

Developing an SEO-Friendly Website

In this chapter, we will examine ways to assess the search engine friendliness of your website. A search engine–friendly website, at the most basic level, is one that allows for search engine access to site content—and having your site content accessible to search engines is the first step toward creating prominent visibility in search results. Once your site’s content is accessed by a search engine, it can then be considered for relevant positioning within search results pages.

As we discussed in the introduction to [Chapter 2](#), search engine crawlers are basically software programs, and like all software programs, they come with certain strengths and weaknesses. Publishers must adapt their websites to make the job of these software programs easier—in essence, leverage their strengths and make their weaknesses irrelevant. If you can do this, you will have taken a major step toward success with SEO.

Developing an SEO-friendly site architecture requires a significant amount of thought, planning, and communication due to the large number of factors that influence how a search engine sees your site and the myriad ways in which a website can be put together, as there are hundreds (if not thousands) of tools that web developers can use to build a website—many of which were not initially designed with SEO or search engine crawlers in mind.

Making Your Site Accessible to Search Engines

The first step in the SEO design process is to ensure that your site can be found and crawled by search engines. This is not as simple as it sounds, as there are many popular web design and implementation constructs that the crawlers may not understand.

Indexable Content

To rank well in the search engines, your site's content—that is, the material available to visitors of your site—should be in HTML text form. Images and Flash files, for example, while crawled by the search engines, are content types that are more difficult for search engines to analyze and therefore are not ideal for communicating to search engines the topical relevance of your pages.

Search engines have challenges with identifying the relevance of images because there are minimum text-input fields for image files in GIF, JPEG, or PNG format (namely the filename, title, and `alt` attribute). While we do strongly recommend accurate labeling of images in these fields, images alone are usually not enough to earn a web page top rankings for relevant queries. While image identification technology continues to advance, processing power limitations will likely keep the search engines from broadly applying this type of analysis to web search in the near future.

Google enables users to perform a search using an image, as opposed to text, as the search query (though users can input text to augment the query). By uploading an image, dragging and dropping an image from the desktop, entering an image URL, or right-clicking on an image within a browser (Firefox and Chrome with installed extensions), users can often find other locations of that image on the Web for reference and research, as well as images that appear similar in tone and composition. While this does not immediately change the landscape of SEO for images, it does give us an indication of how Google is potentially augmenting its current relevance indicators for image content.

With Flash, while specific `.swf` files (the most common file extension for Flash) can be crawled and indexed—and are often found when a user conducts a `.swf` file search for specific words or phrases included in their filename—it is rare for a generic query to return a Flash file or a website generated entirely in Flash as a highly relevant result, due to the lack of “readable” content. This is not to say that websites developed using Flash are inherently irrelevant, or that it is impossible to successfully optimize a website that uses Flash; however, in our experience the preference must still be given to HTML-based files.

Spiderable Link Structures

As we outlined in [Chapter 2](#), search engines use links on web pages to help them discover other web pages and websites. For this reason, we strongly recommend taking the time to build an internal linking structure that spiders can crawl easily. Many sites make the critical mistake of hiding or obfuscating their navigation in ways that limit spider accessibility, thus impacting their ability to get pages listed in the search engines' indexes. Consider [Figure 6-1](#), which shows how this problem can occur.

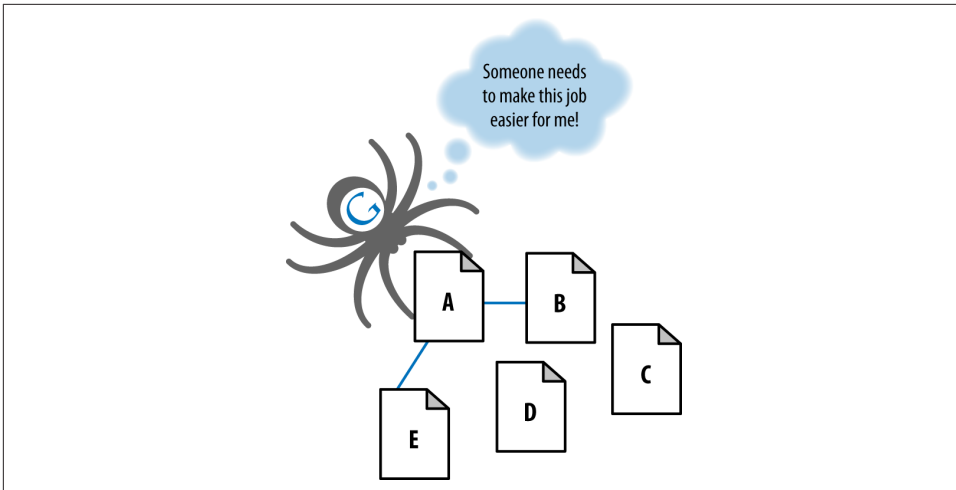


Figure 6-1. *Providing search engines with crawlable link structures*

In **Figure 6-1**, Google’s spider has reached Page A and sees links to pages B and E. However, even though pages C and D might be important pages on the site, the spider has no way to reach them (or even to know they exist) because no direct, crawlable links point to those pages. As far as Google is concerned, they might as well not exist. Great content, good keyword targeting, and smart marketing won’t make any difference at all if the spiders can’t reach those pages in the first place.

To refresh your memory of the discussion in **Chapter 2**, here are some common reasons why pages may not be reachable:

Links in submission-required forms

Search spiders will rarely, if ever, attempt to “submit” forms, and thus, any content or links that are accessible only via a form are invisible to the engines. This even applies to simple forms such as user logins, search boxes, or some types of pull-down lists.

Links in hard-to-parse JavaScript

If you use JavaScript for links, you may find that search engines either do not crawl or give very little weight to the embedded links. In June 2014, Google **announced enhanced crawling of JavaScript and CSS**. Google can now render some JavaScript and follow some JavaScript links. Due to this change, Google recommends against blocking it from crawling your JavaScript and CSS files. For a preview of how your site might render according to Google, go to Search Console -> Crawl -> Fetch as Google, input the URL you would like to preview, and select “Fetch and Render.”

Links in Java or other plug-ins

Traditionally, links embedded inside Java and plug-ins have been invisible to the engines.

Links in Flash

In theory, search engines can detect links within Flash, but don't rely too heavily on this capability.

Links in PowerPoint and PDF files

Search engines sometimes report links seen in PowerPoint files or PDFs. These links are believed to be counted the same as links embedded in HTML documents.

Links pointing to pages blocked by the meta robots tag, rel="nofollow", or robots.txt

The *robots.txt* file provides a very simple means for preventing web spiders from crawling pages on your site. Using the `nofollow` attribute on a link, or placing the meta `robots nofollow` tag with the `content="nofollow"` attribute on the page containing the link, instructs the search engine to not pass link authority via the link (a concept we will discuss further in [“Content Delivery and Search Spider Control” on page 334](#)). The effectiveness of the `nofollow` attribute on links has greatly diminished to the point of irrelevance as a result of overmanipulation by aggressive SEO practitioners. For more on this, see the blog post [“PageRank Sculpting”](#), by Google's Matt Cutts.

Links on pages with many hundreds or thousands of links

Historically, Google had suggested a maximum of 100 links per page before it may stop spidering additional links from that page, but this recommendation has softened over time. Think of it more as a strategic guideline for passing PageRank. If a page has 200 links on it, then none of the links get very much PageRank. Managing how you pass PageRank by limiting the number of links is usually a good idea. Tools such as [Screaming Frog](#) can run reports on the number of outgoing links you have per page.

Links in frames or iframes

Technically, links in both frames and iframes can be crawled, but both present structural issues for the engines in terms of organization and following. Unless you're an advanced user with a good technical understanding of how search engines index and follow links in frames, it is best to stay away from them as a place to offer links for crawling purposes. We will discuss frames and iframes in more detail in [“Creating an Optimal Information Architecture” on page 267](#).

XML Sitemaps

Google, Yahoo!, and Bing (formerly MSN Search, and then Live Search) all support a protocol known as XML Sitemaps. Google first announced it in 2005, and then Yahoo! and MSN Search agreed to support the protocol in 2006. Using the Sitemaps protocol,

you can supply search engines with a list of all the URLs you would like them to crawl and index.

Adding a URL to a sitemap file does not guarantee that a URL will be crawled or indexed. However, it can result in the search engine discovering and indexing pages that it otherwise would not.

This program is a complement to, not a replacement for, the search engines' normal, link-based crawl. The benefits of sitemaps include the following:

- For the pages the search engines already know about through their regular spidering, they use the metadata you supply, such as the last date the content was modified (*lastmod date*) and the frequency at which the page is changed (*changefreq*), to improve how they crawl your site.
- For the pages they don't know about, they use the additional URLs you supply to increase their crawl coverage.
- For URLs that may have duplicates, the engines can use the XML Sitemaps data to help choose a canonical version.
- Verification/registration of XML sitemaps may indicate positive trust/authority signals.
- The crawling/inclusion benefits of sitemaps may have second-order positive effects, such as improved rankings or greater internal link popularity.
- Having a sitemap registered with Google Search Console can give you extra analytical insight into whether your site is suffering from indexation, crawling, or duplicate content issues.

Matt Cutts, the former head of Google's webspam team, has explained XML sitemaps in the following way:

Imagine if you have pages A, B, and C on your site. We find pages A and B through our normal web crawl of your links. Then you build a Sitemap and list the pages B and C. Now there's a chance (but not a promise) that we'll crawl page C. We won't drop page A just because you didn't list it in your Sitemap. And just because you listed a page that we didn't know about doesn't guarantee that we'll crawl it. But if for some reason we didn't see any links to C, or maybe we knew about page C but the URL was rejected for having too many parameters or some other reason, now there's a chance that we'll crawl that page C.⁴

⁴ See <http://www.stephanspencer.com/whats-wrong-with-google-sitemaps/>.

Sitemaps use a simple XML format that you can learn about at <http://www.sitemaps.org/>. XML sitemaps are a useful and in some cases essential tool for your website. In particular, if you have reason to believe that the site is not fully indexed, an XML sitemap can help you increase the number of indexed pages. As sites grow in size, the value of XML sitemap files tends to increase dramatically, as additional traffic flows to the newly included URLs.

Laying out an XML sitemap

The first step in the process of creating an XML sitemap is to create an XML sitemap file in a suitable format. Because creating an XML sitemap requires a certain level of technical know-how, it would be wise to involve your development team in the XML sitemap generator process from the beginning. Figure 6-2 shows an example of some code from a sitemap.

```
<?xml version="1.0" encoding="UTF-8"?>

<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">

  <url>

    <loc>http://www.example.com</loc>

    <lastmod>2005-01-01</lastmod>

    <changefreq>monthly</changefreq>

    <priority>0.8</priority>

  </url>

</urlset>
```

Figure 6-2. Sample XML sitemap from Google.com

To create your XML sitemap, you can use the following:

An XML sitemap generator

This is a simple script that you can configure to automatically create sitemaps, and sometimes submit them as well. Sitemap generators can create these sitemaps from a URL list, access logs, or a directory path hosting static files corresponding to URLs. Here are some examples of XML sitemap generators:

- [SourceForge.net's Google-sitemap_gen](#)
- [XML-Sitemaps.com Sitemap Generator](#)
- [Sitemaps Pal](#)

- **GSite Crawler**

Simple text

You can provide Google with a simple text file that contains one URL per line. However, Google recommends that once you have a text sitemap file for your site, you use the sitemap generator to create a sitemap from this text file using the Sitemaps protocol.

Syndication feed

Google accepts Really Simple Syndication (RSS) 2.0 and Atom 1.0 feeds. Note that the feed may provide information on recent URLs only.

Deciding what to include in a sitemap file

When you create a sitemap file, you need to take care in situations where your site has multiple URLs that refer to one piece of content: include *only* the preferred (canonical) version of the URL, as the search engines may assume that the URL specified in a sitemap file is the preferred form of the URL for the content. You can use the sitemap file to indicate to the search engines which URL points to the preferred version of a given page.

In addition, be careful about what *not* to include. For example, do not include multiple URLs that point to identical content, and leave out pages that are simply pagination pages (or alternate sort orders for the same content) and/or any low-value pages on your site. Last but not least, make sure that none of the URLs listed in the sitemap file include any tracking parameters.

Mobile sitemaps. Mobile sitemaps should be used for content targeted at mobile devices. Mobile information is kept in a separate sitemap file that should not contain any information on nonmobile URLs. Google supports nonmobile markup, XHTML mobile profile, WML (WAP 1.2) and cHTML. Details on the mobile sitemap format can be found here: <https://support.google.com/webmasters/answer/34648>.

Video sitemaps. Including information on your videos in your sitemap file will increase their chances of being discovered by search engines. Google supports the following video formats: *.mpg, .mpeg, .mp4, .m4v, .mov, .wmv, .asf, .avi, .ra, .ram, .rm, .flv*, and *.swf*. You can see the specification on how to implement video sitemap entries here: <https://support.google.com/webmasters/answer/80472>.

Image sitemaps. You can increase visibility for your images by listing them in your sitemap file. For each URL you list in your sitemap file, you can also list the images that appear on those pages. You can list up to 1,000 images per page. Specialized image tags are associated with the URL. The details of the format of these tags are on this page: <https://support.google.com/webmasters/answer/178636>.

Listing images in the sitemap does increase the chances of those images being indexed. If you list some images and not others, it may be interpreted as a signal that the unlisted images are less important.

Uploading your sitemap file

When your sitemap file is complete, upload it to your site in the highest-level directory you want search engines to crawl (generally, the root directory), such as *www.your-site.com/sitemap.xml*. You can include more than one subdomain in your sitemap provided that you verify the sitemap for each subdomain in Google Search Console, though it's frequently easier to understand what's happening with indexation if each subdomain has its own sitemap and its own profile in Google Search Console.

Managing and updating XML sitemaps

Once your XML sitemap has been accepted and your site has been crawled, monitor the results and update your sitemap if there are issues. With Google, you can return to your Google Search Console account to view the statistics and diagnostics related to your XML sitemaps. Just click the site you want to monitor. You'll also find some FAQs from Google on common issues such as slow crawling and low indexation.

Update your XML sitemap with Google and Bing when you add URLs to your site. You'll also want to keep your sitemap file up to date when you add a large volume of pages or a group of pages that are strategic.

There is no need to update the XML sitemap when you're simply updating content on existing URLs. It is not strictly necessary to update when pages are deleted, as the search engines will simply not be able to crawl them, but do update before you have too many broken pages in your feed. Also update your sitemap file whenever you add any new content, and remove any deleted pages at that time. Google and Bing will periodically redownload the sitemap, so you don't need to resubmit your sitemap to Google or Bing unless your sitemap location has changed.

Enable Google and Bing to autodiscover your XML sitemap locations by using the `Sitemap:` directive in your site's *robots.txt* file.

If you are adding or deleting large numbers of new pages to your site on a regular basis, you may want to use a utility, or have your developers build the ability, for your XML sitemap to regenerate with all of your current URLs on a regular basis. Many sites regenerate their XML sitemap daily via automated scripts.

Google and the other major search engines discover and index websites by crawling links. Google XML sitemaps are a way to feed the URLs that you want crawled on your site to Google for more complete crawling and indexation, which results in improved long-tail searchability. By creating and updating this XML file, you help to

ensure that Google recognizes your entire site, and this recognition will help people find your site. It also helps all of the search engines understand which version of your URLs (if you have more than one URL pointing to the same content) is the canonical version.

Creating an Optimal Information Architecture

Making your site friendly to search engine crawlers also requires that you put some thought into your site's *information architecture* (IA). A well-designed site architecture can bring many benefits for both users and search engines.

The Importance of a Logical, Category-Based Flow

Search engines face myriad technical challenges in understanding your site, as crawlers are not able to perceive web pages in the way that humans do, creating significant limitations for both accessibility and indexing. A logical and properly constructed website architecture can help overcome these issues and bring great benefits in search traffic and usability.

At the core of website information architecture are two critical principles: usability (making a site easy to use) and information architecture (crafting a logical, hierarchical structure for content).

One of the very early information architecture proponents, Richard Saul Wurman, developed the following definition for *information architect*:⁵

1) the individual who organizes the patterns inherent in data, making the complex clear. 2) a person who creates the structure or map of information which allows others to find their personal paths to knowledge. 3) the emerging 21st century professional occupation addressing the needs of the age focused upon clarity, human understanding, and the science of the organization of information.

Usability and search friendliness

Search engines are trying to reproduce the human process of sorting relevant web pages by quality. If a real human were to do this job, usability and user experience would surely play a large role in determining the rankings. Given that search engines are machines and don't have the ability to segregate by this metric quite so easily, they are forced to employ a variety of alternative, secondary metrics to assist in the process. The most well known and well publicized among these is a measurement of the

⁵ From Wurman's *Information Architects*.

inbound links to a website (see [Figure 6-3](#)), and a well-organized site is more likely to receive links.

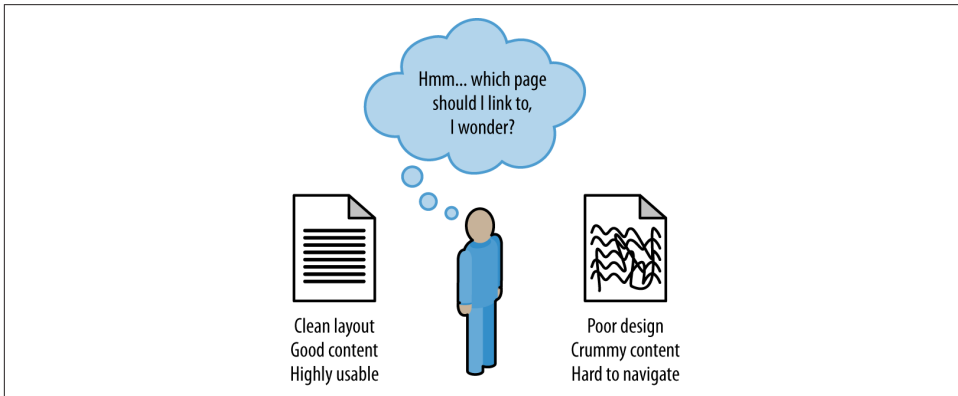


Figure 6-3. *Make your site attractive to link to*

Since Google launched in the late 1990s, search engines have strived to analyze every facet of the link structure on the Web and have extraordinary abilities to infer trust, quality, reliability, and authority via links. If you push back the curtain and examine why links between websites exist and how they come to be, you can see that a human being (or several humans, if the organization suffers from bureaucracy) is almost always responsible for the creation of links.

The engines hypothesize that high-quality links will point to high-quality content, and that great content and positive user experiences will be rewarded with more links than poor user experiences. In practice, the theory holds up well. Modern search engines have done a very good job of placing good-quality, usable sites in top positions for queries.

An analogy

Look at how a standard filing cabinet is organized. You have the individual cabinet, drawers in the cabinet, folders within the drawers, files within the folders, and documents within the files (see [Figure 6-4](#)).

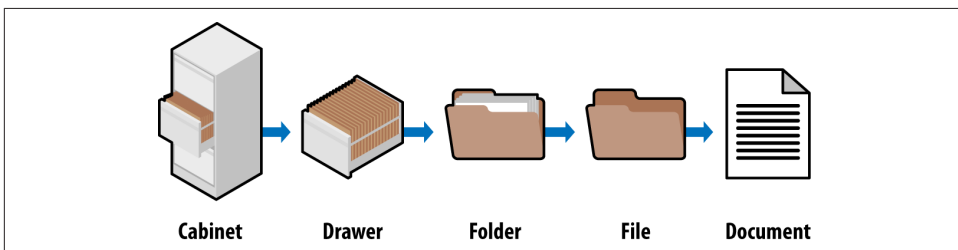


Figure 6-4. *Similarities between filing cabinets and web pages*

There is only one copy of any individual document, and it is located in a particular spot. There is a very clear navigation path to get to it.

If you want to find the January 2015 invoice for a client (Amalgamated Glove & Spat), you would go to the cabinet, open the drawer marked Client Accounts, find the Amalgamated Glove & Spat folder, look for the Invoices file, and then flip through the documents until you come to the January 2015 invoice (again, there is only one copy of this; you won't find it anywhere else).

Figure 6-5 shows what it looks like when you apply this logic to the popular website, Craigslist.

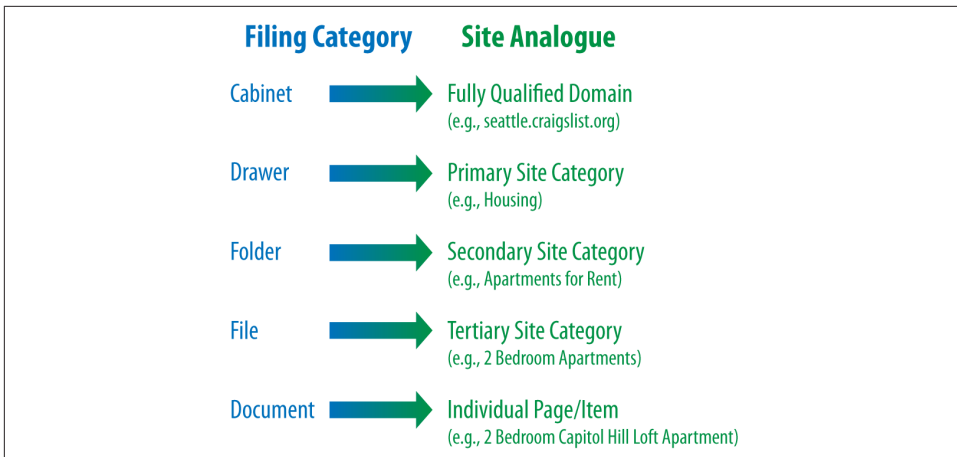


Figure 6-5. Filing cabinet analogy applied to Craigslist

If you're seeking an apartment in Los Angeles, you'd navigate to <http://losangeles.craigslist.org/>, choose apts/housing, narrow that down to two bedrooms, and pick the two-bedroom loft from the list of available postings. Craigslist's simple, logical information architecture makes it easy for you to reach the desired post in four clicks, without having to think too hard at any step about where to go. This principle applies perfectly to the process of SEO, where good information architecture dictates:

- As few clicks as possible to any given page
- One hundred or fewer links per page (so as not to overwhelm either crawlers or visitors)
- A logical, semantic flow of links from home page to categories to detail pages

Here is a brief look at how this basic filing cabinet approach can work for some more complex information architecture issues.

Subdomains. You should think of subdomains as completely separate filing cabinets within one big room. They may share similar architecture, but they shouldn't share the same content; and more importantly, if someone points you to one cabinet to find something, he is indicating that *that* cabinet is the authority, not the other cabinets in the room. Why is this important? It will help you remember that links (i.e., votes or references) to subdomains may not pass all, or any, of their authority to other subdomains within the room (e.g., **.craigslist.org*, wherein *** is a variable subdomain name).

Those cabinets, their contents, and their authority are isolated from one another and may not be considered to be associated with one another. This is why, in most cases, it is best to have one large, well-organized filing cabinet instead of several that may prevent users and bots from finding what they want.

Redirects. If you have an organized administrative assistant, he probably uses 301 redirects (these are discussed more in the section “**Redirects**” on page 270) inside his literal, metal filing cabinet. If he finds himself looking for something in the wrong place, he might put a sticky note there reminding him of the correct location the next time he needs to look for that item. Anytime he looked for something in those cabinets, he could always find it because if he navigated improperly, he would inevitably find a note pointing him in the right direction.

Redirect irrelevant, outdated, or misplaced content to the proper spot in your filing cabinet, and both your users and the engines will know what qualities and keywords you think it should be associated with.

URLs. It would be tremendously difficult to find something in a filing cabinet if every time you went to look for it, it had a different name, or if that name resembled *jklhj25br3g452ikbr52k*—a not-so-uncommon type of character string found in dynamic website URLs. Static, keyword-targeted URLs are much better for users and bots alike. They can always be found in the same place, and they give semantic clues as to the nature of the content.

These specifics aside, thinking of your site information architecture as a virtual filing cabinet is a good way to make sense of best practices. It'll help keep you focused on a simple, easily navigated, easily crawled, well-organized structure. It is also a great way to explain an often-complicated set of concepts to clients and coworkers.

Because search engines rely on links to crawl the Web and organize its content, the architecture of your site is critical to optimization. Many websites grow organically and, like poorly planned filing systems, become complex, illogical structures that force people (and spiders) to struggle to find what they want.

Site Architecture Design Principles

In planning your website, remember that nearly every user will initially be confused about where to go, what to do, and how to find what she wants. An architecture that recognizes this difficulty and leverages familiar standards of usability with an intuitive link structure will have the best chance of making a visit to the site a positive experience. A well-organized site architecture helps solve these problems, and provides semantic and usability benefits to both users and search engines.

As shown in [Figure 6-6](#), a recipes website can use intelligent architecture to fulfill visitors' expectations about content and create a positive browsing experience. This structure not only helps humans navigate a site more easily, but also helps the search engines to see that your content fits into logical concept groups. You can use this approach to help you rank for applications of your product in addition to attributes of your product.

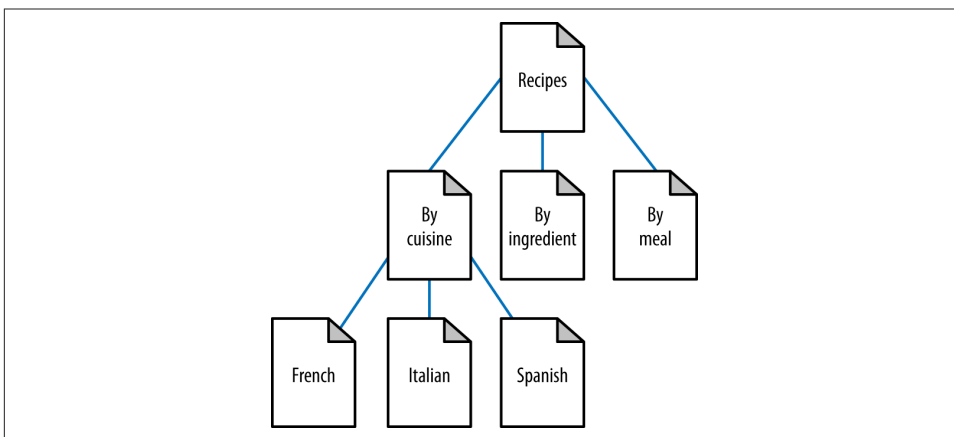


Figure 6-6. *Structured site architecture*

Although site architecture accounts for a small part of the algorithms, search engines do make use of relationships between subjects and give value to content that has been organized sensibly. For example, if in [Figure 6-6](#) you were to randomly jumble the subpages into incorrect categories, your rankings could suffer. Search engines, through their massive experience with crawling the Web, recognize patterns in subject architecture and reward sites that embrace an intuitive content flow.

Site architecture protocol

Although site architecture—the creation of structure and flow in a website's topical hierarchy—is typically the territory of information architects (or is created without assistance from a company's internal content team), its impact on search engine

rankings, particularly in the long run, is substantial. It is, therefore, a wise endeavor to follow basic guidelines of search friendliness.

