only Directly Contain <Li>, <Script> Or <Template> Elements

#### Links

Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. [Learn more](https://web.dev/link-name/?utm_source=lighthouse&utm_medium=lr).

#### Avoid using <b> and <i> to bold and italicize texts on a web page

You shouldn't use <b> and <i> for bolding and italics because they have no semantic meaning. Use the font-weight CSS property or use the <strong> and the <em> tags instead.

### Old Pages

Zombie pages are pages on a website that generate little or no traffic and are difficult or impossible to access through search engine results

When you delete Zombie Pages, you can get higher rankings and more Google traffic.

### Error Pages

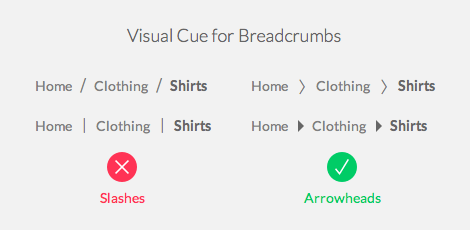
Show useful 404 pages. Consider including a link back to your root page and providing links to popular or related content on your site

Avoid:

* Allowing your 404 pages to be indexed in search engines (make sure that your web server is configured to give a 404 HTTP status code or—in the case of JavaScript-based sites—include the noindex tag when non-existent pages are requested).
* Blocking 404 pages from being crawled through the robots.txt file.

### Breadcrumb Navigation

A breadcrumb or breadcrumb trail is a graphical control element used as a navigational aid in user interfaces and on web pages. It allows users to keep track and maintain awareness of their locations within programs, documents, or websites. Breadcrumb navigation should only be used if there are 3 levels or more.



### Visual Content

#### Alt attribute

The alt attribute is the HTML attribute used in HTML and XHTML documents to specify alternative text that is to be rendered when the element to which it is applied cannot be rendered.

File names

### Layout Shifts

If images are lacking an aspect ratio, the browser has no way of knowing how tall or wide an image is until it loads. This can cause content to shift as the image loads.

Add width="..." and height="..." attributes to specified images, matching their natural width and height, to provide an aspect ratio.

# Tools

## Google Optimize

[Google Optimize](https://optimize.google.com/optimize/home/), formerly called Google Website Optimizer, is a freemium web analytics and testing tool by Google. It allows running some experiments that are aimed to help online marketers and webmasters to increase visitor conversion rates and overall visitor satisfaction.

## Google Search Console

[Google Search Console](https://search.google.com/search-console/about) is a web service by Google which allows webmasters to check indexing status, search queries, crawling errors and optimize visibility of their websites.

## Google Shopping

[Google Shopping](https://shopping.google.com/) is a Google service which allows users to search for products on online shopping websites and compare prices between different vendors.

## Google Mobile Friendly

<https://search.google.com/test/mobile-friendly>

## AHREFS

[Ahrefs](https://ahrefs.com/) is mainly used to analyze a website's link profile, keyword rankings, and SEO health. You can also use [Ahrefs](https://ahrefs.com/) to conduct keyword research for Google, YouTube, and Amazon. And many people use [Ahrefs](https://ahrefs.com/) to find content that's performed well (in terms of social shares and/or links) on a given topic.

## Screaming Frog

The [Screaming Frog SEO Spider](https://www.screamingfrog.co.uk/seo-spider/) is a fast and advanced SEO site audit tool.

## Lighthouse

<https://chrome.google.com/webstore/detail/lighthouse/blipmdconlkpinefehnmjammfjpmpbjk?hl=en>

<https://pagespeed.web.dev/>

[www.webpagetest.org](http://www.webpagetest.org)