The process itself should not be overly arduous, if you follow this simple protocol:

1. List all of the requisite content pages (blog posts, articles, product detail pages, etc.).
2. Create top-level navigation that can comfortably hold all of the unique types of detailed content for the site.
3. Reverse the traditional top-down process by starting with the detailed content and working your way up to an organizational structure capable of holding each page.
4. Once you understand the bottom, fill in the middle. Build out a structure for sub-navigation to sensibly connect top-level pages with detailed content. In small sites, there may be no need for this level, whereas in larger sites, two or even three levels of subnavigation may be required.
5. Include secondary pages such as copyright, contact information, and other nonessentials.
6. Build a visual hierarchy that shows (to at least the last level of subnavigation) each page on the site.

Figure 6-7 shows an example of a well-structured site architecture.

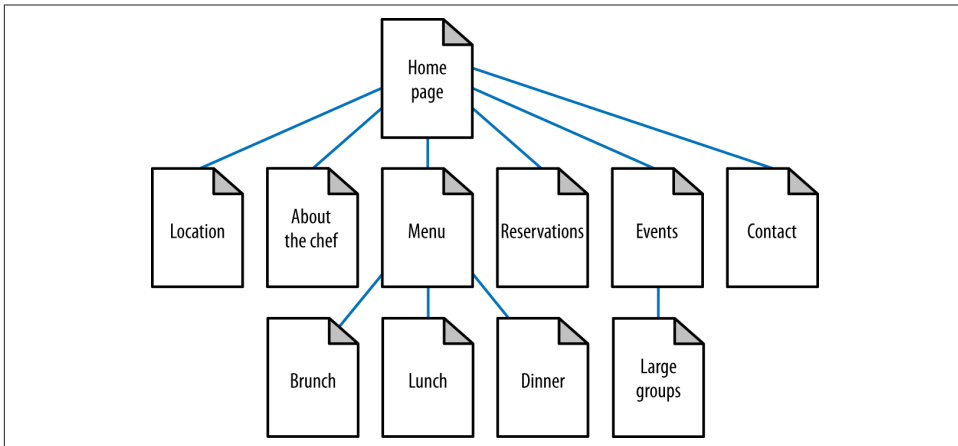


Figure 6-7. Second example of structured site architecture

Category structuring

As search engines crawl the Web, they collect an incredible amount of data (millions of gigabytes) on the structure of language, subject matter, and relationships between content. Though not technically an attempt at artificial intelligence, the engines have

built a repository capable of making sophisticated determinations based on common patterns. As shown in **Figure 6-8**, search engine spiders can learn semantic relationships as they crawl thousands of pages that cover a related topic (in this case, dogs).

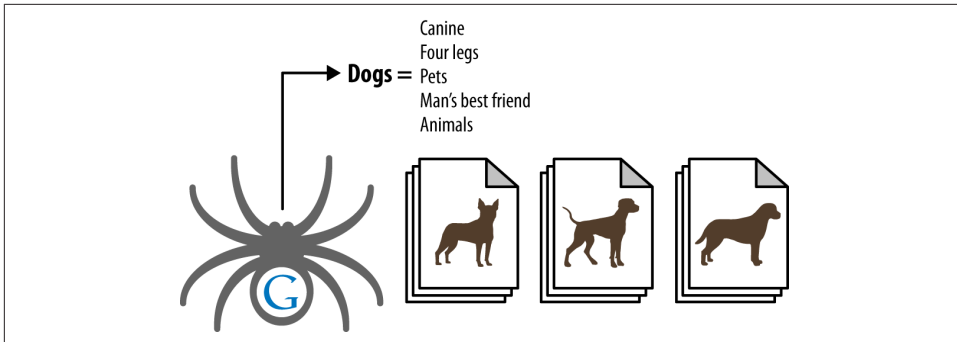


Figure 6-8. Spiders learning semantic relationships

Although content need not always be structured along the most predictable patterns, particularly when a different method of sorting can provide value or interest to a visitor, organizing subjects logically assists both humans (who will find your site easier to use) and engines (which will award you with greater rankings based on increased subject relevance).

Topical relevance. Naturally, this pattern of relevance-based scoring extends from single relationships between documents to the entire category structure of a website. Site creators can take advantage of this best by building hierarchies that flow from broad, encompassing subject matter down to more detailed, specific content. Obviously, in any categorization system, there is a natural level of subjectivity; think first of your visitors, and use these guidelines to ensure that your creativity doesn't overwhelm the project.

Taxonomy and ontology

In designing a website, you should also consider the taxonomy and ontology of the website. Taxonomy is essentially a two-dimensional hierarchical model of the architecture of the site. You can think of ontology as mapping the way the human mind thinks about a topic area. It can be much more complex than taxonomy, because a larger number of relationship types are often involved.

One effective technique for coming up with an ontology is called *card sorting*. This is a user-testing technique whereby users are asked to group items together so that you can organize your site as intuitively as possible. Card sorting can help identify not only the most logical paths through your site, but also ambiguous or cryptic terminology that should be reworded.

With card sorting, you write all the major concepts onto a set of cards that are large enough for participants to read, manipulate, and organize. Your test group assembles the cards in the order they believe provides the most logical flow, as well as into groups that seem to fit together.

By itself, building an ontology is not part of SEO, but when you do it properly it will impact your site architecture, and therefore it interacts with SEO. Coming up with the right site architecture should involve both disciplines.

Flat Versus Deep Architecture

One very strict rule for search friendliness is the creation of flat site architecture. Flat sites require a minimal number of clicks to access any given page, whereas deep sites create long paths of links required to access detailed content. For nearly every site with fewer than 10,000 pages, all content should be accessible through a maximum of four clicks from the home page and/or sitemap page. That said, flatness should not be forced if it does not make sense for other reasons. At 100 links per page, even sites with millions of pages can have every page accessible in five to six clicks if proper link and navigation structures are employed. If a site is not built to be flat, it can take too many clicks for a user or a search engine to reach the desired content, as shown in [Figure 6-9](#). In contrast, a flat site (see [Figure 6-10](#)) allows users and search engines to reach most content in just a few clicks.

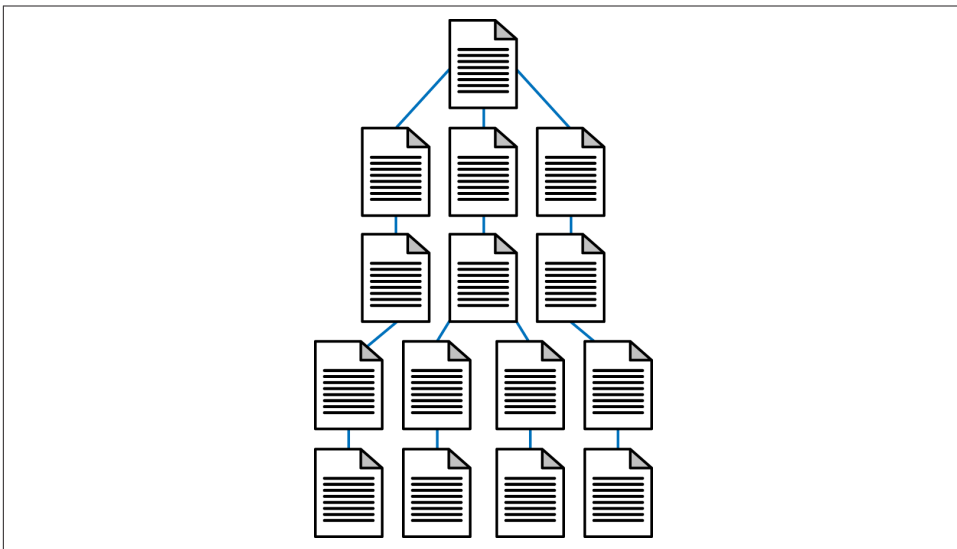


Figure 6-9. *Deep site architecture*

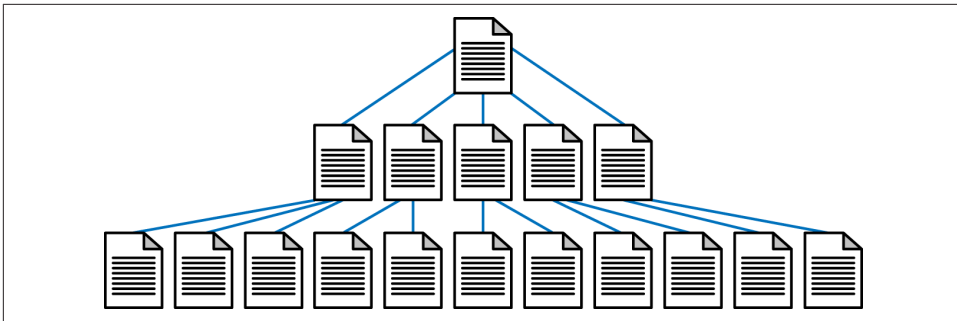


Figure 6-10. *Flat site architecture*

Flat sites aren't just easier for search engines to crawl; they are also simpler for users, as they limit the number of page visits the user requires to reach his destination. This reduces the abandonment rate and encourages repeat visits.

When creating flat sites, be careful to not overload pages with links either. Pages that have 200 links on them are not passing much PageRank to any of those pages. While flat site architectures are desirable, you should not force an architecture to be overly flat if it is not otherwise logical to do so.

The issue of the number of links per page relates directly to another rule for site architects: avoid excessive pagination wherever possible. *Pagination* (see [Figure 6-11](#)), the practice of creating a list of elements on pages separated solely by numbers (e.g., some ecommerce sites use pagination for product catalogs that have more products than they wish to show on a single page), is problematic for many reasons.

First, pagination provides virtually no new topical relevance, as the pages are each largely about the same topic. Second, content that moves into different pagination can potentially create duplicate content problems or be seen as poor-quality or "thin" content. Last, pagination can create spider traps and hundreds or thousands of extraneous, low-quality pages that can be detrimental to search visibility.

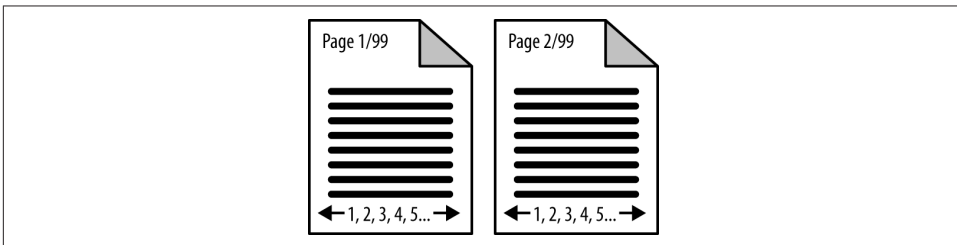


Figure 6-11. *Pagination structures*

So, make sure you implement flat structures and stay within sensible guidelines for the number of links per page, while retaining a contextually rich link structure. This is

not always as easy as it sounds, and accomplishing it may require quite a bit of thought and planning to build a contextually rich structure on some sites. Consider a site with 10,000 different men's running shoes. Defining an optimal structure for that site could be a very large effort, but that effort will pay serious dividends in return.

Solutions to pagination problems vary based on the content of the website. Here are two example scenarios and their solutions:

Use rel="next" and rel="prev"

Google supports link elements called `rel="next"` and `rel="prev"`. The benefit of using these link elements is that it lets Google know when it has encountered a sequence of paginated pages. Once Google recognizes these tags, links to any of the pages will be treated as links to the series of pages as a whole. In addition, Google will show in the index the most relevant page in the series (most of the time this will be the first page, but not always).

Bing **announced support for `rel="next"` and `rel="prev"` in 2012.**

These tags can be used to inform Google about pagination structures, and they can be used whether or not you create a `view-all` page. The concept is simple. The following example outlines how to use the tags for content that is paginated into 12 pages:

- In the `<head>` section of the first page of your paginated content, implement a `rel="next"` tag pointing to the second page of the content. The tag should look something like this:

```
<link rel="next"
      href="http://www.yoursite.com/products?prod=qwert&p=2" />
```

- In the `<head>` section of the last page of your paginated content, implement a `rel="prev"` link element pointing to the second-to-last page of the content. The tag should look something like this:

```
<link rel="prev"
      href="http://www.yoursite.com/products?prod=qwert&p=11" />
```

In the `<head>` section of pages 2 through 11, implement `rel="next"` and `rel="prev"` tags pointing to the following and preceding pages, respectively. The following example shows what it should look like on page 6 of the content:

```
<link rel="prev"
      href="http://www.yoursite.com/products?prod=qwert&p=5" />
<link rel="next"
      href="http://www.yoursite.com/products?prod=qwert&p=7" />
```

Create a view-all page and use canonical tags

You may have lengthy articles that you choose to break into multiple pages. However, this results in links to the pages whose anchor text is something like "1", "2", and so forth. The titles of the various pages may not vary in any significant way, so they tend to compete with each other for search traffic. Finally, if someone links to the article but does not link to the first page, the link authority from that link will largely be wasted.

One way to handle this problem is to retain the paginated version of the article, but also create a single-page version of the article. This is referred to as a `view-all` page. Then use the `rel="canonical"` link element (which is discussed in more detail in the section "[Content Delivery and Search Spider Control](#)" on page 334) to point from the paginated pages to the `view-all` page. This will concentrate all of the link authority and search engine attention on a single page. You should also include a link to the `view-all` page from each of the individual paginated pages as well. However, if the `view-all` page loads too slowly because of the page size, it may not be the best option for you.

Note that if you implement a `view-all` page and do not implement any of these tags, Google will attempt to discover the page and show it instead of the paginated versions in its search results. However, we recommend that you make use of one of the aforementioned two solutions, as Google cannot guarantee that it will discover your `view-all` pages, and it is best to provide it with as many clues as possible.

Search-Friendly Site Navigation

Website navigation is something that web designers have been putting considerable thought and effort into since websites came into existence. Even before search engines were significant, navigation played an important role in helping users find what they wanted. It plays an important role in helping search engines understand your site as well.

Basics of search engine friendliness

The search engine spiders need to be able to read and interpret your website's code to properly spider and index the content on your web pages. Do not confuse this with the rules of organizations such as the W3C, which issues guidelines on HTML construction. Although following the W3C guidelines can be a good idea, the great majority of sites do not follow them, so search engines generally overlook violations of these rules as long as their spiders can parse the code.

Unfortunately, web page navigation and content can be rendered in many ways that function well for humans, but are invisible or at least challenging for search engine spiders.

For example, there are numerous ways to incorporate content and navigation on the pages of a website. For the most part, all of these are designed for humans. Basic HTML text and HTML links such as those shown in [Figure 6-12](#) work equally well for humans and search engine crawlers.

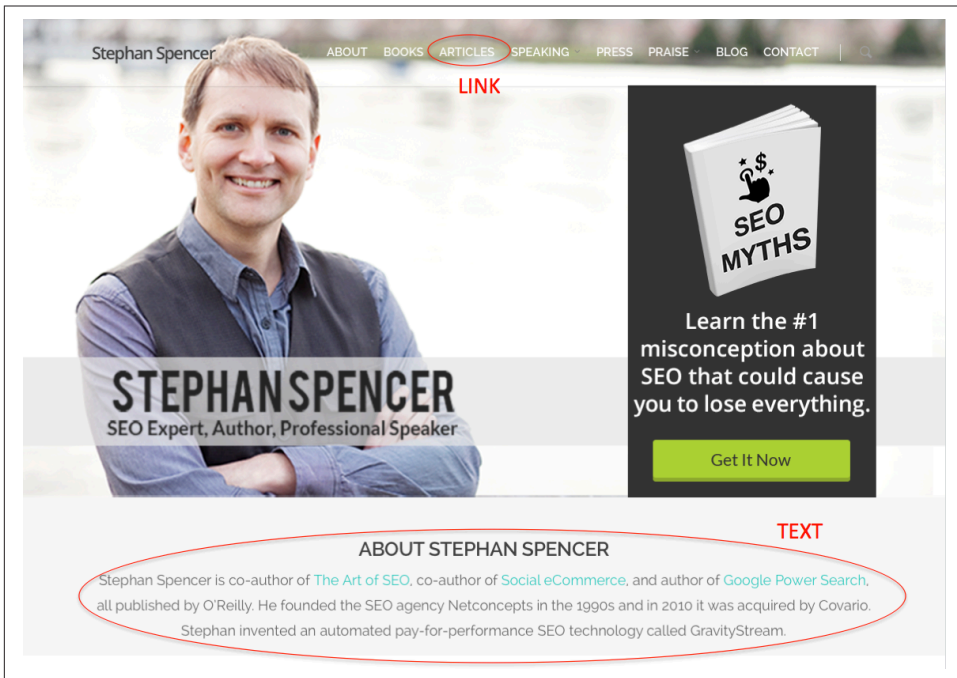


Figure 6-12. Example page with simple text and text link

The text and the link that are indicated on the page shown in [Figure 6-12](#) are in simple HTML format.

Site elements that are problematic for spiders

However, many other types of content may appear on a web page and work well for humans but not so well for search engines. Here are some of the most common ones.

Search and web forms

Many sites incorporate search functionality. These “site search” elements are specialized search engines that index and provide access to one site’s content.

This is a popular method of helping users rapidly find their way around complex sites. For example, the [Pew Internet website](#) provides Site Search in the upper-right corner; this is a great tool for users, but search engines will be stymied by it. Search engines operate by crawling the Web’s link structure—they don’t in most circumstances submit

forms or attempt random queries into search fields, and thus, any URLs or content solely accessible via a form will remain invisible to Google and Bing. In the case of Site Search tools, this is OK, as search engines do not want to index this type of content (they don't like to serve search results within their search results).

Forms are a popular way to provide interactivity, and one of the simplest applications is the "Contact us" form many websites have.

Unfortunately, crawlers will not fill out or submit such forms; thus, any content restricted to those who employ them is inaccessible to the engines. In the case of a "Contact us" form, this is likely to have little impact, but other types of forms can lead to bigger problems.

Websites that have content behind paywall and/or login barriers will need to either provide text links to the content behind the barrier (which defeats the purpose of the login) or implement First Click Free (discussed in "[Content Delivery and Search Spider Control](#)" on page 334).

Java, images, audio, and video. Flash files, Java embeds, audio, and video (in any format) present content that is largely uncrawlable by the major engines. With some notable exceptions that we will discuss later, search engines can read text only when it is presented in HTML format. Embedding important keywords or entire paragraphs in an image or a Java console renders them invisible to the spiders. Likewise, search engines cannot easily read words spoken in an audio file or video. However, Google has begun to leverage tools such as Google Voice Search in order to "crawl" audio content and extract meaning (this was first confirmed in the book *In the Plex* by Steven Levy [Simon and Schuster]). Baidu already has an MP3 search function, and the Shazam and Jsaikoz applications have the ability to identify song hashes.

alt attributes, originally created as metadata for markup and an accessibility tag for vision-impaired users, are a good way to present at least some text content to the engines when you are displaying images or embedded, nontext content. Note that the alt attribute is not a strong signal, and using it on an image link is no substitute for implementing a simple text link with appropriately descriptive anchor text. A good alternative is to employ captions and text descriptions in the HTML content wherever possible.

In the past few years, a number of companies offering transcription services have cropped up, providing automated text creation for the words spoken in audio or video. Providing these transcripts on rich media pages makes your content accessible to the search engines and findable by keyword-searching visitors. You can also use software such as Dragon Naturally Speaking and dictate your "transcript" to your computer.

AJAX and JavaScript. JavaScript enables many dynamic functions inside a website, most of which interfere very minimally with the operations of a search engine spider. The exception is when a page must use a JavaScript call to reach another page, or to pull content that the spiders can't see in the HTML. In some instances, this content is not visible to search engine spiders. However, Google will attempt to execute JavaScript to access this type of content.⁶ Google's capabilities for accessing JavaScript have been improving over time, and you can expect that trend to continue.⁷

One example of Google reading JavaScript is Facebook Comments. Facebook Comments is a system offered by Facebook that allows publishers to collect comments from users on their site. **Figure 6-13** shows **an example of the Facebook Comments on a page on the TechCrunch site.**

If you examine the source code for this particular post, you will not see any of the text strings for these comments in the HTML of the page. This is because the comments are actually stored on Facebook and dynamically retrieved by the web server when the page is rendered.

This is an example of the type of content that was not historically indexed by the search engines. When you use a JavaScript implementation like this, it is not clear what Google or Bing will be able to do with it. Facebook Comments is a broadly used system, and it makes sense for the search engines to learn how to attempt to read that content, but as of March 2012 this content was not indexed by Google.

However, since then, this has changed. As of June 2015, this content is being indexed by Google and associated with the site hosting the Facebook Comments. You can test this (and whether your own content is indexed) by doing a Google search on a unique string of words, surrounded by double quotes to ensure Google searches only for those exact words in that exact order. For example, searching Google for one of the comments in **Figure 6-13**, "*As an ethnic Chinese, learning Mandarin and struggling, I'm extremely impressed*", does return the URL <http://techcrunch.com/2014/10/23/zuckerberg-speaks-chinese-internet-soils-itself> as a result.

6 Webmaster Central Blog, "Updating Our Technical Webmaster Guidelines," October 27, 2014, http://bit.ly/webmaster_guidelines.

7 Adam Audette, "We Tested How Googlebot Crawls Javascript And Here's What We Learned," Search Engine Land, May 8, 2015, <http://searchengineland.com/tested-googlebot-crawls-javascript-heres-learned-220157>.



Figure 6-13. Facebook Comments on TechCrunch

While Google has recently indicated that it executes most JavaScript, it's still possible that it doesn't execute *all* JavaScript. So, if your intent is to create content that you want the search engines to see, it is still safest to implement that content in a form that is directly visible in the HTML of the web page.

AJAX might present problems, most notably in the delivery of content that search engines may not be able to spider. Because AJAX uses database calls to retrieve data without refreshing a page or changing URLs, the content contained behind these technologies may be completely hidden from the search engines (see Figure 6-14).

In fact, in early 2015, Google indicated that it might move away from attempting to crawl any AJAX pages at all.⁸ This was further confirmed in a June 2015 article by Eric Enge in which Google's Gary Illyes said: "If you have one URL only, and people have

⁸ Barry Schwartz, "Google May Discontinue Its AJAX Crawlable Guidelines," Search Engine Land, March 5, 2015, http://bit.ly/ajax_crawlable_guidelines.

to click on stuff to see different sort orders or filters for the exact same content under that URL, then typically we would only see the default content.”⁹

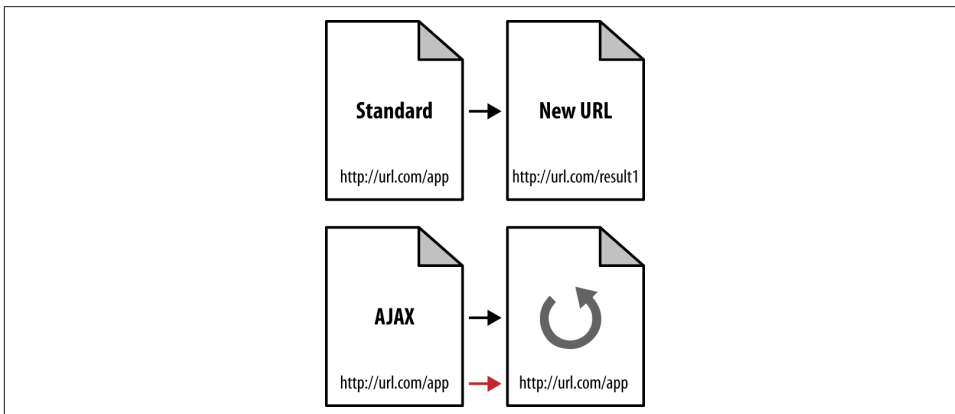


Figure 6-14. *The problem with AJAX*

As a result, if you’re using a traditional AJAX implementation, you may want to consider implementing an alternative spidering system for search engines to follow. AJAX applications are so user-friendly and appealing that forgoing them is simply impractical for many publishers. With these traditional implementations, building out a directory of links and pages that the engines can follow is a far better solution.

When you build these secondary structures of links and pages, make sure to provide users with access to them as well. Inside the AJAX application itself, give your visitors the option to “directly link to this page” and connect that URL with the URL you provide to search spiders through your link structures. AJAX apps not only suffer from content that can’t be crawled, but they also often don’t receive accurate links from users because the URL doesn’t change.

Some versions of AJAX use a # delimiter, which acts as a query string into the AJAX application. This allows you to link directly to different pages within the application. The #, which is used for HTML bookmarking, and everything beyond it are normally ignored by search engines.

This is largely because web browsers use only what’s after the # to jump to the anchor within the page, and that’s done locally within the browser. In other words, the browser doesn’t send the full URL, so the parameter information (i.e., any text after the #) is not passed back to the server.

⁹ Eric Enge, “Eliminate Duplicate Content in Faceted Navigation with Ajax/JSON/JQuery,” Moz Blogs, June 11, 2015, <https://moz.com/blog/using-ajax-json-jquery-to-implement-faceted-navigation>.

In 2009, Google outlined a method for making these AJAX pages visible to search engines. This was later followed up with recommendations made on the Google Developers site. You can find more information at http://bit.ly/ajax_crawling.

The solution proposed by Google involves making some slight modifications to the way your AJAX URLs are formatted so that its crawler can recognize when an AJAX URL can be treated like a static page (one that will always return the same content), in which case Googlebot will read the page and treat it like any other static page for indexing and ranking purposes, affording it the same opportunity to rank as a page coded in plain HTML.

Other types of single-page application frameworks, such as Angular.js, Backbone.js, or Ember.js, may have similar problems. You can read more about how to deal with these in “Angular.js: Making it SEO-friendly” on page 165.

Frames. Frames emerged in the mid-1990s as a popular way to make easy navigation systems. Unfortunately, both their usability (in 99% of cases) and their search friendliness (in 99.99% of cases) were exceptionally poor. Today, iframes and CSS can replace the need for frames, even when a site’s demands call for similar functionality.

For search engines, the biggest problem with frames and iframes is that they often hold the content from two or more URLs on a single page. For users, because search engines, which direct searchers to only a single URL, may get confused by frames and direct visitors to single pages (*orphan* pages) inside a site intended to show multiple URLs at once. Indeed, the search engines consider the content within an iframe as residing on a separate page from the one the iframe is being used on. Thus, pages with nothing but iframed content will look virtually blank to the search engines.

Additionally, because search engines rely on links, and frame pages will often change content for users without changing the URL, external links often point to the wrong URL unintentionally. As a consequence, links to the page containing the frame or iframe may not point to the content the linker wanted to point to. **Figure 6-15** illustrates how multiple pages are combined into a single URL with frames, which results in link distribution and spidering issues.

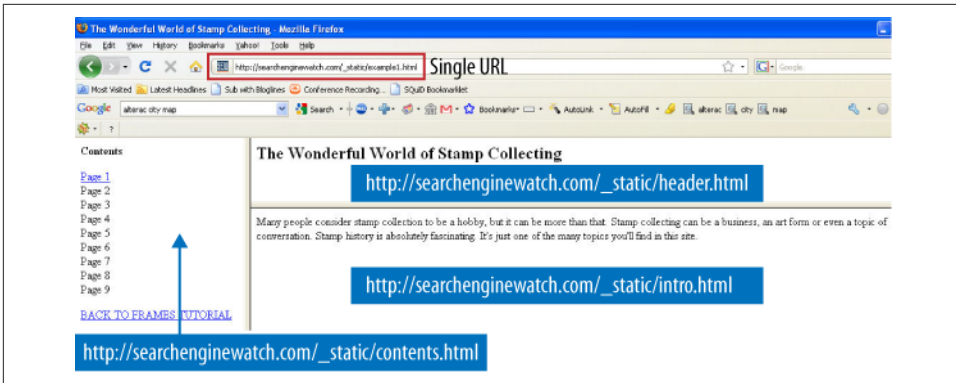


Figure 6-15. Sample page using frames

Search engine–friendly navigation guidelines

Although search engine spiders have become more advanced over the years, the basic premise and goals remain the same: spiders find web pages by following links and record the content of the pages they find in the search engine’s index (a giant repository of data about websites and pages).

In addition to avoiding the techniques we just discussed, there are some additional guidelines for developing search engine–friendly navigation:

Implement a text-link-based navigational structure

If you choose to create navigation in Flash, JavaScript, or other technologies that the search engine may not be able to parse, make sure to offer alternative text links in HTML for spiders to ensure that automated robots (and visitors who may not have the required browser plug-ins) can reach your pages.

Beware of “spider traps”

Even intelligently coded search engine spiders can get lost in infinite loops of links that pass between pages on a site. Intelligent architecture that avoids recursively looping 301 or 302 HTTP server codes (or other redirection protocols) should negate this issue, but sometimes online calendar links, infinite pagination that loops, or content being accessed or sorted in a multitude of ways (faceted navigation) can create tens of thousands of pages for search engine spiders when you intended to have only a few dozen true pages of content. You can read more about Google’s viewpoint on this at <http://googlewebmastercentral.blogspot.com/2008/08/to-infinity-and-beyond-no.html>.

Watch out for session IDs and cookies

As we just discussed, if you limit a user’s ability to view pages or redirect based on a cookie setting or session ID, search engines may be unable to crawl your content. The bots do not have cookies enabled, nor can they deal with session IDs

properly (each visit by the crawler gets a URL with a different session ID, and the search engine sees these URLs with session IDs as different URLs). Although restricting form submissions is fine (as search spiders can't submit forms anyway), limiting content access via cookies and session IDs is a bad idea.

Be mindful of server, hosting, and IP issues

Server issues rarely cause search engine ranking problems—but when they do, disastrous consequences can follow. The engines are acutely aware of common server problems, such as downtime or overloading, and will give you the benefit of the doubt (though this will mean your content cannot be spidered during periods of server dysfunction). On the flip side, sites hosted on content delivery networks (CDNs) may get crawled more heavily, and CDNs offer significant performance enhancements to a website.

The IP address of your host can be of concern in some instances. IPs once belonging to sites that have spammed the search engines may carry with them negative associations that can hinder spidering and ranking. While the engines aren't especially picky about shared hosting versus dedicated servers and dedicated IP addresses, or about server platforms, you can avoid many hassles by going these routes. At the very minimum, you should be cautious and find a host you trust, and inquire into the history and “cleanliness” of the IP address you may be assigned, as the search engines have become paranoid about the use of certain domains, hosts, IP addresses, and blocks of IPs. Experience tells them that many of these have strong correlations with spam, and thus, removing them from the index can have great benefits for users. As a site owner *not* engaging in these practices, you'll find it pays to investigate your web host prior to getting into trouble.

You can read more about server and hosting issues in [“Identifying Current Server Statistics Software and Gaining Access”](#) on page 177.

Root Domains, Subdomains, and Microsites

Among the common questions about structuring a website (or restructuring one) are whether to host content on a new domain, when to use subfolders, and when to employ microsites.

As search engines scour the Web, they identify four kinds of web structures on which to place metrics:

Individual pages/URLs

These are the most basic elements of the Web—filenames, much like those that have been found on computers for decades, which indicate unique documents. Search engines assign query-independent scores—most famously, Google's

PageRank—to URLs and judge them in their ranking algorithms. A typical URL might look something like *http://www.yourdomain.com/page*.

Subfolders

The folder structures that websites use can also inherit or be assigned metrics by search engines (though there's very little information to suggest that they are used one way or another). Luckily, they are an easy structure to understand. In the URL *http://www.yourdomain.com/blog/post17*, */blog/* is the subfolder and *post17* is the name of the file in that subfolder. Engines may identify common features of documents in a given subfolder and assign metrics to these (such as how frequently the content changes, how important these documents are in general, or how unique the content is that exists in these subfolders).

Subdomains/fully qualified domains (FQDs)/third-level domains

In the URL *http://blog.yourdomain.com/page*, three kinds of domain levels are present. The top-level domain (also called the *TLD* or *domain extension*) is *.com*, the second-level domain is *yourdomain*, and the third-level domain is *blog*. The third-level domain is sometimes referred to as a *subdomain*. Common web nomenclature does not typically apply the word *subdomain* when referring to *www*, although technically, this too is a subdomain. A fully qualified domain is the combination of the elements required to identify the location of the server where the content can be found (in this example, *blog.yourdomain.com/*).

These structures can receive individual assignments of importance, trustworthiness, and value from the engines, independent of their second-level domains, particularly on hosted publishing platforms such as WordPress, Blogspot, and so on.

Complete root domains/host domain/pay-level domains (PLDs)/second-level domains

The domain name you need to register and pay for, and the one you point DNS settings toward, is the second-level domain (though it is commonly improperly called the “top-level” domain). In the URL *http://www.yourdomain.com/page*, *yourdomain.com* is the second-level domain. Other naming conventions may refer to this as the “root” or “pay-level” domain.

Figure 6-16 shows some examples.

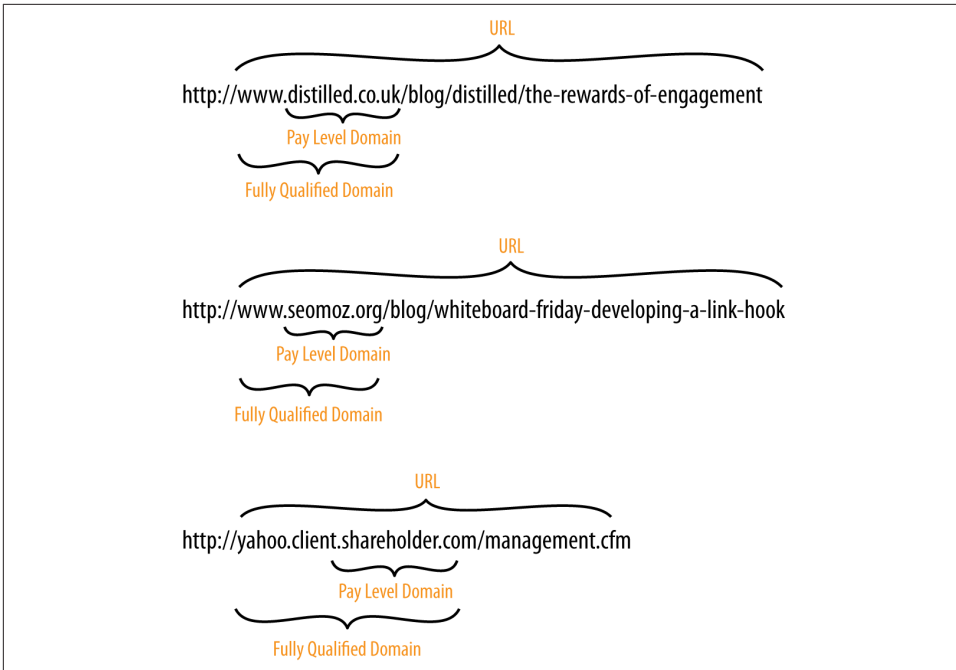


Figure 6-16. *Breaking down some example URLs*

When to Use a Subfolder

If a subfolder will work, it is the best choice 99.9% of the time. Keeping content on a single root domain and single subdomain (e.g., `http://www.yourdomain.com`) gives the maximum SEO benefits, as engines will maintain all of the positive metrics the site earns around links, authority, and trust, and will apply these to every page on the site.

Subfolders have all the flexibility of subdomains (the content *can*, if necessary, be hosted on a unique server or completely unique IP address through post-firewall load balancing) and none of the drawbacks. Subfolder content will contribute directly to how search engines (and users, for that matter) view the domain as a whole. Subfolders can be registered with the major search engine tools and geotargeted individually to specific countries and languages as well.

Although subdomains are a popular choice for hosting content, they are generally not recommended if SEO is a primary concern. Subdomains *may* inherit the ranking benefits and positive metrics of the root domain they are hosted underneath, but they do not always do so (and thus, content can underperform in these scenarios). Of course, there can be exceptions to this general guideline. Subdomains are not inherently harmful, and there are some content publishing scenarios in which they are more

appropriate than subfolders; it is simply preferable for various SEO reasons to use subfolders when possible, as we will discuss next.

When to Use a Subdomain

If your marketing team decides to promote a URL that is completely unique in content or purpose and would like to use a catchy subdomain to do it, using a subdomain can be practical. [Google Maps](#) is an example that illustrates how marketing considerations make a subdomain an acceptable choice. One good reason to use a subdomain is in a situation in which, as a result of creating separation from the main domain, using one looks more authoritative to users.

Subdomains may also be a reasonable choice if keyword usage in the domain name is of critical importance. It appears that search engines do weight keyword usage in the URL somewhat, and have slightly higher benefits for exact matches in the subdomain (or third-level domain name) than subfolders. Note that exact matches in the domain and subdomain carry less weight than they once did. Google updated the weight it assigned to these factors in 2012.¹⁰

Keep in mind that subdomains may inherit very little link equity from the main domain. If you wish to split your site in the subdomains and have all of them rank well, assume that you will have to support each with its own full-fledged SEO strategy.

When to Use a Separate Root Domain

If you have a single, primary site that has earned links, built content, and attracted brand attention and awareness, it is very rarely advisable to place any new content on a completely separate domain. There are rare occasions when this can make sense, and we'll walk through these, as well as explain how singular sites benefit from collecting all of their content in one root domain location.

Splitting similar or relevant content from your organization onto multiple domains can be likened to a store taking American Express Gold cards and rejecting American Express Corporate or American Express Blue—it is overly segmented and dangerous for the consumer mindset. If you can serve web content from a singular domain, that domain will earn branding in the minds of your visitors, references from them, links from other sites, and bookmarks from your regular customers. Switching to a new domain forces you to rebrand and to earn all of these positive metrics all over again.

¹⁰ Christoph C. Cemper, “Deconstructing the Google EMD Update,” Search Engine Land, October 25, 2012, <http://searchengineland.com/google-emd-update-research-and-thoughts-137340>.

Microsites

Although we generally recommend that you do not saddle yourself with the hassle of dealing with multiple sites and their SEO risks and disadvantages, it is important to understand the arguments, if only a few, in favor of doing so.

Optimized properly, a microsite may have dozens or even hundreds of pages. If your site is likely to gain more traction and interest with webmasters and bloggers by being at arm's length from your main site, it may be worth considering—for example, if you have a very commercial main site, and you want to create some great content (perhaps as articles, podcasts, and RSS feeds) that does not fit on the main site.

When should you consider a microsite?

When you own a specific keyword search query domain

For example, if you own *usedtoyotrucks.com*, you might do very well to pull in search traffic for the specific term *used toyota trucks* with a microsite.

When you plan to sell the domains

It is very hard to sell a folder or even a subdomain, so this strategy is understandable if you're planning to churn the domains in the secondhand market.

As discussed earlier, if you're a major brand building a "secret" or buzzworthy microsite

In this case, it can be useful to use a separate domain (however, you should 301-redirect the pages of that domain back to your main site after the campaign is over so that the link authority continues to provide long-term benefit—just as the mindshare and branding do in the offline world).

You should never implement a microsite that acts as a doorway page to your main site, or that has substantially the same content as your main site. Consider a microsite only if you are willing to invest the time and effort to put rich original content on it, and to promote it as an independent site.

Such a site may gain more links by being separated from the main commercial site. A microsite may have the added benefit of bypassing some of the legal and PR department hurdles and internal political battles.

However, a microsite on a brand-new domain can take many months to build enough domain-level link authority to rank in the engines (for more about how Google treats new domains, see ["Determining Searcher Intent and Delivering Relevant, Fresh Content" on page 92](#)). So, what to do if you want to launch a microsite? Start the clock running as soon as possible on your new domain by posting at least a few pages to the URL and then getting at least a few links to it—as far in advance of the official launch as possible. It may take a considerable amount of time before a microsite is able to house enough high-quality content and to earn enough trusted and authoritative links to rank on its own. If the campaign the microsite was created for is time sensitive,

consider redirecting the pages from the microsite to your main site well after the campaign concludes, or at least ensure that the microsite links back to the main site to allow some of the link authority the microsite earns to help the ranking of your main site.

Here are the reasons for not using a microsite:

Search algorithms favor large, authoritative domains

Take a piece of great content about a topic and toss it onto a small, mom-and-pop website; point some external links to it, optimize the page and the site for the target terms, and get it indexed. Now, take that exact same content and place it on Wikipedia, or [CNN.com](#), and you're virtually guaranteed that the content on the large, authoritative domain will outrank the content on the small niche site. The engines' current algorithms favor sites that have built trust, authority, consistency, and history.

Multiple sites split the benefits of links

As suggested in [Figure 6-17](#), a single good link pointing to a page on a domain positively influences the entire domain and every page on it. Because of this phenomenon, it is much more valuable to have any link you can possibly get pointing to the same domain to help boost the rank and value of the pages on it. Having content or keyword-targeted pages on other domains that don't benefit from the links you earn to your primary domain only creates more work.

100 links to Domain A ≠ 100 links to Domain B + 1 link to Domain A (from Domain B)

In [Figure 6-18](#), you can see how earning lots of links to Page G on a separate domain is far less valuable than earning those same links to a page on the primary domain. For this reason, even if you interlink all of the microsites or multiple domains that you build, the value still won't be close to what you could get from those links if they pointed directly to the primary domain.

A large, authoritative domain can host a huge variety of content

Niche websites frequently limit the variety of their discourse and content matter, whereas broader sites can target a wider range of foci. This is valuable not just for targeting the long tail of search and increasing potential branding and reach, but also for viral content, where a broader focus is much less limiting than a niche focus.

Time and energy are better spent on a single property

If you're going to pour your heart and soul into web development, design, usability, user experience, site architecture, SEO, public relations, branding, and so on, you want the biggest bang for your buck. Splitting your attention, time, and resources on multiple domains dilutes that value and doesn't let you build on your past successes on a single domain. As shown in [Figure 6-18](#), every page on a

- When you own the *.com* and want to redirect to an *.org*, *.tv*, *.biz*, and so on, possibly for marketing/branding/geographic reasons. Do this only if you already own the *.com* and can redirect.
- When you can use a *.gov*, *.mil*, or *.edu* domain.
- When you are serving only a single geographic region and are willing to permanently forgo growth outside that region (e.g., *.co.uk*, *.de*, *.it*, etc.).
- When you are a nonprofit and want to distance your organization from the commercial world, *.org* may be for you.

New gTLDs

Many website owners have questions about the new gTLDs (generic top-level domains) that ICANN started assigning in the fall of 2013. Instead of the traditional *.com*, *.net*, *.org*, *.ca*, and so on with which most people are familiar, these new gTLDs range from *.christmas* to *.autos* to *.lawyer* to *.eat* to *.sydney*. A full list of them can be found at <http://newgtlds.icann.org/en/program-status/delegated-strings>. One of the major questions that arises is “Will these help me rank organically on terms related to the TLD?” Currently the answer is no. There is no inherent SEO value in having a TLD that is related to your keywords. Having a *.storage* domain does not mean you have some edge over a *.com* for a storage-related business. In [an online forum](#), Google’s John Mueller stated that these TLDs are treated the same as other generic-level TLDs in that they do not help your organic rankings. He also noted that even the new TLDs that sound as if they are region-specific in fact give you no specific ranking benefit in those regions, though he added that Google reserves the right to change that in the future.

Despite the fact that they do not give you a ranking benefit currently, you should still grab your domains for key variants of the new TLDs. You may wish to consider ones such as *.spam*. You may also wish to register those that relate directly to your business. It is unlikely that these TLDs will give a search benefit in the future, but it *is* likely that if your competition registers your name in conjunction with one of these new TLDs your users might be confused about which is the legitimate site. For example, if you are located in New York City, you should probably purchase your domain name with the *.nyc* TLD; if you happen to own a pizza restaurant, you may want to purchase *.pizza*; and so on.

Optimization of Domain Names/URLs

Two of the most basic parts of any website are the domain name and the URLs for the pages of the website. This section will explore guidelines for optimizing these important elements.

Optimizing Domains

When you're conceiving or designing a new site, one of the critical items to consider is the domain name, whether it is for a new blog, a company launch, or even just a friend's website. Here are 12 indispensable tips for selecting a great domain name:

Brainstorm five top keywords

When you begin your domain name search, it helps to have five terms or phrases in mind that best describe the domain you're seeking. Once you have this list, you can start to pair them or add prefixes and suffixes to create good domain ideas. For example, if you're launching a mortgage-related domain, you might start with words such as *mortgage*, *finance*, *home equity*, *interest rate*, and *house payment*, and then play around until you can find a good match.

Make the domain unique

Having your website confused with a popular site that someone else already owns is a recipe for disaster. Thus, never choose a domain that is simply the plural, hyphenated, or misspelled version of an already established domain. For example, for years Flickr did not own <http://flicker.com>, and the company probably lost traffic because of that. It recognized the problem and bought the domain, and as a result <http://flicker.com> now redirects to <http://flickr.com>.

Choose only dot-com-available domains

If you're not concerned with type-in traffic, branding, or name recognition, you don't need to worry about this one. However, if you're at all serious about building a successful website over the long term, you should be worried about all of these elements, and although directing traffic to a *.net* or *.org* (or any of the other new gTLDs) is fine, owning and 301-ing the *.com*, or the ccTLD for the country your website serves (e.g., *.co.uk* for the United Kingdom), is critical. With the exception of the very tech-savvy, most people who use the Web still make the automatic assumption that *.com* is all that's out there, or that it's more trustworthy. Don't make the mistake of locking out or losing traffic from these folks.

Make it easy to type

If a domain name requires considerable attention to type correctly due to spelling, length, or the use of unmemorable words or sounds, you've lost a good portion of your branding and marketing value. Usability folks even tout the value of having the words include easy-to-type letters (which we interpret as avoiding *q*, *z*, *x*, *c*, and *p*).

Make it easy to remember

Remember that word-of-mouth marketing relies on the ease with which the domain can be called to mind. You don't want to be the company with the terrific

website that no one can ever remember to tell their friends about because they can't remember the domain name.

Keep the name as short as possible

Short names are easy to type and easy to remember (see the previous two rules). Short names also allow more of the URL to display in the SERPs and are a better fit on business cards and other offline media.

Create and fulfill expectations

When someone hears about your domain name for the first time, he should be able to instantly and accurately guess the type of content he might find there. That's why we love domain names such as NYTimes.com, [CareerBuilder.com](#), AutoTrader.com, and WebMD.com. Domains such as Monster.com, Amazon.com, and Zillow.com required far more branding because of their nonintuitive names.

Avoid trademark infringement

This is a mistake that isn't made too often, but it can kill a great domain and a great company when it does. To be sure you're not infringing on anyone's registered trademark with your site's name, visit [the U.S. Patent and Trademark office site](#) and search before you buy. Knowingly purchasing a domain with bad-faith intent that includes a trademarked term is a form of cybersquatting referred to as *domain squatting*.

Set yourself apart with a brand

Using a unique moniker is a great way to build additional value with your domain name. A "brand" is more than just a combination of words, which is why names such as Mortgageforyourhome.com and Shoesandboots.com aren't as compelling as branded names such as [Yelp](#) and [Gilt](#).

Reject hyphens and numbers

Both hyphens and numbers make it hard to convey your domain name verbally and fall down on being easy to remember or type. Avoid spelled-out or Roman numerals in domains, as both can be confusing and mistaken for the other.

Don't follow the latest trends

Website names that rely on odd misspellings, multiple hyphens (such as the SEO-optimized domains of the early 2000s), or uninspiring short adjectives (such as "top x," "best x," and "hot x") aren't always the best choice. This isn't a hard-and-fast rule, but in the world of naming conventions in general, if everyone else is doing it, that doesn't mean it is a surefire strategy. Just look at all the people who named their businesses "AAA x" over the past 50 years to be first in the phone book; how many Fortune 1000s are named "AAA Company?"

Use a domain selection tool

Websites such as **Nameboy** make it exceptionally easy to determine the availability of a domain name. Just remember that you don't have to buy through these services. You can find an available name that you like, and then go to your registrar of choice. You can also try **BuyDomains** as an option to attempt to purchase domains that have already been registered.

Picking the Right URLs

Search engines place some weight on keywords in your URLs. Be careful, however, as the search engines can interpret long URLs with numerous hyphens in them (e.g., *Buy-this-awesome-product-now.html*) as a spam signal. The following are some guidelines for selecting optimal URLs for the pages of your site(s):

Describe your content

An obvious URL is a great URL. If a user can look at the address bar (or a pasted link) and make an accurate guess about the content of the page before ever reaching it, you've done your job. These URLs get pasted, shared, emailed, written down, and yes, even recognized by the engines.

Keep it short

Brevity is a virtue. The shorter the URL, the easier it is to copy and paste, read over the phone, write on a business card, or use in a hundred other unorthodox fashions, all of which spell better usability and increased branding. Remember, however, that you can always create a shortened URL for marketing purposes that redirects to the destination URL of your content—just know that this short URL will have no SEO value.

Static is the way

Search engines treat static URLs differently than dynamic ones. Users also are not fond of URLs in which the big players are `?`, `@`, and `=`. They are just harder to read and understand.

Descriptive text is better than numbers

If you're thinking of using *114/cat223/*, you should go with */brand/adidas/* instead. Even if the descriptive text isn't a keyword or is not particularly informative to an uninitiated user, it is far better to use words when possible. If nothing else, your team members will thank you for making it that much easier to identify problems in development and testing.

Keywords never hurt

If you know you're going to be targeting a lot of competitive keyword phrases on your website for search traffic, you'll want every advantage you can get. Keywords are certainly one element of that strategy, so take the list from marketing,

map it to the proper pages, and get to work. For dynamically created pages through a CMS, create the option of including keywords in the URL.

Subdomains aren't always the answer

First off, never use multiple subdomains (e.g., *product.brand.site.com*); they are unnecessarily complex and lengthy. Second, consider that subdomains have the potential to be treated separately from the primary domain when it comes to passing link and trust value. In most cases where just a few subdomains are used and there's good interlinking, it won't hurt, but be aware of the downsides. For more on this, and for a discussion of when to use subdomains, see "[Root Domains, Subdomains, and Microsites](#)" on page 285.

Fewer folders

A URL should contain no unnecessary folders (or words or characters, for that matter). They do not add to the user experience of the site and can in fact confuse users.

Hyphens separate best

When creating URLs with multiple words in the format of a phrase, hyphens are best to separate the terms (e.g., */brands/dolce-and-gabbana/*), but you can also use plus signs (+).

Stick with conventions

If your site uses a single format throughout, don't consider making one section unique. Stick to your URL guidelines once they are established so that your users (and future site developers) will have a clear idea of how content is organized into folders and pages. This can apply globally as well as for sites that share platforms, brands, and so on.

Don't be case-sensitive

URLs can accept both uppercase and lowercase characters, so don't ever, ever allow any uppercase letters in your structure. Unix/Linux-based web servers are case-sensitive, so *http://www.domain.com/Products/widgets/* is technically a different URL from *http://www.domain.com/products/widgets/*. Note that this is not true in Microsoft IIS servers, but there are a lot of Apache web servers out there. In addition, this is confusing to users, and potentially to search engine spiders as well. Google sees any URLs with even a single unique character as unique URLs. So if your site shows the same content on *www.domain.com/Products/widgets/* and *www.domain.com/products/widgets/*, it could be seen as duplicate content. If you have such URLs now, implement a 301-redirect pointing them to all-lowercase versions, to help avoid confusion. If you have a lot of type-in traffic, you might even consider a 301 rule that sends any incorrect capitalization permutation to its rightful home.

Don't append extraneous data

There is no point in having a URL exist in which removing characters generates the same content. You can be virtually assured that people on the Web will figure it out; link to you in different fashions; confuse themselves, their readers, and the search engines (with duplicate content issues); and then complain about it.

Mobile Friendliness

On April 21, 2015, Google rolled out an update designed to treat the mobile friendliness of a site as a ranking factor. What made this update unique is that it impacted rankings only for people searching from smartphones.

The reason for this update was that the user experience on a smartphone is dramatically different than it is on a tablet or a laptop/desktop device. The main differences are:

- Screen sizes are smaller, so the available space for providing a web page is significantly different.
- There is no mouse available, so users generally use their fingers to tap the screen to select menu items. As a result, more space is needed between links on the screen to make them “tappable.”
- The connection bandwidth is lower, so web pages load more slowly. While having smaller-size web pages helps them load on any device more quickly, this becomes even more important on a smartphone.

To help publishers determine the mobile friendliness of their sites, Google released a tool called **the Mobile-Friendly Test**. In theory, passing this test means that your page is considered mobile-friendly, and therefore would not be negatively impacted for its rankings on smartphones.

There was a lot of debate on the impact of the update. Prior to its release, the industry referred to it as “Mobilegeddon,” but in fact the scope of the update was not nearly that dramatic.

Coauthor Eric Enge led a study to measure the impact of the mobile friendliness update by comparing rankings prior to the update to those after it. This study found that nearly 50% of non-mobile-friendly URLs lost rank. You can see more details from the study at http://bit.ly/enge_mobilegeddon.

Keyword Targeting

Search engines face a tough task: based on a few words in a query (sometimes only one) they must return a list of relevant results ordered by measures of importance,

and hope that the searcher finds what she is seeking. As website creators and web content publishers, you can make this process massively simpler for the search engines and, in turn, benefit from the enormous traffic they send, based on how you structure your content. The first step in this process is to research what keywords people use when searching for businesses that offer products and services like yours.

This practice has long been a critical part of search engine optimization, and although the role keywords play has evolved over time, keyword usage is still one of the first steps in targeting search traffic.

The first step in the keyword targeting process is uncovering popular terms and phrases that searchers regularly use to find the content, products, or services your site offers. There's an art and science to this process, but it consistently begins with a list of keywords to target (see [Chapter 5](#) for more on this topic).

Once you have that list, you'll need to include these keywords in your pages. In the early days of SEO, the process involved stuffing keywords repetitively into every HTML tag possible. Now, keyword relevance is much more aligned with the usability of a page from a human perspective.

Because links and other factors make up a significant portion of the search engines' algorithms, they no longer rank pages with 61 instances of *free credit report* above pages that contain only 60. In fact, *keyword stuffing*, as it is known in the SEO world, can actually get your pages devalued via search engine penalties. The engines don't like to be manipulated, and they recognize keyword stuffing as a disingenuous tactic.

[Figure 6-19](#) shows an example of a page utilizing accurate keyword targeting.

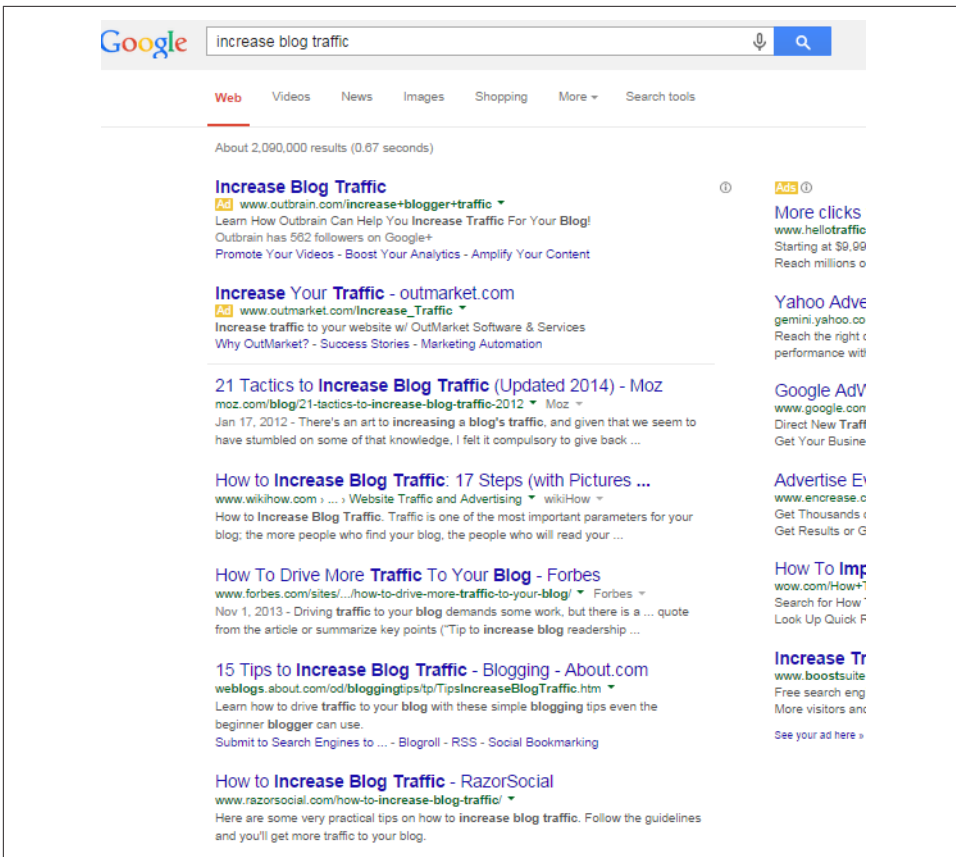


Figure 6-19. Title and headings tags—powerful for SEO

Keyword usage includes creating titles, headlines, and content designed to appeal to searchers in the results (and entice clicks), as well as building relevance for search engines to improve your rankings. In today’s SEO, there are also many other factors involved in ranking, including term frequency—inverse document frequency (TF-IDF), co-occurrence, entity salience, page segmentation, and several others, which will be described in detail later in this chapter.

However, keywords remain important, and building a search-friendly site requires that you prominently employ the keywords that searchers use to find content. Here are some of the more prominent places where a publisher can place those keywords.

HTML <title> Tags

For keyword placement, <title> tags are an important element for search engine relevance. The <title> tag is in the <head> section of an HTML document, and is the only piece of meta information about a page that directly influences relevancy and ranking.

The following nine rules represent best practices for <title> tag construction. Do keep in mind, however, that a <title> tag for any given page must directly correspond to that page's content. You may have five different keyword categories and a unique site page (or section) dedicated to each, so be sure to align a page's <title> tag content with its actual visible content as well.

Place your keywords at the beginning of the <title> tag

This positioning provides the most search engine benefit; thus, if you want to employ your brand name in the <title> tag as well, place it at the end. There is a trade-off here, however, between SEO benefit and branding benefit that you should think about: major brands may want to place their brand at the start of the <title> tag, as it may increase click-through rates. To decide which way to go, you need to consider which need is greater for your business.

Limit length to 50 characters (including spaces)

Content in <title> tags after 50 characters is probably given less weight by the search engines. In addition, the display of your <title> tag in the SERPs may get cut off as early as 49 characters.¹¹

There is no hard-and-fast rule for how many characters Google will display. Google now truncates the display after a certain number of pixels, so the exact characters you use may vary in width. At the time of this writing, this width varies from 482px to 552px depending on your operating system and platform.¹²

Also be aware that Google may not use your <title> tag in the SERPs. Google frequently chooses to modify your <title> tag based on several different factors that are beyond your control. If this is happening to you, it may be an indication that Google thinks that your <title> tag does not accurately reflect the contents of the page, and you should probably consider updating either your <title> tags or your content.

Incorporate keyword phrases

This one may seem obvious, but it is critical to prominently include in your <title> tag the keywords your research shows as being the most valuable for capturing searches.

Target longer phrases if they are relevant

When choosing what keywords to include in a <title> tag, use as many as are completely relevant to the page at hand while remaining accurate and descriptive.

11 Dr. Peter J. Meyers, "New Title Tag Guidelines & Preview Tool," Moz Blog, March 20, 2014, <https://moz.com/blog/new-title-tag-guidelines-preview-tool>.

12 Dan Sharp, "An Update on Pixel Width in Google SERP Snippets," May 15, 2014, <http://www.screamingfrog.co.uk/an-update-on-pixel-width-in-google-serp-snippets/>.

Thus, it can be much more valuable to have a <title> tag such as “SkiDudes | Downhill Skiing Equipment & Accessories” rather than simply “SkiDudes | Skiing Equipment.” Including those additional terms that are both relevant to the page and receive significant search traffic can bolster your page’s value.

However, if you have separate landing pages for “skiing accessories” versus “skiing equipment,” don’t include one term in the other’s title. You’ll be cannibalizing your rankings by forcing the engines to choose which page on your site is more relevant for that phrase, and they might get it wrong. We will discuss the cannibalization issue in more detail shortly.

Use a divider

When you’re splitting up the brand from the descriptive text, options include | (a.k.a. the pipe), >, -, and :, all of which work well. You can also combine these where appropriate—for example, “Major Brand Name: Product Category – Product.” These characters do not bring an SEO benefit, but they can enhance the readability of your title.

Focus on click-through and conversion rates

The <title> tag is exceptionally similar to the title you might write for paid search ads, only it is harder to measure and improve because the stats aren’t provided for you as easily. However, if you target a market that is relatively stable in search volume week to week, you can do some testing with your <title> tags and improve the click-through rate.

Watch your analytics and, if it makes sense, buy search ads on the page to test click-through and conversion rates of different ad text as well, even if it is for just a week or two. You can then look at those results and incorporate them into your titles, which can make a huge difference in the long run. A word of warning, though: don’t focus entirely on click-through rates. Remember to continue measuring conversion rates.

Target searcher intent

When writing titles for web pages, keep in mind the search terms your audience employed to reach your site. If the intent is browsing or research-based, a more descriptive <title> tag is appropriate. If you’re reasonably sure the intent is a purchase, download, or other action, make it clear in your title that this function can be performed at your site. Here is an example from <http://www.bestbuy.com/site/video-games/playstation-4-ps4/pcmcat295700050012.c?id=pcmcat295700050012>. The <title> tag of that page is “PS4: PlayStation 4 Games & Consoles - Best Buy.” The <title> tag here makes it clear that you can buy PS4 games and consoles at Best Buy.

Communicate with human readers

This needs to remain a primary objective. Even as you follow the other rules here to create a <title> tag that is useful to the search engines, remember that humans will likely see your <title> tag presented in the search results for your page. Don't scare them away with a <title> tag that looks like it's written for a machine.

Be consistent

Once you've determined a good formula for your pages in a given section or area of your site, stick to that regimen. You'll find that as you become a trusted and successful "brand" in the SERPs, users will seek out your pages on a subject area and have expectations that you'll want to fulfill.

Meta Description Tags

Meta descriptions have three primary uses:

- To describe the content of the page accurately and succinctly
- To serve as a short text "advertisement" to prompt searchers to click on your pages in the search results
- To display targeted keywords, not for ranking purposes, but to indicate the content to searchers

Great meta descriptions, just like great ads, can be tough to write, but for keyword-targeted pages, particularly in competitive search results, they are a critical part of driving traffic from the engines through to your pages. Their importance is much greater for search terms where the intent of the searcher is unclear or different searchers might have different motivations.

Here are six good rules for meta descriptions:

Tell the truth

Always describe your content honestly. If it is not as "sexy" as you'd like, spice up your content; don't bait and switch on searchers, or they'll have a poor brand association.

Keep it succinct

Be wary of character limits—currently Google displays as few as 140 characters, Yahoo! up to 165, and Bing up to 200+ (it'll go to three vertical lines in some cases). Stick with the smallest—Google—and keep those descriptions at 140 characters (including spaces) or less.

Write ad-worthy copy

Write with as much sizzle as you can while staying descriptive, as the perfect meta description is like the perfect ad: compelling and informative.

Analyze psychology

The motivation for an organic-search click is frequently very different from that of users clicking on paid results. Users clicking on PPC ads may be very directly focused on making a purchase, and people who click on an organic result may be more interested in research or learning about the company. Don't assume that successful PPC ad text will make for a good meta description (or the reverse).

Include relevant keywords

It is extremely important to have your keywords in the meta description tag—the boldface that the engines apply can make a big difference in visibility and click-through rate. In addition, if the user's search term is not in the meta description, chances are reduced that the meta description will be used as the description in the SERPs.

Don't employ descriptions universally

You shouldn't always write a meta description. Conventional logic may hold that it is usually wiser to write a good meta description yourself to maximize your chances of it being used in the SERPs, rather than let the engines build one out of your page content; however, this isn't always the case. If the page is targeting one to three heavily searched terms/phrases, go with a meta description that hits those users performing that search.

However, if you're targeting longer-tail traffic with hundreds of articles or blog entries or even a huge product catalog, it can sometimes be wiser to let the engines themselves extract the relevant text. The reason is simple: when engines pull, they always display the keywords (and surrounding phrases) that the user searched for. If you try to force a meta description, you can detract from the relevance that the engines make naturally. In some cases, they'll overrule your meta description anyway, but because you can't consistently rely on this behavior, opting out of meta descriptions is OK (and for massive sites, it can save hundreds or thousands of man-hours). Because the meta description isn't a ranking signal, it is a second-order activity at any rate.

Heading Tags

The heading tags in HTML (<h1>, <h2>, <h3>, etc.) are designed to indicate a headline hierarchy in a document. Thus, an <h1> tag might be considered the headline of the page as a whole, whereas <h2> tags would serve as subheadings, <h3>s as tertiary-level subheadings, and so forth. The search engines have shown a slight preference for keywords appearing in heading tags. Generally when there are multiple heading tags on a page, the engines will weight the higher-level heading tags heavier than those below them. For example, if the page contains <h1>, <h2>, and <h3> tags, the <h1> will be weighted the heaviest. If a page contains only <h2> and <h3> tags, the <h2> would be weighted the heaviest.

In some cases, you can use the <title> tag of a page, containing the important keywords, as the <h1> tag. However, if you have a longer <title> tag, you may want to use a more focused, shorter heading tag including the most important keywords from the <title> tag. When a searcher clicks a result from the engines, reinforcing the search term he just typed in with the prominent headline helps to indicate that he has arrived on the right page with the same content he sought.

Many publishers assume that they have to use an <h1> tag on every page. What matters most, though, is the highest-level heading tag you use on a page, and its placement. If you have a page that uses an <h3> heading at the very top, and any other heading tags further down on the page are <h3> or lower level, then that first <h3> tag will carry just as much weight as if it were an <h1>.

Again, what matters most is the semantic markup of the page, and the first heading tag presumably is intended to be a label for the entire page (so it plays a complementary role to the <title> tag), and you should treat it as such. Other heading tags on the page should be used to label subsections of the content.

It's also a common belief that the size at which the heading tag is displayed is a factor. For the most part, the styling of your heading tags is not a factor in the SEO weight of the heading tag. You can style the tag however you want, as shown in [Figure 6-20](#), provided that you don't go to extremes (because it acts as a title for the whole page, it should probably be the largest text element on the page).



Figure 6-20. Headings styled to match the site

Document Text

The HTML text on a page was once the center of keyword optimization activities. In the early days of SEO, metrics such as keyword density and keyword saturation were used to measure the perfect level of keyword usage on a page. To the search engines, however, text in a document, particularly the frequency with which a particular term or phrase is used, has very little impact on how happy a searcher will be with that page.

In fact, quite often a page laden with repetitive keywords attempting to please the engines will provide a very poor user experience, and this can result in lower rankings instead of higher ones. It's much more valuable to create semantically rich content that covers the topic matter implied by the page's <title> tag in a comprehensive way. This means naturally including synonyms, and covering related topic areas in a manner that increases the chances of satisfying the needs of a large percentage of visitors to that page. It's a good idea to use the main keyword for a page in the <title> tag and the main heading tag. It might also appear in the main content, but the use of synonyms for the main keyword and related concepts is at least as important. As a result, it's more important to focus on creating high-quality content than it is to keep repeating the main keyword.

Term frequency—Inverse document frequency

TF-IDF consists of two parts. The first is *term frequency*, which relates to the frequency of usage of a keyword or key phrase on a page, in comparison to usage levels in competing documents. This is similar to keyword density, except weighting is done logarithmically to reduce the impact of keyword repetition. The result is that a page which uses a phrase 10 times might be seen only as twice as good a match as a page that uses that phrase once.

Term frequency analysis can be very useful in understanding how your page compares semantically with pages that rank highly in Google's results. Coauthor Eric Enge has written an [article about this](#).

Inverse document frequency (IDF) is more about identifying the uniqueness of a term. For example, the word "omnipotent" is used much less on the web than the word "powerful." Therefore, a page using the word "omnipotent" may be seen as a bit more unique. If a user enters the word "omnipotent" as part of a search query, it will be far more likely to surface a page using that word in the results. IDF can be a very way to identify new ranking opportunities for your web page as coauthor Eric Enge [has written elsewhere](#).

TF-IDF helps search engines understand what terms a page emphasizes most, and what terms most uniquely define a page at the same time. Publishers can use TF-IDF

analysis on competing pages ranking in the top 10 for a given search term to learn what search engines appear to value the most in content for a given search query.

Used properly, this is not about keyword stuffing, but instead focuses on learning key information being sought out by users in relation to a search query. For example, if someone searches on “oil filters” and lands on your page, he may also want information on oil filter wrenches.

Using TF-IDF analysis on competing pages can help you learn about such opportunities to improve the user experience of a page, and help you with SEO for that page at the same time.

Page segmentation

It used to be that Google could not understand the layout of a page that well, simply because it could not read CSS files and process them like a browser does. However, that has changed, as documented in [a post on the Google Webmaster Central Blog](#).

As a result, Google is quite likely to fully understand the layout of your pages. Given this, where the keywords are used on the page also matters. Use of keywords in your left or right sidebar, or your footer, probably matters less than the content used in the main body of your page.

In addition, with HTML5, new markup exists that allows you to explicitly identify the section of your page that represents the main content. You can use this markup to help make Google’s job easier, and to make sure that other search engines are able to locate that content.

Synonyms

Use of related terms is also a factor. A page about “left-handed golf clubs” should not use that exact phrase every time the product is referenced. This would not be a natural way of writing, and could be interpreted by the search engines as a signal of poor document quality, lowering the page’s rankings.

Instead, allow your content creators to write naturally. This will cause them to use other phrases, such as “the sticks,” “set of clubs,” “lefty clubs,” and other variants that people use in normal writing style.

Using synonyms represents a key step away from manipulative SEO techniques for creating pages to try to rank for specific search terms.

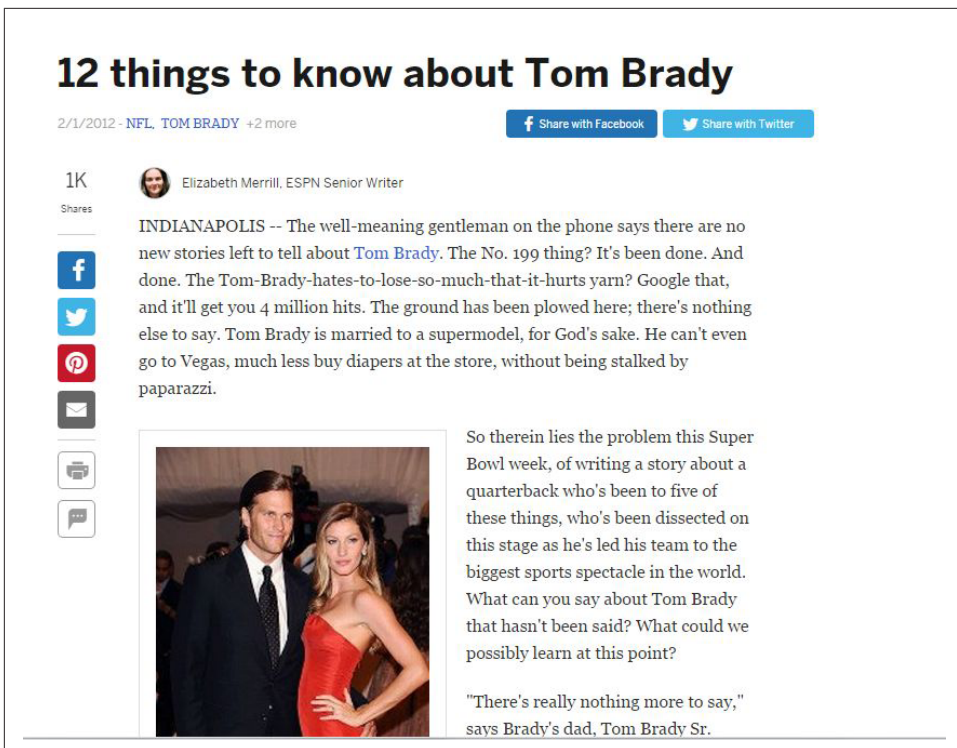
Co-occurrence, phrase-based indexing, and entity salience

The presence of a specific keyword or phrase on a page will likely increase the probability of finding other words on those pages. For example, if you are reading an article

on the life of New England Patriots quarterback Tom Brady, you would expect to see mentions of Gisele Bündchen, or his children. The absence of such mentions could be a signal of a poor-quality page.

To put this in the positive, inclusion of more information that you would expect to see in the content can be interpreted as an indication that it's a good page. Chances are good that more people reading the article will be satisfied with it as well, as they might expect to learn about Tom Brady's family.

Of course, a high-quality article will probably talk about his parents, his high school football coach, his sisters, and all the other aspects of his life as well. The key is to focus on providing a more complete response to the topic than others covering the same topic might do. The ESPN article shown in [Figure 6-21](#) is an example of such an in-depth article.



The image shows a screenshot of an ESPN article. The title is "12 things to know about Tom Brady" in a large, bold, black font. Below the title, the date "2/1/2012" and the author "NFL, TOM BRADY +2 more" are visible. There are two blue buttons for social sharing: "Share with Facebook" and "Share with Twitter". The author's name, "Elizabeth Merrill, ESPN Senior Writer", is displayed next to her profile picture. The article text begins with "INDIANAPOLIS -- The well-meaning gentleman on the phone says there are no new stories left to tell about Tom Brady. The No. 199 thing? It's been done. And done. The Tom-Brady-hates-to-lose-so-much-that-it-hurts yarn? Google that, and it'll get you 4 million hits. The ground has been plowed here; there's nothing else to say. Tom Brady is married to a supermodel, for God's sake. He can't even go to Vegas, much less buy diapers at the store, without being stalked by paparazzi." To the right of the text, there is a photo of Tom Brady and Gisele Bündchen. Below the photo, the text continues: "So therein lies the problem this Super Bowl week, of writing a story about a quarterback who's been to five of these things, who's been dissected on this stage as he's led his team to the biggest sports spectacle in the world. What can you say about Tom Brady that hasn't been said? What could we possibly learn at this point?" At the bottom of the article, a quote reads: "There's really nothing more to say," says Brady's dad, Tom Brady Sr.

Figure 6-21. A comprehensive article on Tom Brady from ESPN

Not all of the topic needs to be addressed on each individual page. Linking to other relevant resources and high-quality content, both on your site as well as on third-party sites, can play a key role in establishing your page as a great answer to the user's question.

This last step may well be equally important in the overall page optimization process. No single page will answer every question from every possible user, so addressing a significant percentage of questions, and then connecting with other pages to answer follow-on questions on the same topic, is an optimal structure.

On the product pages of an ecommerce site, where there will not be article-style content, this can mean a combination of well-structured and unique description text and access to key refinements, such as individual brands, related product types, the presence of a privacy policy, “About us” information, a shopping cart, and more.

Image Filenames and alt Attributes

Incorporating images on your web pages can substantively enrich the user experience. However, the search engines cannot read the images directly. There are two elements that you can control to give the engines context for images:

The filename

Search engines look at the image filename to see whether it provides any clues to the content of the image. Don’t name your image *example.com/img4137a-b12.jpg*, as it tells the search engine nothing at all about the image, and you are passing up the opportunity to include keyword-rich text.

If it is a picture of Abe Lincoln, name the file *abe-lincoln.jpg* and/or have the src URL string contain it, as in *example.com/abe-lincoln/portrait.jpg*.

The alt attribute text

Image tags in HTML permit you to specify the alt attribute. This is a place where you can provide more information about what is in the image, and again where you can use your targeted keywords. Here is an example for the picture of Abe Lincoln:

```

```

Use the quotes if you have spaces in the text string of the alt content! Sites that have invalid tags frequently lump a few words without quotes into the tag, intended for the alt content—but with no quotes, all terms after the first word will be lost.

This usage of the image filename and the alt attribute permits you to reinforce the major keyword themes of the page. This is particularly useful if you want to rank in image search. Make sure the filename and the alt text reflect the content of the picture, and do not artificially emphasize keywords unrelated to the image (even if they are related to the page). Although the alt attribute and the image filename are helpful, you should not use image links as a substitute for text links with rich anchor text, which carry much more weight from an SEO perspective.

Presumably, your picture will relate very closely to the content of the page, and using the image filename and the alt text will help reinforce the page's overall theme.

Boldface Text

While it used to be true that including keywords in bold text had a very slight effect in rankings, this is no longer the case.

Keyword Cannibalization

As we discussed earlier, you should not use common keywords across multiple page titles. This advice applies to more than just the <title> tags.

One of the nastier problems that often crops up during the course of a website's information architecture, *keyword cannibalization* refers to a site's targeting of popular keyword search phrases on multiple pages, forcing the engines to pick which one is most relevant. In essence, a site employing cannibalization competes with itself for rankings and dilutes the ranking power of internal anchor text, external links, and keyword relevancy.

Avoiding cannibalization requires strict site architecture with attention to detail. Plot out your most important terms on a visual flowchart (or in a spreadsheet file, if you prefer), and pay careful attention to what search terms each page is targeting. Note that when pages feature two-, three-, or four-word phrases that contain the target search phrase of another page, linking back to that page within the content with the appropriate anchor text will avoid the cannibalization issue.

For example, if you had a page targeting "mortgages" and another page targeting "low-interest mortgages," you would link back to the "mortgages" page from the "low-interest mortgages" page using the anchor text "mortgages" (see [Figure 6-22](#)). You can do this in the breadcrumb or in the body copy. *The New York Times* does the latter, where keywords in the body copy link to the related resource page on the site.

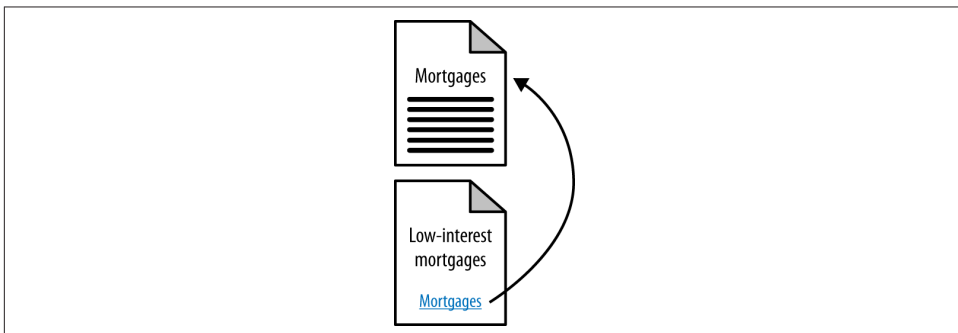


Figure 6-22. Adding lots of value with relevant cross-links

Keyword Targeting in Content Management Systems and Automatically Generated Content

Large-scale publishing systems, or those that produce automatically generated content, present some unique challenges. If hundreds of pages are being created every day, it is not feasible to do independent keyword research on each and every page, making page optimization an interesting challenge.

In these scenarios, the focus turns to methods/recipes for generating unique titles, heading tags, and content for each page. It is critical to educate the writers on ways to implement titles and headings that capture unique, key aspects of the articles' content. More advanced teams can go further with this and train their writing staff on the use of keyword research tools to optimize this process even more.

In the case of automatically generated material (such as that produced from algorithms that mine data from larger textual bodies), the key is to automate means for extracting a short (fewer than 55 characters) description of the article and making it unique from other titles generated elsewhere on the site and on the Web at large.

Effective Keyword Targeting by Content Creators

Very frequently, someone other than an SEO professional is responsible for content creation. Content creators often do not have an innate knowledge of how SEO works, or worse, they may think they know how it works, but have the wrong idea about it. Some training for your writers is critical. This is particularly important when you're dealing with large websites and large teams of writers.

Here are the main components of web page copywriting that your writers must understand:

- Search engines look to match up a user’s search queries with the keyword phrases, their synonyms, and related concepts on your web pages. If some combination of all of these does not appear on the page, chances are good that your page will never achieve significant ranking for those search phrases.
- The search phrases users may choose to use when looking for something are infinite in variety, but certain phrases will be used much more frequently than others.
- Using the more popular phrases you wish to target on a web page in the content for that page is essential to SEO success for that page.
- Make sure that the writers understand the concepts of co-occurrence and entity salience, discussed earlier in this chapter, so they don’t create content that uses the main keyword excessively. They need to focus on creating semantically rich content that stays on the topic of the main target keyword phrase for the page, while still writing naturally.
- The <title> tag is the most important element on the page. Next is the first header (usually <h1>), and then the main body of the content.
- There are tools (as outlined in [Chapter 5](#)) that allow you to research and determine what the most interesting phrases are.

If you can get these six points across, you are well on your way to empowering your content creators to perform solid SEO. The next key element is training them on how to pick the right keywords to use.

This can involve teaching them how to use keyword research tools similar to the ones we discussed in [Chapter 5](#), or having the website’s SEO person do the research and provide the terms to the writer.

The most important factor to reiterate to content creators is that content quality and user experience still come first. Then, by intelligently making sure the right keywords and phrases are properly used throughout the content, they can help bring search engine traffic to your site. Reverse these priorities, and you can end up with keyword stuffing or other spam issues.

Long-Tail Keyword Targeting

As we outlined in [Chapter 5](#), the small-volume search terms, when tallied up, represent 70% or more of overall search traffic, and the more obvious, high-volume terms represent only 30%.

For example, if you run a site targeting searches for *new york pizza* and *new york pizza delivery*, you might be surprised to find that hundreds of single searches each day for terms such as *pizza delivery on the corner of 57th & 7th*, or *Manhattan’s tastiest Italian-style*

sausage pizza, when taken together, will actually provide considerably more traffic than the popular phrases you've researched. As we covered in [Chapter 5](#), this concept is called the *long tail* of search.

Targeting the long tail is another aspect of SEO that combines art and science. In [Figure 6-23](#), you may not want to implement entire web pages for a history of pizza dough, pizza with white anchovies, or Croatian pizza.

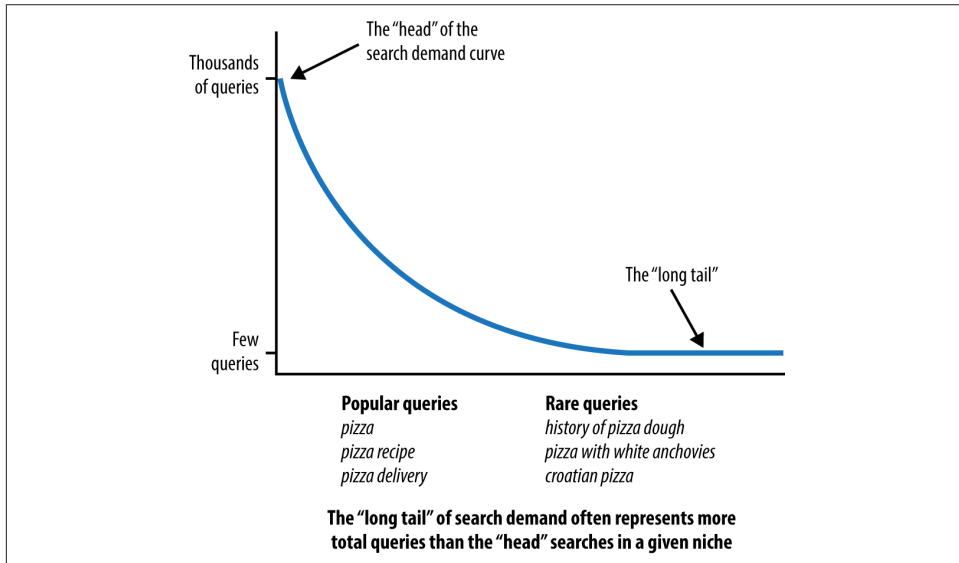


Figure 6-23. Example of the long-tail search curve

Finding scalable ways to chase long-tail keywords is a complex topic. It is also one where many publishers get into a lot of trouble, as they think they need to create a new page for each potential search phrase that a user might type in related to their business, and this is not the case. You can address much of the long tail of search by using the right content optimization practices on your site.

Perhaps you have a page for ordering pizza in New York City, and you have a good title and heading tag on the page (e.g., "New York City Pizza: Order Here"), a phone number and a form for ordering the pizza, and no other content. If that is all you have, that page is not competing effectively for rankings on long-tail search terms. To fix this, you need to write additional content for the page. Ideally, this would be content that talks about the types of pizza that are popular in New York City, the ingredients used, and other related topics that might draw in long-tail search traffic.

If you have a page for San Jose pizza, the picture gets even more complicated. You don't want your content on the San Jose page to be the same as it is on the New York City page. This presents potential duplicate content problems, as we will outline in

“Duplicate Content Issues” on page 320, or the keyword cannibalization issues we discussed earlier in this chapter.

To maximize your success, find a way to generate different content for those two pages, ideally tuned to the specific needs of the audience that arrives at those pages. Perhaps the pizza preferences of the San Jose crowd are different from those in New York City. Of course, the geographic information is inherently different between the two locations, so driving directions from key locations might be a good thing to include on the page.

If you have pizza parlors in 100 cities, this can get very complex indeed. The key here is to remain true to the diverse needs of your users, yet use your knowledge of the needs of search engines and searcher behavior to obtain that long-tail traffic.

Content Optimization

Content optimization relates to how the presentation and architecture of the text, image, and multimedia content on a page can be optimized for search engines. Many of these recommendations are second-order effects. Having the right formatting or display won't boost your rankings directly, but through it, you're more likely to earn links, get clicks, and eventually benefit in search rankings. If you regularly practice the techniques in this section, you'll earn better consideration from the engines and from the human activities on the Web that influence their algorithms.

Content Structure

Because SEO has become such a holistic part of website development and improvement, it is no surprise that *content formatting*—the presentation, style, and layout choices you select for your content—is a part of the process. A browser-safe sans serif font such as Arial or Helvetica is a wise choice for the Web; Verdana in particular has received high praise from usability/readability experts (for a full discussion of this topic, see <http://webaim.org/techniques/fonts/>).

Verdana is one of the most popular of the fonts designed for on-screen viewing. It has a simple, straightforward design, and the characters or glyphs are not easily confused. For example, the uppercase *I* and the lowercase *L* have unique shapes, unlike in Arial, in which the two glyphs may be easily confused (see [Figure 6-24](#)).

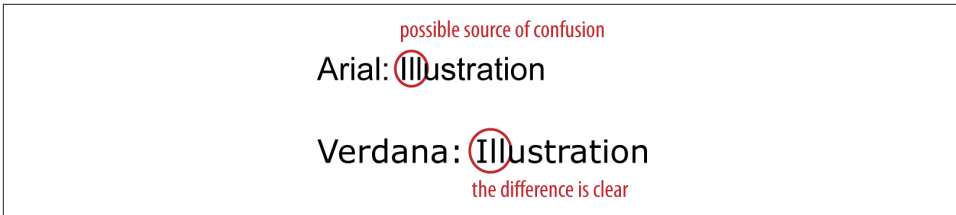


Figure 6-24. *Arial and Verdana font comparison*

Another advantage of Verdana is the amount of spacing between letters. One consideration to take into account with Verdana is that it is a relatively large font. The words take up more space than words in Arial, even at the same point size (see [Figure 6-25](#)).

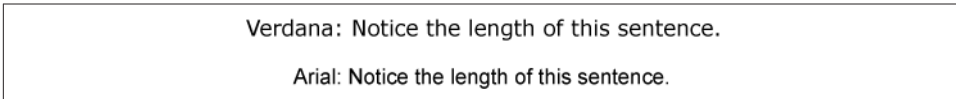


Figure 6-25. *How fonts impact space requirements*

The larger size improves readability but also can potentially disrupt carefully planned page layouts.

In addition to font choice, sizing and contrast issues are important considerations. Type that is smaller than 10 points is typically very challenging to read, and in all cases, relative font sizes are recommended so that users can employ browser options to increase/decrease size if necessary. Contrast—the color difference between the background and text—is also critical; legibility usually drops for anything that isn’t black (or very dark) on a white background.

Content length and word count

Content length is another critical piece of the optimization puzzle that’s mistakenly placed in the “keyword density” or “unique content” bucket of SEO. In fact, content length can play a big role in terms of whether your material is easy to consume and easy to share.

People often ask about the ideal length for a piece of content. The reality is that the perfect length for a piece of content is determined by the nature of the topic being addressed. Many pieces of content do well because they are short and very easy to consume. On the other hand, some content will fare best when it’s lengthy and comprehensive in nature.

Longer articles also have the opportunity to show up in [Google’s in-depth articles section](#). In some cases, where appropriate, Google will feature several longer, more detailed articles on a given topic. For more information on having your articles appear as “in-depth” articles, see http://bit.ly/in-depth_articles.

Visual layout

Last but not least in content structure optimization is the display of the material. Beautiful, simple, easy-to-use, and consumable layouts instill trust and garner far more readership and links than poorly designed content wedged between ad blocks that threaten to overtake the page. For more on this topic, check out “[The Golden Ratio in Web Design](#)” from [NetTuts](#), which has some great illustrations and advice on laying out web content on the page.

CSS and Semantic Markup

CSS is commonly mentioned as a best practice for general web design and development, but its principles provide some indirect SEO benefits as well. Google used to recommend keeping pages smaller than 101 KB, and it was a common belief that there were benefits to implementing pages that were small in size. Now, however, search engines deny that code size is a factor at all, unless it is extreme. Still, keeping file size low means your pages have faster load times, lower abandonment rates, and a higher probability of being fully read and more frequently linked to. This is particularly important in mobile environments.

It also used to be the case that search engines could not read CSS code and render pages in the same manner as a browser does. In October 2014, [Google made it clear that it is able to do just that](#), so good clean page layout, as set up by your CSS, could potentially be considered as a factor in evaluating page quality.

Your experience may vary, but good CSS makes it easy, so there’s no reason not to make it part of your standard operating procedure for web development. Use tableless CSS stored in external files, keep JavaScript calls external, and separate the content layer from the presentation layer, as shown on [CSS Zen Garden](#), a site that offers many user-contributed stylesheets for formatting the same HTML content.

You can use CSS code to provide emphasis, to quote/reference, and to reduce the use of tables and other bloated HTML mechanisms for formatting, which can positively impact your SEO. Be sure to allow Googlebot access to your CSS files.

Google, Bing, and Yahoo! have come together to sponsor a standard for markup called [Schema.org](#).¹³ This represented a new level of commitment from the search engines to the concept of marking up content, or more broadly, to allowing the publisher to provide information about the content to the search engines. By “marking up,” content, we mean tagging your content using XML tags to categorize it. For example, you may label a block of content as containing a recipe, and another block of content as containing a review.

¹³ You can see a copy of the announcement at http://bit.ly/intro_schema_org.

This notion of advanced markup was not new, as all of the search engines have supported semantic markup at a limited level, and have used this markup to show *rich snippets*, an example of which is shown in [Figure 6-26](#).

One of the original ways a publisher had to communicate information about a web page to search engines was with metatags. Unfortunately, these were so badly abused by spammers that Google stopped using them as a ranking signal. Google confirmed this publicly in a post in 2009, which noted that “Google has ignored the keywords meta tag for years and currently we see no need to change that policy.”¹⁴

Google used to publicly state that it does not use markup as a ranking factor, and while those statements are no longer being publicly made, there continues to be no evidence that it has been made a ranking factor. However, there are important SEO benefits to using markup.

Markup in search results

As previously mentioned, markup is sometimes used by search engines to create a rich snippet. [Figure 6-26](#) shows an example of rich snippets in the search results for a recipe for a Cambodian dish called Loc Lac.

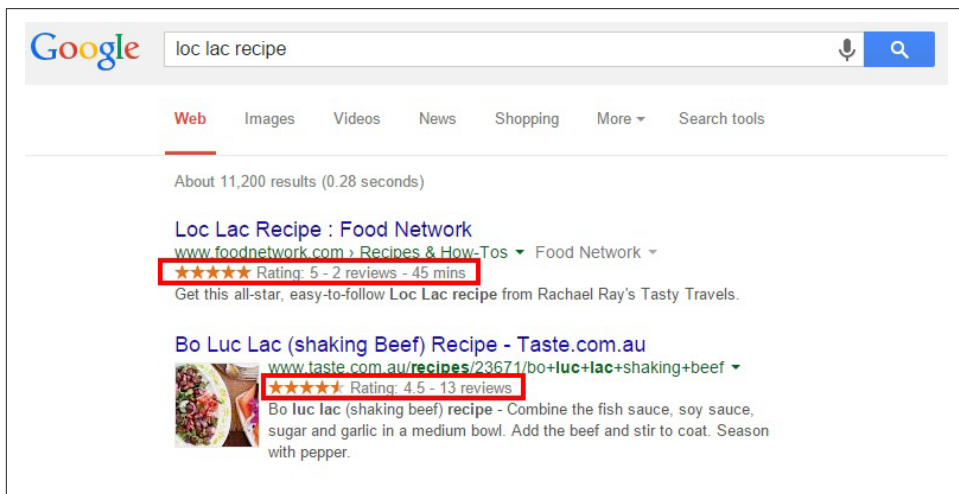


Figure 6-26. Example of recipe rich snippet on Google

Based on the markup that Google found in the HTML, it has enhanced the result by showing the recipe reviews (the number of stars), the required cooking time, and the

¹⁴ Webmaster Central Blog, “Google does not use the keywords meta tag in web ranking,” September 21, 2009, http://bit.ly/keywords_meta_google.

calories of the meal. The type of markup used for this example is called *microformats*. Figure 6-27 shows what the source code looks like for this example.

```
</div>
<div id="zoneRecipe" itemscope="" itemType="http://schema.org/Recipe" data-itemtype="Recipe" data-itemtypeid="1"
data-typespecificid="15869">
  <link itemprop="url" href="http://allrecipes.com/recipe/cream-puffs/" />
  <meta itemprop="mainEntityOfPage" content="True" />
  <div class="detail-section greydotted ingredients">
    <!-- Picture, Title, Description, etc. -->
    <div id="divHeroPhotoContainer" class="detail-left fl-left testB">
      <a href="/recipe/cream-puffs/photo-gallery.aspx" id="lnkOpenCarousel" class="modal-link unsavedExempt
open_modal-recipe-videos frame" rel="modal-recipe-photos" data-close-layer-on-login="" calltoaction="" style=""></a>
```

Figure 6-27. Sample of microformats code for a recipe

Supported types of markup

There are a few different standards for markup. The most common ones are *microdata*, *microformats*, and *RDFa*. Schema.org is based off of the microdata standard. However, the search engines have implemented rich snippets based on some (but not all) aspects of microformats prior to the announcement of Schema.org, and they will likely continue support for these for some period of time.

It is likely that any new forms of rich snippets implemented by the search engines will be based off of Schema.org (microdata), not microformats or RDFa. Some of the formats already supported by Google include:

- People
- Products
- Events
- Business and organizations
- Video

Impact of rich snippets

The key reason that the search engines are pursuing rich snippets is that they have done extensive testing that has proven that rich snippets can increase click-through rates. Searchers like seeing more information about the page in the search results. Thus, you can expect that the search engines will continue to implement support for more of these types of search result enhancements based on markup.

From an SEO perspective, increasing click-through rate is highly desirable; it brings us more relevant traffic. In addition, we know that search engines measure user interaction with the search results and that click-through rate is a ranking factor. This was first publicly confirmed in [an interview with Bing's Duane Forrester](#).

So, while the search engines do not use semantic markup directly as a ranking signal, the indirect impact of rich snippets providing a higher click-through rate acts as a ranking signal.

For more information on semantic markup, see the sections “[Semantic Search](#)” on [page 381](#) and “[Schema.org](#)” on [page 386](#).

Content Uniqueness and Depth

Few can debate the value the engines place on robust, unique, value-added content—Google in particular has had several rounds of kicking low-quality-content sites out of its indexes, and the other engines have followed suit.

The first critical designation to avoid is *thin content*—a phrase that (loosely) refers to a page the engines do not feel contributes enough unique content to warrant the page’s inclusion in the search results. How much content is enough content to not be considered thin? The criteria have never been officially listed, but here are some examples gathered from engineers and search engine representatives:

- At least 30 to 50 unique words, forming unique, parsable sentences that other sites/pages do not have (for many pages much more is appropriate, so consider this a minimum).
- Unique HTML text content, different from other pages on the site in more than just the replacement of key verbs and nouns (yes, this means all those webmasters who build the same page and just change the city and state names thinking it is “unique” are mistaken).
- Unique titles and meta description elements. If you can’t write unique meta descriptions, just exclude them. Algorithms can trip up pages and boot them from the index simply for having near-duplicate meta tags.
- Unique video/audio/image content. The engines have started getting smarter about identifying and indexing pages for vertical search that wouldn’t normally meet the “uniqueness” criteria.

NOTE

By the way, you can often bypass these limitations if you have a good quantity of high-value external links pointing to the page in question (though this is very rarely scalable) or an extremely powerful, authoritative site (note how many one-sentence Wikipedia stub pages still rank).

The next criterion from the engines demands that websites “add value” to the content they publish, particularly if it comes from (wholly or partially) a secondary source.

A word of caution to affiliates

This word of caution most frequently applies to affiliate sites whose republishing of product descriptions, images, and so forth has come under search engine fire numerous times. In fact, it is best to anticipate manual evaluations here even if you've dodged the algorithmic sweep.

The basic tenets are:

- Don't simply republish something that's found elsewhere on the Web unless your site adds substantive value to users, and don't infringe on others' copyrights or trademarks.
- If you're hosting affiliate content, expect to be judged more harshly than others, as affiliates in the SERPs are one of users' top complaints about search engines.
- Small changes such as a few comments, a clever sorting algorithm or automated tags, filtering, a line or two of text, simple mashups, or advertising do *not* constitute "substantive value."

For some exemplary cases where websites fulfill these guidelines, check out the way sites such as **CNET**, **Urbanspoon**, and **Metacritic** take content/products/reviews from elsewhere, both aggregating *and* adding value for their users.

Last but not least, Google has provided a guideline to refrain from trying to place "search results in the search results." For reference, look at the post from Google's Matt Cutts, including the comments, at <http://www.matcutts.com/blog/search-results-in-search-results/>. Google's stated position is that search results generally don't add value for users, though others have made the argument that this is merely an anticompetitive move.

Sites can benefit from having their search results transformed into more valuable listings and category/subcategory landing pages. Sites that have done this have had great success recovering rankings and gaining traffic from Google.

In essence, you want to avoid the potential for your site pages being perceived, both by an engine's algorithm and by human engineers and quality raters, as search results. Refrain from:

- Pages labeled in the title or headline as "search results" or "results"
- Pages that appear to offer a query-based list of links to "relevant" pages on the site without other content (add a short paragraph of text, an image, and formatting that make the "results" look like detailed descriptions/links instead)
- Pages whose URLs appear to carry search queries (e.g., `?q=miami+restaurants` or `?search=Miami+restaurants` versus `/miami-restaurants`)

- Pages with text such as “Results 1 through 10”

Though it seems strange, these subtle, largely cosmetic changes can mean the difference between inclusion and removal. Err on the side of caution and dodge the appearance of search results.

Content Themes

A less discussed but also important issue is the fit of each piece of content to your site. If you create an article about pizza, but the rest of your site is about horseshoes, your article is unlikely to rank for the term *pizza*. Search engines analyze and understand what sites, or sections of sites, focus on.

You can think of this as being the “theme” of the site (or section). If you start creating content that is not on the same theme, that content will have a very difficult time ranking. Further, your off-topic content could potentially weaken the theme of the rest of the site.

One site can support multiple themes, but each themed section needs to justify its own existence by following good SEO practices, including getting third parties to implement links from the pages of their sites to that section. Make sure you keep your content on topic, and this will help the SEO for all of the pages of your site.

Copyblogger has created a tool to help measure the fit of a given article to your site, known as **Scribe**. Not only will Scribe measure the fit of an article to your site, it will also offer a more general look at the consistency of the content across your site overall.

Duplicate Content Issues

Duplicate content generally falls into three categories: exact (or true) duplicates, whereby two URLs output identical content; near duplicates, whereby there are small content differentiators (sentence order, image variables, etc.); and cross-domain duplicates, whereby exact or near duplication exists on multiple domains.

There are two related concepts that are not treated by Google the same way as duplicate content, but are often confused by publishers and inexperienced SEO practitioners. These are:

Thin content

As noted previously, these are pages that don’t have much content on them at all. An example might be a set of pages built out to list all the locations for a business with 5,000 locations, but the only content on all the pages is the address of each location.

Thin slicing

These are pages with very minor differences in focus. Consider a site that sells running shoes, and one of the shoes offered is men's Nike Air Max LTD running shoes, which comes in sizes 6, 6.5, 7, 7.5, 8,...15. If the site had a different page for each size of this shoe, even though each page would actually be showing a different product, there is just not much useful difference between the pages overall.

Google has been clear that it doesn't like thin content or thin slicing. Either can trigger Google's Panda algorithm, which is discussed more in [Chapter 9](#). Exactly how Bing differentiates duplicate content, thin content, and thin slicing is less clear, but it also prefers that publishers avoid creating these types of pages.

Duplicate content can result from many causes, including licensing of content to or from your site, site architecture flaws due to non-SEO-friendly content management systems, or plagiarism. Not too long ago, however, spammers in desperate need of content began the now much-reviled process of scraping content from legitimate sources, scrambling the words (through many complex processes), and repurposing the text to appear on their own pages in the hopes of attracting long-tail searches and serving contextual ads (and various other nefarious purposes).

Thus, today we're faced with a world of duplicate content issues and their corresponding penalties. Here are some definitions that are useful for this discussion:

Unique content

This is written by humans; is completely different from any other combination of letters, symbols, or words on the Web; and is clearly not manipulated through computer text-processing algorithms (such as Markov-chain-employing spam tools).

Snippets

These are small chunks of content, such as quotes, that are copied and reused; they are almost never problematic for search engines, especially when included in a larger document with plenty of unique content.

Shingles

Search engines look at relatively small phrase segments (e.g., five to six words) for the presence of the same segments on other pages on the Web. When there are too many shingles in common between two documents, the search engines may interpret them as duplicate content.

Duplicate content issues

This phrase is typically used to refer to duplicate content that is not in danger of getting a website penalized, but rather is simply a copy of an existing page that forces the search engines to choose which version to display in the index (a.k.a. duplicate content filter).

Duplicate content filter

This is when the search engine removes substantially similar content from a search result to provide a better overall user experience.

Duplicate content penalty

Penalties are applied rarely and only in egregious situations. Engines may devalue or ban other web pages on the site, too, or even the entire website.

Consequences of Duplicate Content

Assuming your duplicate content is a result of innocuous oversights on your developer's part, the search engine will most likely simply filter out all but one of the pages that are duplicates because it wants to display one version of a particular piece of content in a given SERP. In some cases, the search engine may filter out results prior to including them in the index, and in other cases it may allow a page in the index and filter it out when it is assembling the SERPs in response to a specific query. In the latter case, a page may be filtered out in response to some queries and not others.

Searchers want diversity in the results, not the same results repeated again and again. Search engines therefore try to filter out duplicate copies of content, and this has several consequences:

- A search engine bot comes to a site with a *crawl budget*, which is the number of pages it plans to crawl in each particular session. Each time it crawls a page that is a duplicate (which is simply going to be filtered out of search results) you have let the bot waste some of its crawl budget. That means fewer of your "good" pages will get crawled. This can result in fewer of your pages being included in the search engine index.
- Links to duplicate content pages represent a waste of link authority. Duplicated pages can gain PageRank, or link authority, and because it does not help them rank, that link authority is misspent.
- No search engine has offered a clear explanation for how its algorithm picks which version of a page it shows. In other words, if it discovers three copies of the same content, which two does it filter out? Which one does it still show? Does it vary based on the search query? The bottom line is that the search engine might not favor the version you want.

Although some SEO professionals may debate some of the preceding specifics, the general points will meet with near-universal agreement. However, there are a handful of caveats to take into account.

For one, on your site you may have a variety of product pages and also offer print versions of those pages. The search engine might pick just the printer-friendly page as the

one to show in its results. This does happen at times, and it can happen even if the printer-friendly page has lower link authority and will rank less well than the main product page.

The best potential fix for this is to apply the `rel="canonical"` link element to all versions of the page to indicate which version is the original.

A second version of this can occur when you syndicate content to third parties. The problem is that the search engine may filter your copy of the article out of the results in favor of the version in use by the person republishing your article. There are three potential solutions to this:

- Get the person publishing your syndicated content to publish a `rel="canonical"` link element tag back to the original page on your site. This will help indicate to the search engines that your copy of the page is the original, and any links pointing to the syndicated page will be credited to your original instead.
- Have the syndicating partner `noindex` its copy of the content. This will keep the duplicate copy out of the search engine index. In addition, any links in that content back to your site will still pass link authority to you.
- Have the partner implement a link back to the original source page on your site. Search engines usually interpret this correctly and emphasize your version of the content when you do that. Note, however, that there have been instances where Google attributes the originality of the content to the site republishing it, particularly if that site has vastly more authority and trust than the true original source of the content.

How Search Engines Identify Duplicate Content

Some examples will illustrate the process for Google as it finds duplicate content on the Web. In the examples shown in [Figure 6-28](#) through [Figure 6-31](#), three assumptions have been made:

- The page with text is assumed to be a page that contains duplicate content (not just a snippet, despite the illustration).
- Each page of duplicate content is presumed to be on a separate domain.
- The steps that follow have been simplified to make the process as easy and clear as possible. This is almost certainly not the exact way in which Google performs (but it conveys the effect).

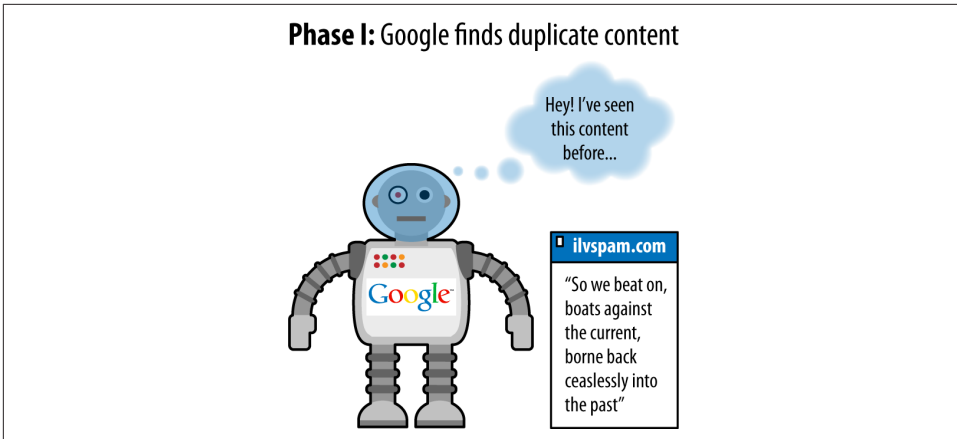


Figure 6-28. Google finding duplicate content

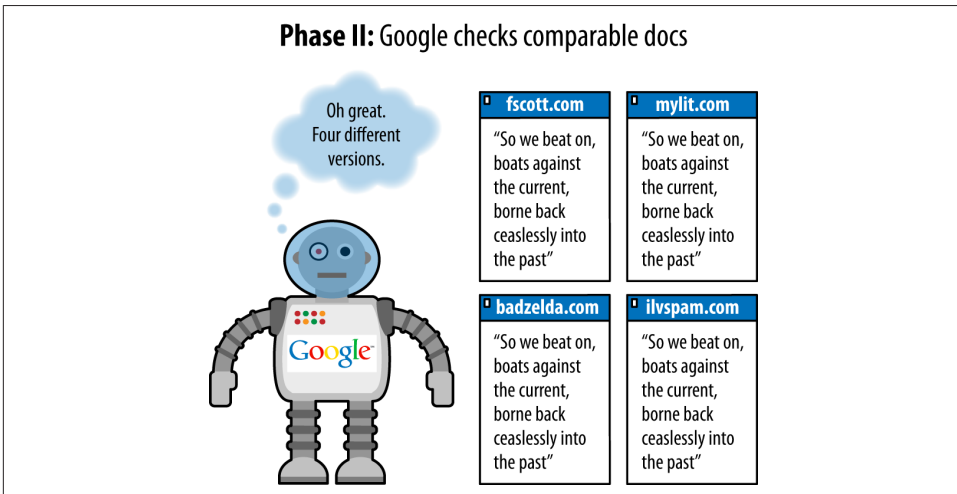


Figure 6-29. Google comparing the duplicate content to the other copies

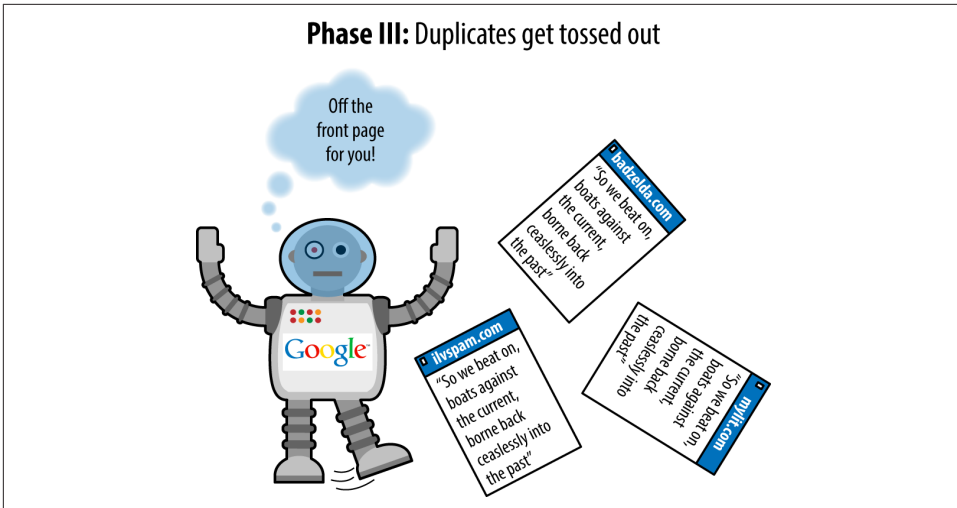


Figure 6-30. Duplicate copies getting filtered out

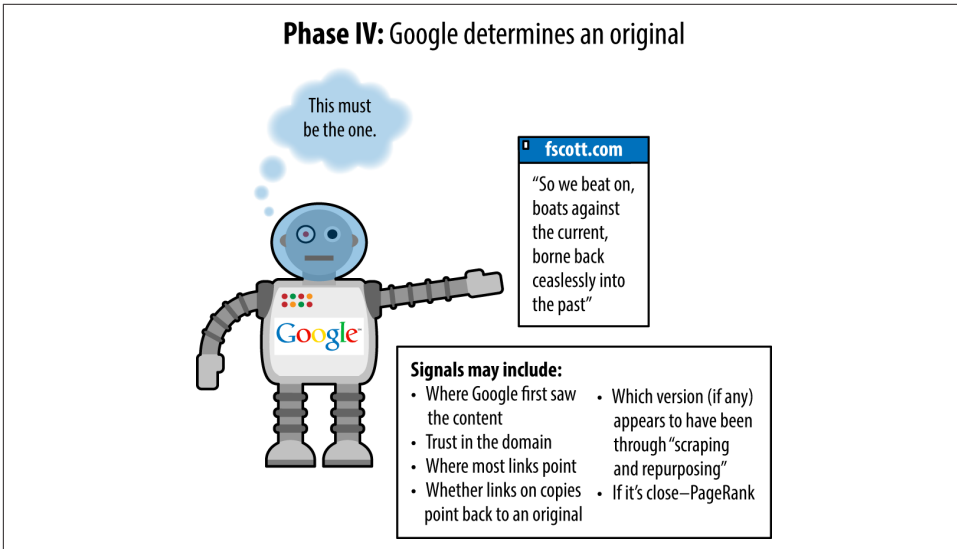


Figure 6-31. Google choosing one as the original

There are a few facts about duplicate content that bear mentioning, as they can trip up webmasters who are new to the duplicate content issue:

Location of the duplicate content

Is it duplicated content if it is all on my site? Yes, in fact, duplicate content can occur within a site or across different sites.

Percentage of duplicate content

What percentage of a page has to be duplicated before I run into duplicate content filtering? Unfortunately, the search engines would never reveal this information because it would compromise their ability to prevent the problem.

It is also a near certainty that the percentage at each engine fluctuates regularly and that more than one simple direct comparison goes into duplicate content detection. The bottom line is that pages do not need to be identical to be considered duplicates.

Ratio of code to text

What if my code is huge and there are very few unique HTML elements on the page? Will Google think the pages are all duplicates of one another? No. The search engines do not care about your code; they are interested in the content on your page. Code size becomes a problem only when it becomes extreme.

Ratio of navigation elements to unique content

Every page on my site has a huge navigation bar, lots of header and footer items, but only a little bit of content; will Google think these pages are duplicates? No. Google and Bing factor out the common page elements, such as navigation, before evaluating whether a page is a duplicate. They are very familiar with the layout of websites and recognize that permanent structures on all (or many) of a site's pages are quite normal. Instead, they'll pay attention to the "unique" portions of each page and often will largely ignore the rest. Note, however, that these will almost certainly be considered thin content by the engines.

Licensed content

What should I do if I want to avoid duplicate content problems, but I have licensed content from other web sources to show my visitors? Use `meta name = "robots" content="noindex, follow"`. Place this in your page's header and the search engines will know that the content isn't for them. This is a general best practice, because then humans can still visit and link to the page, and the links on the page will still carry value.

Another alternative is to make sure you have exclusive ownership and publication rights for that content.

Copyright Infringement

One of the best ways to monitor whether your site's copy is being duplicated elsewhere is to use [CopyScape](#), a site that enables you to instantly view pages on the Web that are using your content. Do not worry if the pages of these sites rank far behind your own pages for any relevant queries—if any large, authoritative, content-rich

domain tried to fight all the copies of its work on the Web, it would have at least two full-time jobs on its hands. Luckily, the search engines have placed trust in these types of sites to issue high-quality, relevant content, and therefore recognize them as the original issuer.

If, on the other hand, you have a relatively new site or a site with few inbound links, and the scrapers are consistently ranking ahead of you (or someone with a powerful site is stealing your work), you've got some recourse. One option is just to ask the publisher to remove the offending content. In some cases, the publisher is simply unaware that copying your content is not allowed. Another option is to contact the site's hosting company. Hosting companies could potentially be liable for hosting duplicate content, so they are frequently quick to react to such inquiries. Just be sure to provide as much documentation as possible to show that the content was originally yours.

Another option is to file a DMCA infringement request with Google, Yahoo!, and Bing (you should also file this request with the infringing site's hosting company).

A further option is to file a legal suit (or threaten such) against the website in question. You may want to try to start with a more informal communication asking the publisher to remove the content before you send a letter from the attorneys, as the DMCA motions can take up to several months to go into effect; but if the publisher is nonresponsive, there is no reason to delay taking stronger action, either. If the site republishing your work has an owner in your country, this latter course of action is probably the most effective first step.

A very effective and inexpensive option for this process is DMCA.com.

An actual penalty situation

The preceding examples show duplicate content filters and are not actual penalties, but, for all practical purposes, they have the same impact as a penalty: lower rankings for your pages. But there are scenarios where an actual penalty can occur.

For example, sites that aggregate content from across the Web can be at risk, particularly if little unique content is added from the site itself. In this type of scenario, you might see the site actually penalized.

If you find yourself in this situation, the only fixes are to reduce the number of duplicate pages accessible to the search engine crawler. You can accomplish this by deleting them, using `canonical` on the duplicates, `noindex`-ing the pages themselves, or adding a substantial amount of unique content.

One example of duplicate content that may get filtered out on a broad basis is a *thin affiliate* site. This nomenclature frequently describes a site promoting the sale of someone else's products (to earn a commission), yet provides little or no information differ-

entiated from other sites selling the product. Such a site may have received the descriptions from the manufacturer of the products and simply replicated those descriptions along with an affiliate link (so that it can earn credit when a click or purchase is performed).

The problem arises when a merchant has thousands of affiliates generally promoting websites using the same descriptive content, and search engineers have observed user data suggesting that, from a searcher's perspective, these sites add little value to their indexes. Thus, the search engines attempt to filter out this type of site, or even ban it from their index. Plenty of sites operate affiliate models but also provide rich new content, and these sites generally have no problem; it is when duplication of content and a lack of unique, value-adding material come together on a domain that the engines may take action.

How to Avoid Duplicate Content on Your Own Site

As we outlined, duplicate content can be created in many ways. Internal duplication of material requires specific tactics to achieve the best possible results from an SEO perspective. In many cases, the duplicate pages are pages that have no value to either users or search engines. If that is the case, try to eliminate the problem altogether by fixing the implementation so that all pages are referred to by only one URL. Also, 301-redirect (these are discussed in more detail in "Redirects") the old URLs to the surviving URLs to help the search engines discover what you have done as rapidly as possible, and preserve any link authority the removed pages may have had.

If that process proves to be impossible, there are many options, as we will outline in "Content Delivery and Search Spider Control" on page 334. Here is a summary of the guidelines on the simplest solutions for dealing with a variety of scenarios:

- Use *robots.txt* to block search engine spiders from crawling the duplicate versions of pages on your site.
- Use the `rel="canonical"` link element. This is the next best solution to eliminating the duplicate pages.
- Use `<meta name="robots" content="noindex">` to tell the search engine to not index the duplicate pages.

Be aware, however, that if you use *robots.txt* to prevent a page from being crawled, then using `noindex` or `nofollow` on the page itself does not make sense—the spider can't read the page, so it will never see the `noindex` or `nofollow`. With these tools in mind, here are some specific duplicate content scenarios:

HTTPS pages

If you make use of *SSL* (encrypted communications between the browser and the web server), and you have not converted your entire site, you will have some pages on your site that begin with *https:* instead of *http:*. The problem arises when the links on your *https:* pages link back to other pages on the site using relative instead of absolute links, so (for example) the link to your home page becomes *https://www.yourdomain.com* instead of *http://www.yourdomain.com*.

If you have this type of issue on your site, you may want to use the `rel="canonical"` link element, which we describe in [“Content Delivery and Search Spider Control” on page 334](#), or 301 redirects to resolve problems with these types of pages. An alternative solution is to change the links to absolute links (*http://www.yourdomain.com/content* instead of */content*), which also makes life more difficult for content thieves that scrape your site.

A CMS that creates duplicate content

Sometimes sites have many versions of identical pages because of limitations in the CMS where it addresses the same content with more than one URL. These are often unnecessary duplications with no end-user value, and the best practice is to figure out how to eliminate the duplicate pages and 301 the eliminated pages to the surviving pages. Failing that, fall back on the other options listed at the beginning of this section.

Print pages or multiple sort orders

Many sites offer print pages to provide the user with the same content in a more printer-friendly format. Or some ecommerce sites offer their products in multiple sort orders (such as size, color, brand, and price). These pages do have end-user value, but they do not have value to the search engine and will appear to be duplicate content. For that reason, use one of the options listed previously in this subsection, or set up a print CSS stylesheet such as the one outlined in [this post by Yoast](#).

Duplicate content in blogs and multiple archiving systems (e.g., pagination)

Blogs present some interesting duplicate content challenges. Blog posts can appear on many different pages, such as the home page of the blog, the permalink page for the post, date archive pages, and category pages. Each instance of the post represents duplicates of the other instances. Few publishers attempt to address the presence of the post on the home page of the blog and also at its permalink, and this is common enough that the search engines likely deal reasonably well with it. However, it may make sense to show only excerpts of the post on the category and/or date archive pages.

User-generated duplicate content (e.g., repostings)

Many sites implement structures for obtaining user-generated content, such as a blog, forum, or job board. This can be a great way to develop large quantities of content at a very low cost. The challenge is that users may choose to submit the same content on your site and in several other sites at the same time, resulting in duplicate content among those sites. It is hard to control this, but there are two things you can do to mitigate the problem:

- Have clear policies that notify users that the content they submit to your site must be unique and cannot be, or cannot have been, posted to other sites. This is difficult to enforce, of course, but it will still help some to communicate your expectations.
- Implement your forum in a different and unique way that demands different content. Instead of having only the standard fields for entering data, include fields that are likely to be unique over what other sites do, but that will still be interesting and valuable for site visitors to see.

Controlling Content with Cookies and Session IDs

Sometimes you want to more carefully dictate what a search engine robot sees when it visits your site. In general, search engine representatives refer to the practice of showing different content to users than to crawlers as *cloaking*, which violates the engines' Terms of Service (TOS) and is considered spam.

However, there are legitimate uses for this practice that are not deceptive to the search engines or malicious in intent. This section will explore methods for controlling content with cookies and sessions IDs.

What's a Cookie?

A *cookie* is a small text file that websites can leave on a visitor's hard disk, helping them to track that person over time. Cookies are the reason Amazon remembers your username between visits and the reason you don't necessarily need to log in to your Gmail account every time you open your browser. Cookie data typically contains a short set of information regarding when you last accessed a site, an ID number, and potentially, information about your visit (see [Figure 6-32](#)).

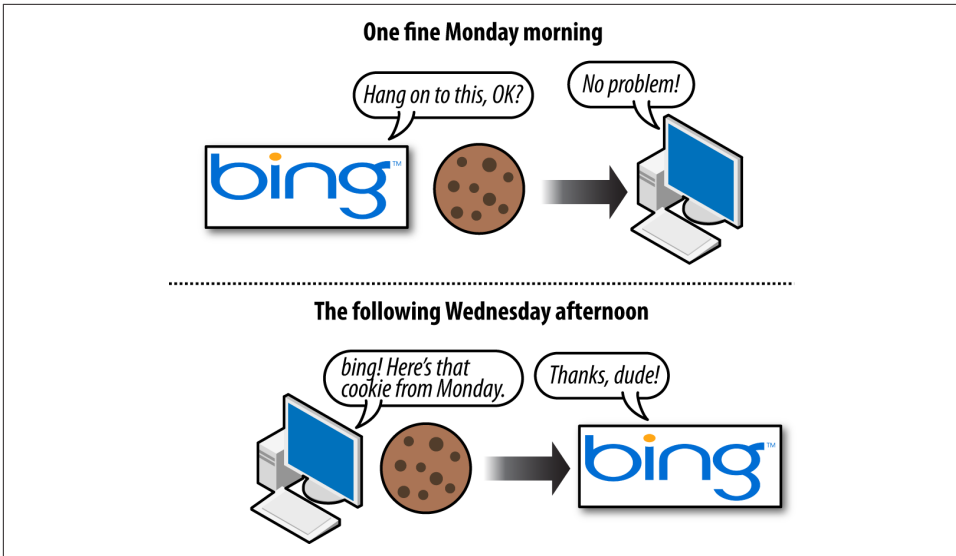


Figure 6-32. Using cookies to store data

Website developers use cookies for tracking purposes or to display different information to users based on their actions or preferences. Common uses include remembering a username, maintaining a shopping cart, and keeping track of previously viewed content. For example, if you've signed up for an account with Moz, it will provide you with options on your My Account page about how you want to view the blog and will remember those settings the next time you visit.

What Are Session IDs?

Session IDs are virtually identical to cookies in functionality, with one big difference. When you close your browser (or restart), session ID information is no longer stored on your hard drive (usually); see [Figure 6-33](#). The website you were interacting with may remember your data or actions, but it cannot retrieve session IDs from your machine that don't persist (and session IDs by default expire when the browser shuts down). In essence, session IDs are more like temporary cookies (although, as you'll see shortly, there are options to control this).



Figure 6-33. How session IDs are used

Although technically speaking session IDs are just a form of cookie without an expiration date, it is possible to set session IDs with expiration dates similar to cookies (going out decades). In this sense, they are virtually identical to cookies. Session IDs do come with an important caveat, though: they are frequently passed in the URL string, which can create serious problems for search engines (as every request produces a unique URL with duplicate content).

It is highly desirable to eliminate session IDs from your URLs, and you should avoid them if it is at all possible. If you currently have them, a short-term fix is to use the `rel="canonical"` link element (which we'll discuss in *"Content Delivery and Search Spider Control"* on page 334) to tell the search engines that you want them to ignore the session IDs.

NOTE

Any user has the ability to turn off cookies in his browser settings. This often makes web browsing considerably more difficult, and many sites will actually display a page saying that cookies are required to view or interact with their content. Cookies, persistent though they may be, are also deleted by users on a semiregular basis. For example, a [2011 comScore study](#) found that 33% of web users deleted their first-party cookies at least once per month.

How Do Search Engines Interpret Cookies and Session IDs?

Search engine spiders do not look at cookies or session IDs, and act as browsers with this functionality shut off. However, unlike visitors whose browsers won't accept cookies, the crawlers can sometimes reach sequestered content by virtue of webmasters who want to specifically let them through. Many sites have pages that require cookies or sessions to be enabled but have special rules for search engine bots, permitting them to access the content as well. Although this is technically cloaking, there is a form of this known as *First Click Free* that search engines generally allow (we will discuss this in more detail in [“Content Delivery and Search Spider Control” on page 334](#)).

Despite the occasional access engines are granted to cookie/session-restricted pages, the vast majority of cookie and session ID usage creates content, links, and pages that limit access. Web developers can leverage the power of options such as First Click Free to build more intelligent sites and pages that function in optimal ways for both humans and engines.

Why Would You Want to Use Cookies or Session IDs to Control Search Engine Access?

There are numerous potential tactics to leverage cookies and session IDs for search engine control. Here are many of the major strategies you can implement with these tools, but there are certainly limitless other possibilities:

Show multiple navigation paths while controlling the flow of link authority

Visitors to a website often have multiple ways in which they'd like to view or access content. Your site may benefit from offering many paths to reaching content (by date, topic, tag, relationship, ratings, etc.), but doing so expends PageRank or link authority that would be better optimized by focusing on a single, search engine–friendly navigational structure. This is important because these varied sort orders may be seen as duplicate content.

You can require a cookie for users to access the alternative sort order versions of a page, and prevent the search engine from indexing multiple pages with the same content. One alternative (but not foolproof) solution is to use the `rel="canonical"` link element to tell the search engine that these alternative sort orders are really

just the same content as the original page (we will discuss canonical in “Content Delivery and Search Spider Control” on page 334).

Keep limited pieces of a page’s content out of the engines’ indexes

Many pages may contain content that you’d like to show to search engines and other pieces you’d prefer appear only for human visitors. These could include ads, login-restricted information, links, or even rich media. Once again, showing non-cookied users the plain version and cookie-accepting visitors the extended information can be invaluable. Note that this option is often used in conjunction with a login, so only registered users can access the full content (such as on sites like Facebook and LinkedIn).

Grant access to pages requiring a login

As with snippets of content, there are often entire pages or sections of a site on which you’d like to restrict search engine access. This can be easy to accomplish with cookies/sessions, and it can even help to bring in search traffic that may convert to “registered user” status. For example, if you had desirable content that you wished to restrict, you could create a page with a short snippet and an offer for the visitor to continue reading upon registration, which would then allow him access to that work at the same URL. We will discuss this more in “Content Delivery and Search Spider Control” on page 334.

Avoid duplicate content issues

One of the most promising areas for cookie/session use is to prohibit spiders from reaching multiple versions of the same content, while allowing visitors to get the version they prefer. As an example, at Moz, logged-in users can see full blog entries on the blog home page, but search engines and nonregistered users will see only the excerpts. This prevents the content from being listed on multiple pages (the blog home page and the specific post pages) and provides a richer user experience for members.

Content Delivery and Search Spider Control

On occasion, it can be valuable to show search engines one version of content and show humans a different version. As we’ve discussed, this is technically called cloaking, and the search engines’ guidelines have near-universal policies restricting it. In practice, many websites, large and small, appear to use content delivery effectively and without being penalized by the search engines. However, use great care if you implement these techniques, and know the risks that you are taking.

Cloaking and Segmenting Content Delivery

Before we discuss the risks and potential benefits of cloaking-based practices, take a look at [Figure 6-34](#), which illustrates how cloaking works.

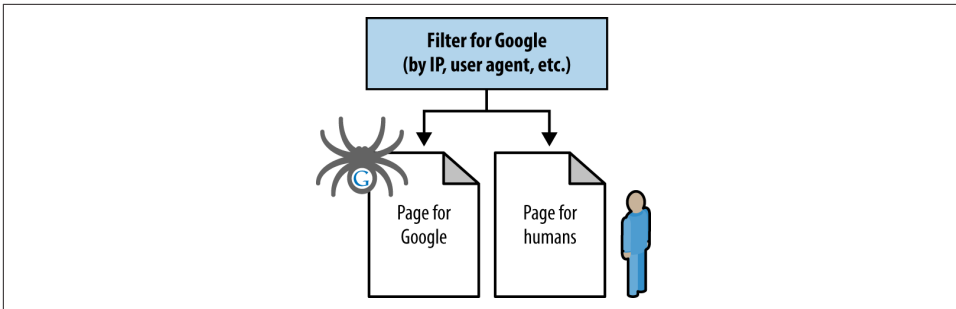


Figure 6-34. How cloaking works

Google’s Matt Cutts, former head of Google’s webspam team, has made strong public statements indicating that all forms of cloaking (with the only exception being First Click Free) are subject to penalty. This was largely backed by statements from Google’s John Mueller in a May 2009 interview.¹⁵ In August 2011, Matt Cutts later confirmed this again in a YouTube video, in which he asserted, “There is no such thing as white hat cloaking.”¹⁶

Google also makes its policy pretty clear in its guidelines on cloaking (<https://support.google.com/webmasters/answer/66355>):

Serving up different results based on user agent may cause your site to be perceived as deceptive and removed from the Google index.

There are two critical pieces in the preceding quote: *may* and *user agent*. It is true that if you cloak in the wrong ways, with the wrong intent, Google and the other search engines *may* remove you from their index, and if you do it egregiously, they certainly *will*.

A big factor is intent: if the engines feel you are attempting to manipulate their rankings or results through cloaking, they may take adverse action against your site. If, however, the intent of your content delivery doesn’t interfere with their goals, you’re less likely to be subject to a penalty, but there is never zero risk of a penalty. Google has taken a strong stand against all forms of cloaking regardless of intent.

What follows are some examples of websites that perform some level of cloaking:

Google

Search for *google toolbar* or *google translate* or *adwords* or any number of Google properties, and note how the URL you see in the search results and the one you

¹⁵ Caitlin O’Connell, “Google’s John Mueller Interviewed by Eric Enge,” Stone Temple Consulting, May 11, 2009, http://bit.ly/mueller_interview.

¹⁶ Available on the Google Webmasters YouTube channel: “Cloaking,” http://www.youtube.com/watch?feature=player_embedded&v=QHtnfOgp65Q.

land on almost never match. What's more, on many of these pages, whether you're logged in or not, you might see some content that is different from what's in the cache.

New York Times

The interstitial ads, the request to log in/create an account after five clicks, and the archive inclusion are all showing different content to engines versus humans.

Wine.com

In addition to some redirection based on your path, there's the state overlay forcing you to select a shipping location prior to seeing any prices (or any pages). That's a form the engines don't have to fill out.

Yelp

Geotargeting through cookies based on location is a very popular form of local targeting that hundreds, if not thousands, of sites use.

Trulia

Trulia was found to be doing some interesting redirects on partner pages and its own site (http://bit.ly/trulias_integrity).

The message should be clear: cloaking won't always get you banned, and you can do some pretty smart things with it. Again, the key to all of this is your intent. If you are doing it for reasons that are not deceptive and that provide a positive experience for users and search engines, you might not run into problems. However, there is no guarantee of this, so use these types of techniques with great care, and know that you may still get penalized for it.

Showing Different Content to Engines and Visitors

There are a few common causes for displaying content differently to different visitors, including search engines:

Multivariate and A/B split testing

Testing landing pages for conversions requires that you show different content to different visitors to test performance. In these cases, it is best to display the content using JavaScript/cookies/sessions and give the search engines a single, canonical version of the page that doesn't change with every new spidering (though this won't necessarily hurt you). Google previously offered software called Google Website Optimizer to perform this function, but it has been discontinued and replaced with [Google Analytics Content Experiments](#). If you have used Google Website Optimizer in the past, Google recommends removing the associated tags from your site pages.

Content requiring registration and First Click Free

If you force users to register (paid or free) in order to view specific content pieces, it is best to keep the URL the same for both logged-in and non-logged-in users and to show a snippet (one to two paragraphs is usually enough) to non-logged-in users and search engines. If you want to display the full content to search engines, you have the option to provide some rules for content delivery, such as showing the first one to two pages of content to a new visitor without requiring registration, and then requesting registration after that grace period. This keeps your intent more honest, and you can use cookies or sessions to restrict human visitors while showing the full pieces to the engines.

In this scenario, you might also opt to participate in Google's First Click Free program, wherein websites can expose "premium" or login-restricted content to Google's spiders, as long as users who click from the engine's results are given the ability to view that first article for free. Many prominent web publishers employ this tactic, including the popular site [Experts Exchange](#).

To be specific, to implement First Click Free, publishers must grant Googlebot (and presumably the other search engine spiders) access to all the content they want indexed, even if users normally have to log in to see the content. The user who visits the site will still need to log in, but the search engine spider will not have to do so. This will lead to the content showing up in the search engine results when applicable. However, if a user clicks on that search result, you must permit him to view the entire article (all pages of a given article if it is a multiple-page article). Once the user clicks to look at another article on your site, you can still require him to log in. Publishers can also limit the number of free accesses a user gets using this technique to five articles per day.

For more details, visit Google's First Click Free program pages at <http://googlewebmastercentral.blogspot.com/2008/10/first-click-free-for-web-search.html> and <http://googlewebmastercentral.blogspot.com/2009/12/changes-in-first-click-free.html>.

Navigation unspiderable by search engines

If your navigation is in Flash, JavaScript, a Java application, or another format where the search engine's ability to parse it is uncertain, you should consider showing search engines a version that has spiderable, crawlable content in HTML. Many sites do this simply with CSS layers, displaying a human-visible, search-invisible layer and a layer for the engines (and less capable browsers, such as mobile browsers). You can also employ the `<noscript>` tag for this purpose, although it is generally riskier, as many spammers have applied `<noscript>` as a way to hide content. Make sure the content shown in the search-visible layer is substantially the same as it is in the human-visible layer.

Duplicate content

If a significant portion of a page's content is duplicated, you might consider restricting spider access to it by placing it in an iframe that's restricted by *robots.txt*. This ensures that you can show the engines the unique portion of your pages, while protecting against duplicate content problems. We will discuss this in more detail in the next section.

Different content for different users

At times you might target content uniquely to users from different geographies (such as different product offerings that are more popular in their area), users with different screen resolutions (to make the content fit their screen size better), or users who entered your site from different navigation points. In these instances, it is best to have a "default" version of content that's shown to users who don't exhibit these traits to show to search engines as well.

Displaying Different Content to Search Engines Versus Visitors

There are a variety of strategies to segment content delivery. The most basic is to serve content that is not meant for the engines in unspiderable formats (e.g., placing text in images, Flash files, plug-ins, etc.). You should not use these formats for the purpose of cloaking; use them only if they bring a substantial end-user benefit (such as an improved user experience). In such cases, you may want to show the search engines the same content in a spiderable format. When you're trying to show the engines something you don't want visitors to see, you can use CSS formatting styles (preferably not `display:none`, as the engines have filters to watch specifically for this); JavaScript-, user agent-, cookie-, or session-based delivery; or IP delivery (showing content based on the visitor's IP address).

Be very wary when employing these strategies. As noted previously, the search engines expressly prohibit cloaking practices in their guidelines, and though there may be some leeway based on intent and user experience (e.g., your site is using cloaking to improve the quality of the user's experience, not to game the search engines), the engines take these tactics seriously and may penalize or ban sites that implement them inappropriately or with the intention of manipulation. In addition, even if your intent is good, the search engines may not see it that way and penalize you anyway.

Leveraging the robots.txt file

This file is located on the root level of your domain (e.g., <http://www.yourdomain.com/robots.txt>), and it is a highly versatile tool for controlling what the spiders are permitted to access on your site. You can use *robots.txt* to:

- Prevent crawlers from accessing nonpublic parts of your website
- Block search engines from accessing index scripts, utilities, or other types of code

- Avoid the indexation of duplicate content on a website, such as print versions of HTML pages, or various sort orders for product catalogs
- Autodiscover XML Sitemaps

The *robots.txt* file must reside in the root directory, and the filename must be entirely in lowercase (*robots.txt*, not *Robots.txt* or any other variation that includes uppercase letters). Any other name or location will not be seen as valid by the search engines. The file must also be entirely in text format (not in HTML format).

When you tell a search engine robot not to access a page, it prevents it from crawling the page. **Figure 6-35** illustrates what happens when the search engine robot sees a directive in *robots.txt* not to crawl a web page.

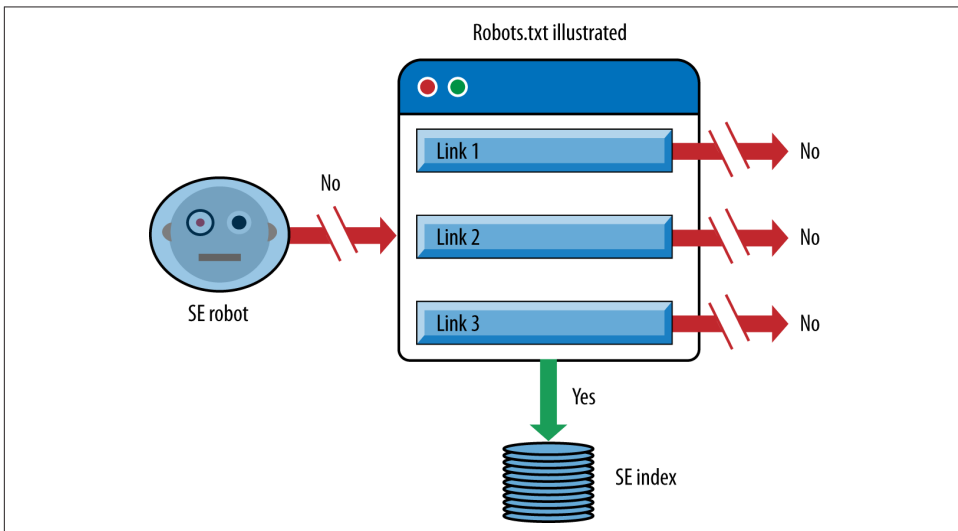


Figure 6-35. *Impact of robots.txt*

In essence, the page will not be crawled, so links on the page cannot pass link authority to other pages, because the search engine does not see the links. However, the page can be in the search engine index. This can happen if other pages on the Web link to it. Of course, the search engine will not have very much information on the page, as it cannot read it, and will rely mainly on the anchor text and other signals from the pages linking to it to determine what the page may be about. Any resulting search listings end up being pretty sparse when you see them in the Google index, as shown in **Figure 6-36**.

Figure 6-36 shows the results for the Google query *site:www.nytimes.com/cnet/*. This is not a normal query that a user would enter, but you can see what the results look like. Only the URL is listed, and there is no description. This is because the spiders aren't

permitted to read the page to get that data. In today's algorithms, these types of pages don't rank very high because their relevance scores tend to be quite low for any normal queries.

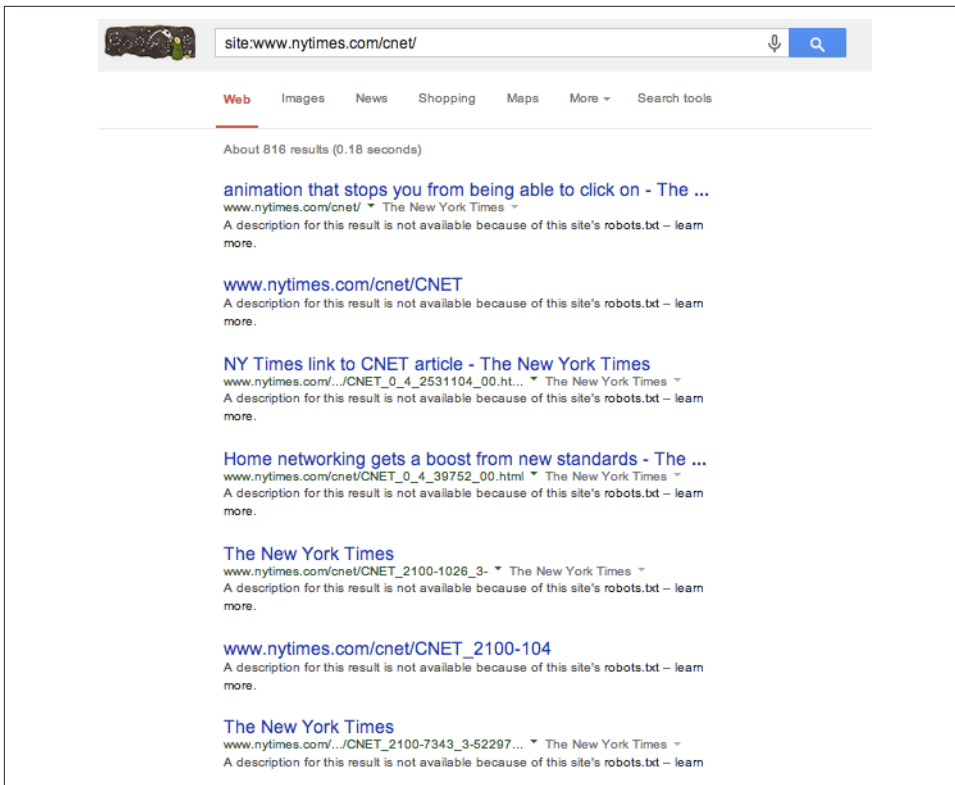


Figure 6-36. SERPs for pages that are listed in robots.txt

Google, Bing, and nearly all of the legitimate crawlers on the Web will follow the instructions you set out in the *robots.txt* file. Commands in *robots.txt* are primarily used to prevent spiders from accessing pages and subfolders on a site, though they have other options as well. Note that subdomains require their own *robots.txt* files, as do files that reside on an *https:* server.

Syntax of the robots.txt file. The basic syntax of *robots.txt* is fairly simple. You specify a robot name, such as “googlebot,” and then you specify an action. The robot is identified by user agent, and then the actions are specified on the lines that follow. The major action you can specify is `Disallow:`, which lets you indicate any pages you want to block the bots from accessing (you can use as many disallow lines as needed).

Some other restrictions apply:

- Each `User-agent/Disallow` group should be separated by a blank line; however, no blank lines should exist within a group (between the `User-agent` line and the last `Disallow`).
- The hash symbol (`#`) may be used for comments within a `robots.txt` file, where everything after `#` on that line will be ignored. This may be used either for whole lines or for the end of lines.
- Directories and filenames are case-sensitive: `private`, `Private`, and `PRIVATE` are all different to search engines.

Here is an example of a `robots.txt` file:

```
User-agent: Googlebot
Disallow:

User-agent: BingBot
Disallow: /

# Block all robots from tmp and logs directories
User-agent: *
Disallow: /tmp/
Disallow: /logs # for directories and files called logs
```

The preceding example will do the following:

- Allow “Googlebot” to go anywhere.
- Prevent “BingBot” from crawling any part of the site.
- Block all robots (other than Googlebot) from visiting the `/tmp/` directory or directories or files called `/logs` (e.g., `/logs` or `logs.php`).

Notice that the behavior of Googlebot is not affected by instructions such as `Disallow: /`. Because Googlebot has its own instructions from `robots.txt`, it will ignore directives labeled as being for all robots (i.e., those that use an asterisk).

One common problem that novice webmasters run into occurs when they have SSL installed so that their pages may be served via HTTP and HTTPS. A `robots.txt` file at `http://www.yourdomain.com/robots.txt` will not be interpreted by search engines as guiding their crawl behavior on `https://www.yourdomain.com`. To manage this, you need to create an additional `robots.txt` file at `https://www.yourdomain.com/robots.txt`. So, if you want to allow crawling of all pages served from your HTTP server and prevent crawling of all pages from your HTTPS server, you would need to implement the following:

For HTTP:

```
User-agent: *
Disallow:
```

For HTTPS:

```
User-agent: *  
Disallow: /
```

These are the most basic aspects of *robots.txt* files, but there are more advanced techniques as well. Some of these methods are supported by only some of the engines, as detailed here:

Crawl delay

Crawl delay is supported by Google, Bing, and Ask. It instructs a crawler to wait the specified number of seconds between crawling pages. The goal of the directive is to reduce the load on the publisher's server:

```
User-agent: BingBot  
Crawl-delay: 5
```

Pattern matching

Pattern matching appears to be usable by Google and Bing. The value of pattern matching is considerable. You can do some basic pattern matching using the asterisk wildcard character. Here is how you can use pattern matching to block access to all subdirectories that begin with *private* (*/private1/*, */private2/*, */private3/*, etc.):

```
User-agent: Googlebot  
Disallow: /private*/
```

You can match the end of the string using the dollar sign (\$). For example, to block URLs that end with *.asp*:

```
User-agent: Googlebot  
Disallow: /*.asp$
```

You may wish to prevent the robots from accessing any URLs that contain parameters. To block access to all URLs that include a question mark (?), simply use the question mark:

```
User-agent: *  
Disallow: /*?*
```

The pattern-matching capabilities of *robots.txt* are more limited than those of programming languages such as Perl, so the question mark does not have any special meaning and can be treated like any other character.

Allow

The **Allow** directive appears to be supported only by Google and Ask. It works the opposite of the **Disallow** directive and provides the ability to specifically call out directories or pages that may be crawled. When this is implemented, it can partially override a previous **Disallow** directive. This may be beneficial after large sections of the site have been disallowed, or if the entire site itself has been disallowed.

Here is an example that allows Googlebot into only the *google* directory:

```
User-agent: Googlebot
Disallow: /
Allow: /google/
```

Noindex

This directive works in the same way as the `meta robots noindex` command (which we will discuss shortly) and tells the search engines to explicitly exclude a page from the index. Because a `Disallow` directive prevents crawling but not indexing, this can be a very useful feature to ensure that the pages don't show in search results. Google supports this directive in *robots.txt*, and only unofficially.

Sitemaps

We discussed XML sitemaps at the beginning of this chapter. You can use *robots.txt* to provide an autodiscovery mechanism for the spider to find the XML sitemap file. The search engines can be told to find the file with one simple line in the *robots.txt* file:

```
Sitemap: sitemap_location
```

The *sitemap_location* should be the complete URL to the sitemap, such as `http://www.yourdomain.com/sitemap.xml`. You can place this anywhere in your file.

For full instructions on how to apply *robots.txt*, see [Martijn Koster's "A Standard for Robot Exclusion"](#). You can also test your *robots.txt* file in Google Search Console under Crawl -> robots.txt Tester.

You should use great care when making changes to *robots.txt*. A simple typing error can, for example, suddenly tell the search engines to no longer crawl any part of your site. After updating your *robots.txt* file, it is always a good idea to check it with the Google Search Console Test Robots.txt tool. You can find this by logging in to Search Console and then selecting Crawl -> Blocked URLs.

Using the `rel="nofollow"` attribute

In 2005, the three major search engines—Google, Microsoft, and Yahoo! (which still had its own search engine at that time)—all agreed to support an initiative intended to reduce the effectiveness of automated spam. Unlike the `meta robots nofollow` version of `nofollow`, the new directive could be employed as an attribute within an `<a>` or link tag to indicate that the linking site “does not editorially vouch for the quality of the linked-to page.” This enables a content creator to link to a web page without passing on any of the normal search engine benefits that typically accompany a link (trust, anchor text, PageRank, etc.).

Originally, the intent was to enable blogs, forums, and other sites where user-generated links were offered to shut down the value of spammers who built crawlers

that automatically created links. However, this has expanded as Google, in particular, recommends use of `nofollow` on links that are paid for—as the search engine’s preference is that only those links that are truly editorial and freely provided by publishers (without being compensated) should count toward bolstering a site’s/page’s rankings.

You can implement `nofollow` using the following format:

```
<a href="http://www.google.com" rel="nofollow">
```

Note that although you can use `nofollow` to restrict the passing of link value between web pages, the search engines still crawl through those links (despite the lack of semantic logic) and crawl the pages they link to. The search engines have provided contradictory input on this point. To summarize, `nofollow` does not expressly forbid indexing or spidering, so if you link to your own pages with it in an effort to keep them from being indexed or ranked, others may find them and link to them and your original goal will be thwarted.

Figure 6-37 shows how a search engine robot interprets a `nofollow` attribute when it finds one associated with a link (Link 1 in this example).

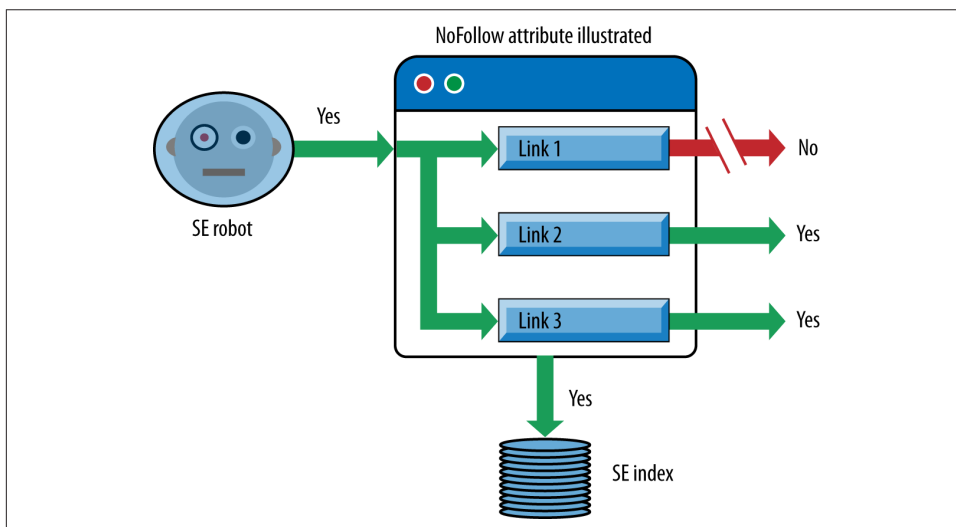


Figure 6-37. Impact of `nofollow` attribute

The specific link with the `nofollow` attribute was, for a number of years, considered to be disabled from passing link authority, and the notion of PageRank sculpting using `nofollow` was popular. The belief was that when you `nofollow` a particular link, the link authority that would have been passed to that link was preserved and the search engines would reallocate it to the other links found on the page. As a result, many publishers implemented `nofollow` links to lower value pages on their site (such as the About Us and Contact Us pages, or alternative sort order pages for product catalogs).

In fact, data from Moz's [Open Site Explorer tool](#), published in March 2009, showed that at that time about 3% of all links on the Web were `nofollow`, and that 60% of those `nofollow`s were applied to internal links.

In June 2009, however, Google's Matt Cutts wrote a post that made it clear that the link authority associated with that `nofollow` link is discarded rather than reallocated.¹⁷ In theory, you can still use `nofollow` however you want, but using it on internal links does not (at the time of this writing, according to Google) bring the type of benefit people have been looking for in the past. In fact, in certain scenarios it can be harmful.

In addition, many SEOs speculate that in some cases some value is indeed being placed on some `nofollow`ed links, and we suggest erring on the side of caution when using this attribute, as its use has been associated with a site being "flagged" as overoptimized or otherwise aggressive in SEO tactics.

This is a great illustration of the ever-changing nature of SEO. Something that was a popular, effective tactic is now being viewed as ineffective. Some more aggressive publishers will continue to pursue PageRank sculpting by using even more aggressive approaches, such as implementing links in encoded JavaScript or within iframes that have been disallowed in *robots.txt*, so that the search engines don't see them as links. Such aggressive tactics are probably not worth the trouble for most publishers.

Using the meta robots tag

The meta robots tag has three components: `cache`, `index`, and `follow`. The `cache` component instructs the engine about whether it can keep the page in the engine's public index, available via the "cached snapshot" link in the search results (see [Figure 6-38](#)).

¹⁷ Matt Cutts, "PageRank Sculpting," Matt Cutts: Gadgets, Google, and SEO, June 15, 2009, <https://www.mattcutts.com/blog/pagerank-sculpting/>.

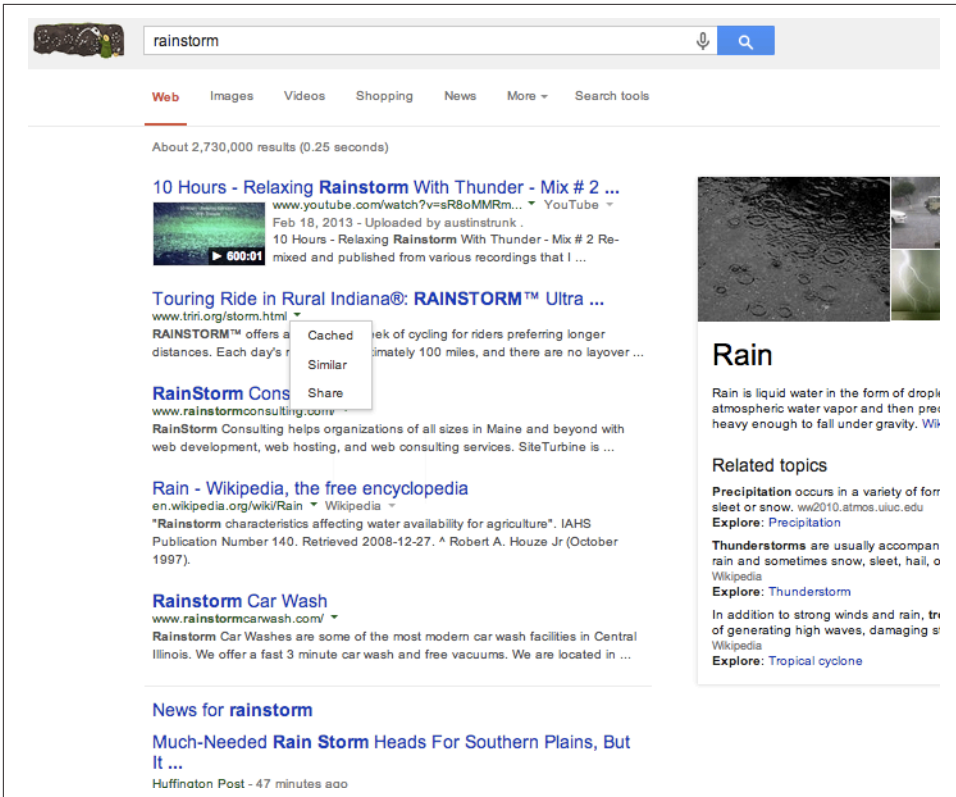


Figure 6-38. Accessing a cached page in the SERPs

The second, `index`, tells the engine that the page is allowed to be crawled and stored in any capacity. This is the default value, so it is unnecessary to place the `index` directive on each page. By contrast, a page marked `noindex` will be excluded entirely by the search engines. **Figure 6-39** shows what a search engine robot does when it sees a `noindex` tag on a web page.

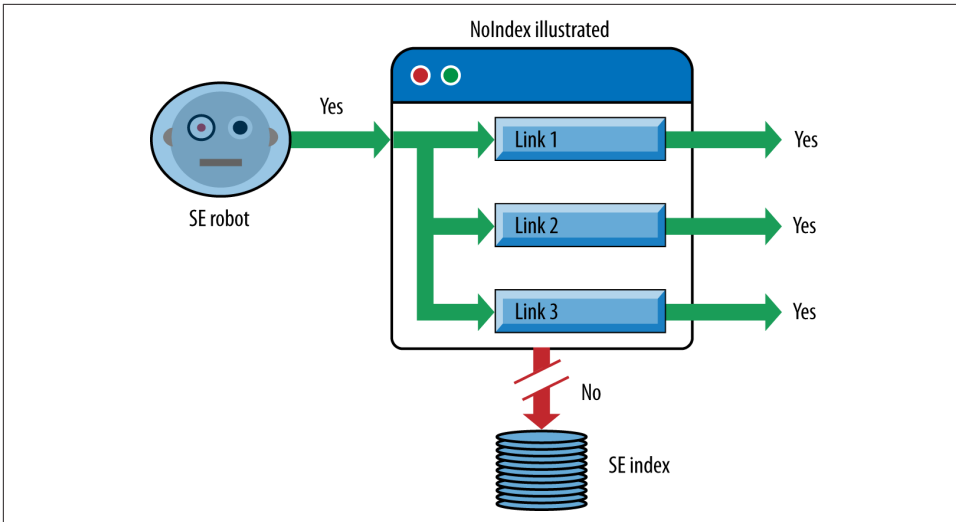


Figure 6-39. *Impact of noindex*

The page will still be crawled, and the page can still accumulate and pass link authority to other pages, but it will not appear in search indexes.

The final instruction available through the meta robots tag is `follow`. This command, like `index`, defaults to “yes, crawl the links on this page and pass link authority through them.” Applying `nofollow` tells the engine that none of the links on that page should pass link value. By and large, it is unwise to use this directive as a way to prevent links from being crawled. Human beings will still reach those pages and have the ability to link to them from other sites, so `nofollow` (in the meta robots tag) does little to restrict crawling or spider access. Its only function is to prevent link authority from spreading out, which has very limited application since the 2005 launch of the `rel="nofollow"` attribute (discussed earlier), which allows this directive to be placed on individual links.

Figure 6-40 outlines the behavior of a search engine robot when it finds a `nofollow` meta tag on a web page (assuming there are no other links pointing to the three linked URLs).

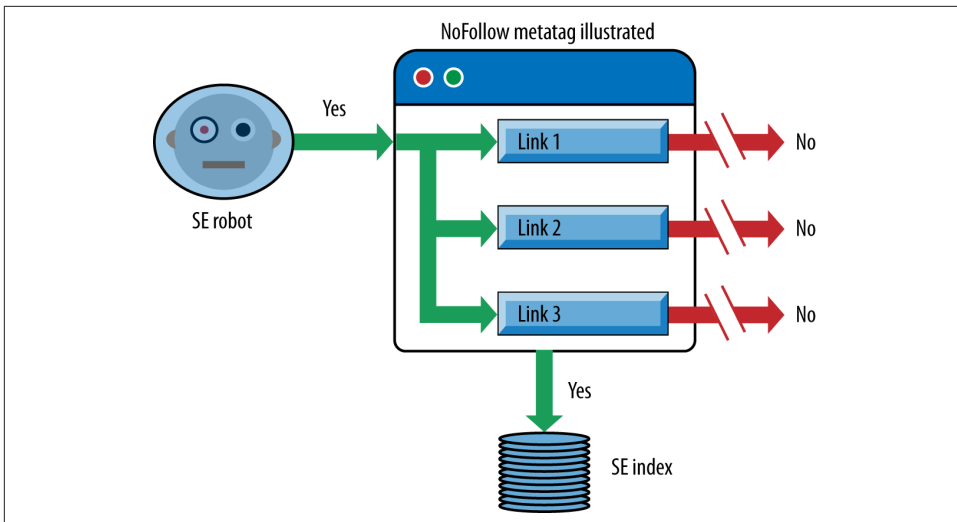


Figure 6-40. Impact of *nofollow* meta tag

When you use the `nofollow` meta tag on a page, the search engine will still crawl the page and place the page in its index. However, all links (both internal and external) on the page will be disabled from passing link authority to other pages.

One good application for `noindex` is to place this tag on HTML sitemap pages. These are pages designed as navigational aids for users and search engine spiders to enable them to efficiently find the content on your site. However, on some sites these pages are unlikely to rank for anything of importance in the search engines, yet you still want them to pass link authority to the pages they link to. Putting `noindex` on these pages keeps these HTML sitemaps out of the index and removes that problem. Make sure you *do not* apply the `nofollow` meta tag on the pages or the `nofollow` attribute on the links on the pages, as these will prevent the pages from passing link authority.

Using the `rel="canonical"` link element

In February 2009, Google, Yahoo!, and Microsoft debuted the `rel="canonical"` link element (sometimes referred to as the `canonical` tag). This element was a new construct designed explicitly for the purpose of identifying and dealing with duplicate content. Implementation is very simple and looks like this:

```
<link rel="canonical" href="http://moz.com/blog" />
```

This tag tells the search engines that the page in question should be treated as though it were a copy of the URL <http://moz.org/blog>, and that all of the link and content metrics the engines apply should technically flow back to that URL (see [Figure 6-41](#)).

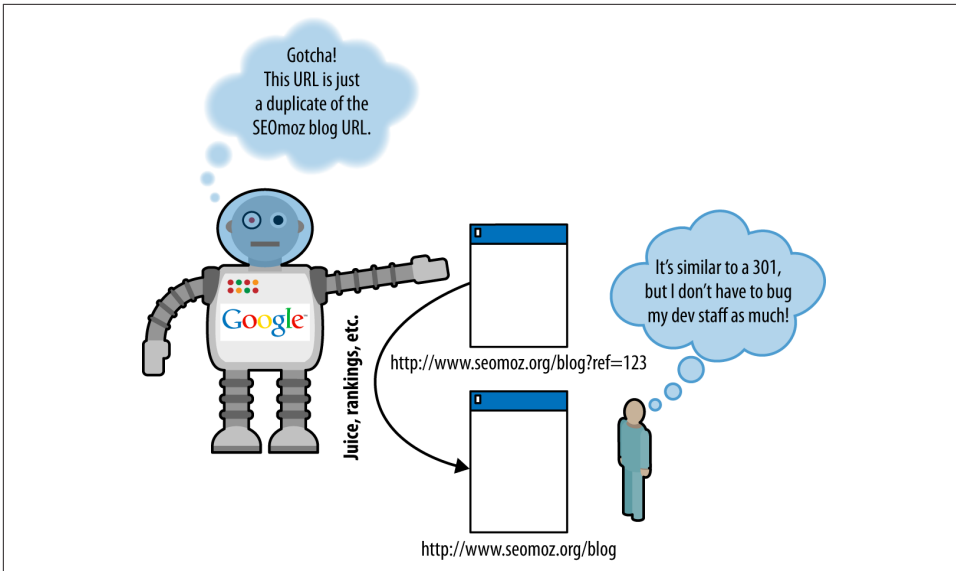


Figure 6-41. How search engines understand the canonical link element

The `rel="canonical"` link element is similar in many ways to a 301 redirect from an SEO perspective. In essence, you're telling the engines that multiple pages should be considered as one (which a 301 does), without actually redirecting visitors to the new URL (for many publishers this is less effort than some of the other solutions for their development staff). There are some differences, though:

- Whereas a 301 redirect points all traffic (bots and human visitors), `canonical` is just for engines, meaning you can still separately track visitors to the unique URL versions.
- A 301 is a much stronger signal that multiple pages have a single, canonical source. While 301s are considered a directive that search engines and browsers are obligated to honor, `canonical` is treated as a suggestion. Although the engines generally support this new tag and trust the intent of site owners, there will be limitations. Content analysis and other algorithmic metrics will be applied to ensure that a site owner hasn't mistakenly or manipulatively applied `canonical`, and you can certainly expect to see mistaken use of it, resulting in the engines maintaining those separate URLs in their indexes (meaning site owners would experience the same problems noted in "Duplicate Content Issues" on page 320).

We will discuss some applications for this tag later in this chapter. In general practice, the best solution is to resolve the duplicate content problems at their core, and eliminate them if you can. This is because the `rel="canonical"` link element is not guaran-

teed to work. However, it is not always possible to resolve the issues by other means, and `canonical` provides a very effective backup plan.

You can also include `canonical` directly within the HTTP response header for your page. The code might look something like the following:

```
HTTP/1.1 200 OK
Content-Type: application/pdf
Link: <http://www.example.com/white-paper.html>; rel="canonical"
Content-Length: 785710
(... rest of HTTP response headers...)
```

You can read more about this here: http://bit.ly/canonical_headers.

Blocking and cloaking by IP address range

You can customize entire IP addresses or ranges to block particular bots through server-side restrictions on IPs. Most of the major engines crawl from a limited number of IP ranges, making it possible to identify them and restrict access. This technique is, ironically, popular with webmasters who mistakenly assume that search engine spiders are spammers attempting to steal their content, and thus block the IP ranges to restrict access and save bandwidth. Use caution when blocking bots, and make sure you're not restricting access to a spider that could bring benefits, either from search traffic or from link attribution.

Blocking and cloaking by user agent

At the server level, it is possible to detect user agents and restrict their access to pages or websites based on their declaration of identity. As an example, if a website detected a rogue bot, you might double-check its identity before allowing access. The search engines all use a similar protocol to verify their user agents via the Web: a reverse DNS lookup followed by a corresponding forward DNS IP lookup. An example for Google would look like this:

```
> host 66.249.66.1
1.66.249.66.in-addr.arpa domain name pointer crawl-66-249-66-1.googlebot.com.

> host crawl-66-249-66-1.googlebot.com
crawl-66-249-66-1.googlebot.com has address 66.249.66.1
```

A reverse DNS lookup by itself may be insufficient, because a spoofer could set up reverse DNS to point to *xyz.googlebot.com* or any other address.

Using iframes

Sometimes there's a certain piece of content on a web page (or a persistent piece of content throughout a site) that you'd prefer search engines didn't see. As we discussed

earlier in this chapter, clever use of iframes can come in handy for this situation, as [Figure 6-42](#) illustrates.

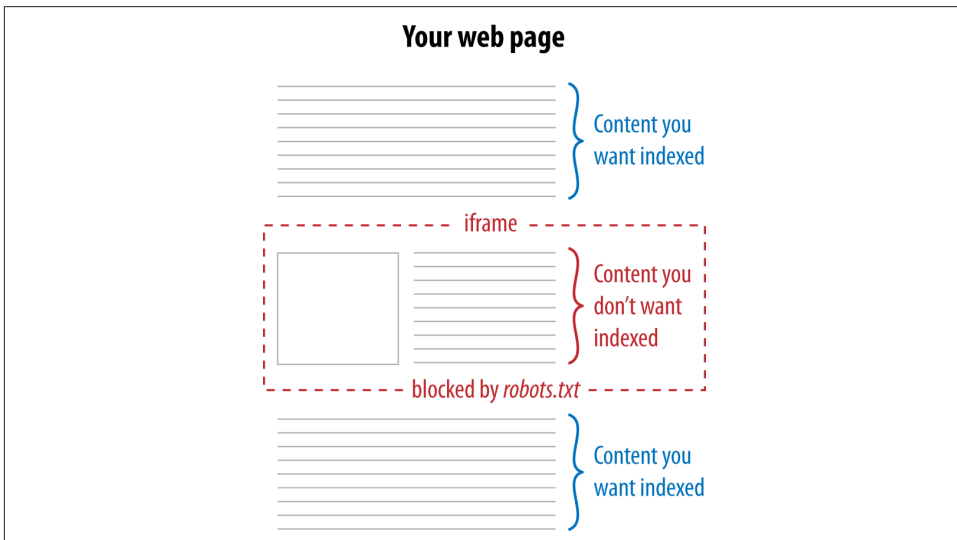


Figure 6-42. Using iframes to prevent indexing of content

The concept is simple: by using iframes, you can embed content from another URL onto any page of your choosing. By then blocking spider access to the iframe with *robots.txt*, you ensure that the search engines won't "see" this content on your page. Websites may do this for many reasons, including avoiding duplicate content problems, reducing the page size for search engines, or lowering the number of crawlable links on a page (to help control the flow of link authority).

Hiding text in images

As discussed earlier, the major search engines still have very little capacity to read text in images (and the processing power required makes for a severe barrier). Hiding content inside images isn't generally advisable, as it can be impractical for alternative devices (mobile, in particular) and inaccessible to others (such as screen readers).

Hiding text in Java applets

As with text in images, the content inside Java applets is not easily parsed by search engines, though using them as a tool to hide text would certainly be a strange choice.

Forcing form submission

Search engines will not submit HTML forms in an attempt to access the information retrieved from a search or submission. Thus, if you keep content behind a forced-form

submission and never link to it externally, your content will remain out of the engines (as [Figure 6-43](#) demonstrates).



Figure 6-43. *Using forms, which are generally not navigable by crawlers*

The problem arises when content behind forms earns links outside your control, such as when bloggers, journalists, or researchers decide to link to the pages in your archives without your knowledge. Thus, although form submission may keep the engines at bay, make sure that anything truly sensitive has additional protection (e.g., through *robots.txt* or meta robots).

Using login/password protection

Password protection and/or paywalls of any kind will effectively prevent any search engines from accessing content, as will any form of human-verification requirements, such as CAPTCHAs (the boxes requiring users to copy letter/number combinations to gain access to content). The major engines won't try to guess passwords or bypass these systems.

Removing URLs from a search engine's index

A secondary, post-indexing tactic, URL removal from most of the major search engines is possible through verification of your site and the use of the engines' tools. For example, [Google allows you to remove URLs through Search Console](#). Bing also allows you to remove URLs from its index, [via Bing Webmaster Tools](#).

Redirects

A redirect is used to indicate when content has moved from one location to another. For example, you may have some content at *http://www.yourdomain.com/old* and decide to restructure your site. As a result of this move, your content may move to *http://www.yourdomain.com/critical-keyword*.

Once a redirect is implemented, users who go to the old versions of your pages (perhaps via a bookmark they kept for the page) will be sent to the new versions. Without the redirect, the user would get a Page Not Found (404) error. With the redirect, the web server tells the incoming user agent (whether a browser or a spider) to instead fetch the requested content from the new URL.

Why and When to Redirect

Redirects are also important for letting search engines know when you have moved content. After you move content, the search engines will continue to have the old URL in their index and return it in their search results until they discover the page is no longer there and swap in the new page. You can help speed up this process by implementing a redirect. Here are some scenarios in which you may need to implement redirects:

- You have old content that expires, so you remove it.
- You find that you have broken URLs that have links and traffic.
- You change your hosting company.
- You change your CMS.
- You want to implement a canonical redirect (redirect all pages on *http://yourdomain.com* to *http://www.yourdomain.com*).
- You change the URLs where your existing content can be found, for any reason.

Not all of these scenarios require a redirect. For example, you can change hosting companies without impacting any of the URLs used to find content on your site, in which case no redirect is required. However, for any scenario in which any of your URLs change, you need to implement redirects.

Good and Bad Redirects

There are many ways to perform a redirect, but not all are created equal. The basic reason for this is that there are two major types of redirects that can be implemented, tied specifically to the HTTP status code returned by the web server to the browser. These are:

“301 moved permanently”

This status code tells the browser (or search engine crawler) that the resource has been permanently moved to another location, and there is no intent to ever bring it back.

“302 moved temporarily”

This status code tells the browser (or search engine crawler) that the resource has been temporarily moved to another location, and that the move should not be treated as permanent.

Both forms of redirect send a human or a search engine crawler to the new location, but the search engines interpret these two HTTP status codes in very different ways. When a crawler sees a 301 HTTP status code, it assumes it should pass the historical link authority (and any other metrics) from the old page to the new one. When a search engine crawler sees a 302 HTTP status code, it assumes it should not pass the historical link authority from the old page to the new one. In addition, the 301 redirect will lead the search engine to remove the old page from the index and replace it with the new one.

The preservation of historical link authority is very critical in the world of SEO. For example, imagine you had 1,000 links to *http://www.yourolddomain.com* and you decided to relocate everything to *http://www.yournewdomain.com*. If you used redirects that returned a 302 status code, you would be starting your link-building efforts from scratch again. In addition, the old version of the page may remain in the index and compete for search rankings in the search engines.

Note that there also can be redirects that pass no status code, or the wrong status code, such as a 404 error (Page Not Found) or a 200 OK (Page Loaded Successfully). These are also problematic, and should be avoided. There are other types of redirects as well, such as those that return a 303 or 307 status code. These also should be avoided, as the search engine’s response to them is at best unpredictable.

You want to definitively return a 301 HTTP status code for a redirect whenever you permanently move a page’s location.

Methods for URL Redirecting and Rewriting

As we just mentioned, there are many possible ways to implement redirects. On Apache web servers (normally present on machines running Unix or Linux as the operating system), **it is possible to implement redirects quite simply in a standard file called *.htaccess*** using the `Redirect` and `RedirectMatch` directives. You can also employ more advanced directives known as *rewrite rules* using the Apache module known as `mod_rewrite`, which we will discuss in a moment.

On web servers running **Microsoft IIS**, different methods are provided for implementing redirects. As described in [“IIS Redirects - 301 , 302”](#), the basic method is through the IIS console. People with IIS servers can also make use of a text file with directives, provided they use an ISAPI plug-in such as **ISAPI_Rewrite**, and this scripting language offers capabilities similar to Apache’s `mod_rewrite` module.

Many programmers use other techniques for implementing redirects, such as directly in programming languages like Perl, PHP, ASP, and JavaScript. If implementing redirects in this fashion, the programmer must make sure the HTTP status code returned by the web server is a 301. You can check the returned header with [the Firefox plug-in Live HTTP Headers](#), with [a Chrome extension](#), or with [a web-based server header checker](#).

Another method that you can use to implement a redirect occurs at the page level, via the meta refresh tag, which looks something like this:

```
<meta http-equiv="refresh"
      content="5;url=http://www.yourdomain.com/newlocation.htm" />
```

The first parameter in the `content` section, 5, indicates the number of seconds the web server should wait before redirecting the user to the indicated page. A publisher might use this to display a page letting users know that they’re going to get redirected to a different page than the one they requested.

The problem is that most meta refreshes are treated as though they are a 302 redirect. The sole exception to this is if you specify a redirect delay of 0 seconds. You will have to give up your helpful page telling users that you are redirecting them, but the search engines treat this as though it were a 301 redirect (to be safe, the best practice is simply to use a 301 redirect if at all possible).

mod_rewrite and ISAPI_Rewrite for URL rewriting and redirecting

There is much more to discuss on this topic than we can reasonably address in this book. The following description is intended only as an introduction to help orient more technical readers, including web developers and site webmasters, on how rewrites and redirects function. To skip this technical discussion, proceed to [“How to Redirect a Home Page Index File Without Looping” on page 360](#).

`mod_rewrite` for Apache and `ISAPI_Rewrite` for Microsoft IIS Server offer very powerful ways to rewrite your URLs. Here are some reasons for using these tools:

- You have changed your URL structure on your site so that content has moved from one location to another. This can happen when you change your CMS, or change your site organization for any reason.
- You want to map your search engine-unfriendly URLs into friendlier ones.

If you are running Apache as your web server, you would place directives known as *rewrite rules* within your *.htaccess* file or your Apache configuration file (e.g., *httpd.conf* or the site-specific config file in the *sites_conf* directory). Similarly, if you are running IIS Server, you'd use an ISAPI plug-in such as ISAPI_Rewrite and place rules in an *httpd.ini* config file.

Note that rules can differ slightly on ISAPI_Rewrite compared to *mod_rewrite*, and the following discussion focuses on *mod_rewrite*. Your *.htaccess* file would start with:

```
RewriteEngine on
RewriteBase /
```

You should omit the second line if you're adding the rewrites to your server config file, as *RewriteBase* is supported only in *.htaccess*. We're using *RewriteBase* here so that you won't have to type *^/* at the beginning of all the rules, just *^* (we will discuss regular expressions in a moment).

After this step, the rewrite rules are implemented. Perhaps you want to have requests for product page URLs of the format *http://www.yourdomain.com/products/123* to display the content found at *http://www.yourdomain.com/get_product.php?id=123*, without the URL changing in the location bar of the user's browser and without you having to recode the *get_product.php* script. Of course, this doesn't replace all occurrences of dynamic URLs within the links contained on all the site pages; that's a separate issue. You can accomplish this first part with a single rewrite rule, like so:

```
RewriteRule ^products/([0-9]+)/?$ /get_product.php?id=$1 [L]
```

This example tells the web server that all requests that come into the */product/* directory should be mapped into requests to */get_product.php*, while using the subfolder to */product/* as a parameter for the PHP script.

The *^* signifies the start of the URL following the domain, *\$* signifies the end of the URL, *[0-9]* signifies a numerical digit, and the *+* immediately following it means one or more occurrences of a digit. Similarly, the *?* immediately following the */* means zero or one occurrences of a slash character. The *()* puts whatever is wrapped within it into memory. You can then use *\$1* to access what's been stored in memory (i.e., whatever's within the first set of parentheses). Not surprisingly, if you included a second set of parentheses in the rule, you'd access that with *\$2*, and so on. The *[L]* flag saves on server processing by telling the rewrite engine to stop if it matches on that rule. Otherwise, all the remaining rules will be run as well.

Here's a slightly more complex example, where URLs of the format *http://www.yourdomain.com/webapp/wcs/stores/servlet/ProductDisplay?storeId=10001&catalogId=10001&langId=-1&categoryId=4&productId=123* would be rewritten to *http://www.yourdomain.com/4/123.htm*:

```
RewriteRule ^([^\s]+)/([^\s]+)\.htm$  
/webapp/wcs/stores/servlet/ProductDisplay?storeId=10001&catalogId=10001&  
langId=-1&categoryID=$1&productID=$2 [QSA,L]
```

The `[^/]` signifies any character other than a slash. That's because, within square brackets, `^` is interpreted as *not*. The `[QSA]` flag is for when you don't want the query string dropped (like when you want a tracking parameter preserved).

To write good rewrite rules you will need to become a master of *pattern matching* (which is simply another way to describe the use of regular expressions). Here are some of the most important special characters and how the rewrite engine interprets them:

- *
Zero or more of the immediately preceding character.
- +
One or more of the immediately preceding character.
- ?
Zero or one occurrences of the immediately preceding character.
- ^
The beginning of the string.
- \$
The end of the string.
- .
Any character (i.e., it acts as a wildcard).
- \
"Escapes" the character that follows; for example, `\.` means the dot is not meant to be a wildcard, but an actual character.
- ^
Inside brackets `[]` means *not*; for example, `[^/]` means *not slash*.

It is incredibly easy to make errors in regular expressions. Some of the common gotchas that lead to unintentional substring matches include:

- Using `.*` when you should be using `.+` (because `.*` can match on nothing).
- Not "escaping" with a backslash a special character that you don't want interpreted, as when you specify `.` instead of `\.` and you really meant the dot character rather than any character (thus, `default.htm` would match on `defaultthtm`, and `default\.` would match only on `default.htm`).

- Omitting `^` or `$` on the assumption that the start or end is implied (thus, `default\.htm` would match on `mydefault.html`, whereas `^default\.htm$` would match only on `default.htm`).
- Using “greedy” expressions that will match on all occurrences rather than stopping at the first occurrence.

The easiest way to illustrate what we mean by “greedy” is to provide an example:

```
RewriteRule ^(.*)/?index\.html$ /$1/ [L,R=301]
```

This will redirect requests for `http://www.yourdomain.com/blah/index.html` to `http://www.yourdomain.com/blah/`. This is probably not what was intended. Why did this happen? Because `*` will capture the slash character within it before the `/?` gets to see it. Thankfully, there’s an easy fix. Simply use `[^` or `.*?` instead of `*` to do your matching. For example, use `^(.*?)/?` instead of `^(.*)/?`, or `[^/]+/[^/]` instead of `.*/*.*`.

So, to correct the preceding rule, you could use the following:

```
RewriteRule ^(.*)/?index\.html$ /$1/ [L,R=301]
```

Why wouldn’t you use the following?

```
RewriteRule ^([^/]*)/?index\.html$ /$1/ [L,R=301]
```

This is more limited because it will match only on URLs with one directory. URLs containing multiple subdirectories, such as `http://www.yourdomain.com/store/cheese/swiss/wheel/index.html`, would not match.

As you might imagine, testing/debugging is a big part of URL rewriting. When you are debugging, the `RewriteLog` and `RewriteLogLevel` directives are your friends! Set the `RewriteLogLevel` to 4 or more to start seeing what the rewrite engine is up to when it interprets your rules.

By the way, the `[R=301]` flag in the last few examples—as you might guess—tells the rewrite engine to do a 301 redirect instead of a standard rewrite.

There’s another handy directive to use in conjunction with `RewriteRule`, called `RewriteCond`. You would use `RewriteCond` if you were trying to match on something in the query string, the domain name, or other elements not present between the domain name and the question mark in the URL (which is what `RewriteRule` looks at).

Note that neither `RewriteRule` nor `RewriteCond` can access what is in the anchor part of a URL—that is, whatever follows a `#`—because that is used internally by the browser and is not sent to the server as part of the request. The following `RewriteCond` example looks for a positive match on the hostname before it will allow the rewrite rule that follows to be executed:

```
RewriteCond %{HTTP_HOST} !^www\.yourdomain\.com$ [NC]
RewriteRule ^(.*)$ http://www.yourdomain.com/$1 [L,R=301]
```

Note the exclamation point at the beginning of the regular expression. The rewrite engine interprets that as *not*.

For any hostname other than *http://www.yourdomain.com*, a 301 redirect is issued to the equivalent canonical URL on the *www* subdomain. The [NC] flag makes the rewrite condition case-insensitive. Where is the [QSA] flag so that the query string is preserved, you might ask? It is not needed for redirecting; it is implied.

If you don't want a query string retained on a rewrite rule with a redirect, put a question mark at the end of the destination URL in the rule, like so:

```
RewriteCond %{HTTP_HOST} !^www\.yourdomain\.com$ [NC]
RewriteRule ^(.*)$ http://www.yourdomain.com/$1? [L,R=301]
```

Why not use *^yourdomain\.com\$* instead? Consider:

```
RewriteCond %{HTTP_HOST} ^yourdomain\.com$ [NC]
RewriteRule ^(.*)$ http://www.yourdomain.com/$1? [L,R=301]
```

That would not have matched on typo domains, such as *yourdoamin.com*, that the DNS server and virtual host would be set to respond to (assuming that misspelling was a domain you registered and owned).

Under what circumstances might you want to omit the query string from the redirected URL, as we did in the preceding two examples? When a session ID or a tracking parameter (such as *source=banner_ad1*) needs to be dropped. Retaining a tracking parameter after the redirect is not only unnecessary (because the original URL with the source code appended would have been recorded in your access logfiles as it was being accessed); it is also undesirable from a canonicalization standpoint. What if you wanted to drop the tracking parameter from the redirected URL, but retain the other parameters in the query string? Here's how you'd do it for static URLs:

```
RewriteCond %{QUERY_STRING} ^source=[a-z0-9]*$
RewriteRule ^(.*)$ /$1? [L,R=301]
```

And for dynamic URLs:

```
RewriteCond %{QUERY_STRING} ^(.*)&source=[a-z0-9]+(&?.*)$
RewriteRule ^(.*)$ /$1?%1%2 [L,R=301]
```

Need to do some fancy stuff with cookies before redirecting the user? Invoke a script that cookies the user and then 301s him to the canonical URL:

```
RewriteCond %{QUERY_STRING} ^source=(\[a-z0-9\]*)$
RewriteRule ^(.*)$ /cookiefirst.php?source=%1&dest=$1 [L]
```

Note the lack of a [R=301] flag in the preceding code. That’s intentional. There’s no need to expose this script to the user. Use a rewrite and let the script itself send the 301 after it has done its work.

Other canonicalization issues worth correcting with rewrite rules and the [R=301] flag include when the engines index online catalog pages under HTTPS URLs, and when URLs are missing a trailing slash that should be there. First, the HTTPS fix:

```
# redirect online catalog pages in the /catalog/ directory if HTTPS
RewriteCond %{HTTPS} on
RewriteRule ^catalog/(.*) http://www.yourdomain.com/catalog/$1 [L,R=301]
```

Note that if your secure server is separate from your main server, you can skip the RewriteCond line.

Now to append the trailing slash:

```
RewriteRule ^(.*/)?$ /$1/ [L,R=301]
```

After completing a URL rewriting project to migrate from dynamic URLs to static, you’ll want to phase out the dynamic URLs not just by replacing all occurrences of the legacy URLs on your site, but also by 301-redirecting the legacy dynamic URLs to their static equivalents. That way, any inbound links pointing to the retired URLs will end up leading both spiders and humans to the correct new URL—thus ensuring that the new URLs are the ones that are indexed, blogged about, linked to, and bookmarked, and the old URLs will be removed from the index. Generally, here’s how you’d accomplish that:

```
RewriteCond %{QUERY_STRING} id=([0-9]+)
RewriteRule ^get_product\.php$ /products/%1.html? [L,R=301]
```

However, you’ll get an infinite loop of recursive redirects if you’re not careful. One quick-and-dirty way to avoid that situation is to add a nonsense parameter to the destination URL for the rewrite and ensure that this nonsense parameter isn’t present before you do the redirect. Specifically:

```
RewriteCond %{QUERY_STRING} id=([0-9]+)
RewriteCond %{QUERY_STRING} !blah=blah
RewriteRule ^get_product\.php$ /products/%1.html? [L,R=301]
RewriteRule ^products/([0-9]+)/?$ /get_product.php?id=$1&blah=blah [L]
```

Notice that this example used two RewriteCond lines, stacked on top of each other. All redirect conditions listed together in the same block will be “ANDed” together. If you wanted the conditions to be “ORed,” you’d need to use the [OR] flag.

How to Redirect a Home Page Index File Without Looping

Many websites link to their own home page in a form similar to *http://www.yourdomain.com/index.html*. The problem with that is that most incoming links to the site’s

home page specify *http://www.yourdomain.com*, thus dividing the link authority into the site. Once a publisher realizes this, she will want to fix her internal links and then 301-redirect *http://www.yourdomain.com/index.html* to *http://www.yourdomain.com/*, but recursive redirects can develop if she does not do this correctly.

When someone comes to your website by typing in *http://www.yourdomain.com*, the DNS system of the Web helps the browser locate the web server for your website. The web server decides what to show to the browser by loading a file from its hard drive.

When no file is specified (i.e., as in the preceding example, where only the domain name is given), the web server loads the default file, which is often a file with a name such as *index.html*, *index.htm*, *index.shtml*, *index.php*, or *default.asp*.

The filename can actually be anything, but most web servers default to one type of filename or another. Where the problem comes in is that many CMSs will expose both forms of your home page, both *http://www.yourdomain.com* and *http://www.yourdomain.com/index.php*.

Perhaps all the pages on the site link only to *http://www.yourdomain.com/index.php*, but given human nature, most of the links to your home page from third parties will most likely point at *http://www.yourdomain.com/*. This can create a duplicate content problem if the search engine now sees two versions of your home page and thinks they are separate, but duplicate, documents. Google is pretty smart at figuring out this particular issue, but it is best to not rely on that.

Because you learned how to do 301 redirects, you might conclude that the solution is to 301-redirect *http://www.yourdomain.com/index.php* to *http://www.yourdomain.com/*. Sounds good, right? Unfortunately, there is a big problem with this approach.

What happens is the server sees the request for *http://www.yourdomain.com/index.php* and then sees that it is supposed to 301-redirect that to *http://www.yourdomain.com/*, so it does. But when it loads *http://www.yourdomain.com/* it retrieves the default filename (*index.php*) and proceeds to load *http://www.yourdomain.com/index.php*. Then it sees that you want to redirect that to *http://www.yourdomain.com/*, and it creates an infinite loop.

The default document redirect solution

The solution that follows is specific to the preceding *index.php* example. You will need to plug in the appropriate default filename for your own web server.

1. Copy the contents of *index.php* to another file. For this example, we'll be using *sitehome.php*.
2. Create an Apache `DirectoryIndex` directive for your document root. Set it to *sitehome.php*. Do not set the directive on a serverwide level; otherwise, it may cause problems with other folders that still need to use *index.php* as a directory index.

3. Put this in an *.htaccess* file in your document root: *DirectoryIndex sitehome.php*. Or, if you aren't using per-directory context files, put this in your *httpd.conf*:

```
<Directory /your/document/root/examplesite.com/>  
    DirectoryIndex sitehome.php  
</Directory>
```

4. Clear out the contents of your original *index.php* file. Insert this line of code:

```
<? header("Location: http://www.example.com"); ?>
```

This sets it up so that *index.php* is not a directory index file (i.e., the default filename). It forces *sitehome.php* to be read when someone types in the canonical URL (*http://www.yourdomain.com*). Any requests to *index.php* from old links can now be 301-redirected while avoiding an infinite loop.

If you are using a CMS, you also need to make sure when you are done with this process that all the internal links now go to the canonical URL, *http://www.yourdomain.com*. If for any reason the CMS started to point to *http://www.yourdomain.com/sitehome.php* the loop problem would return, forcing you to go through this entire process again.

Content Management System Issues

When looking to publish a new site, many publishers may wonder whether they need to use a content management system (CMS), and if so, how to ensure that it is SEO-friendly.

It is essential to determine whether you need a CMS before you embark on a web development project. You can use the flowchart in [Figure 6-44](#) to help guide you through the process.

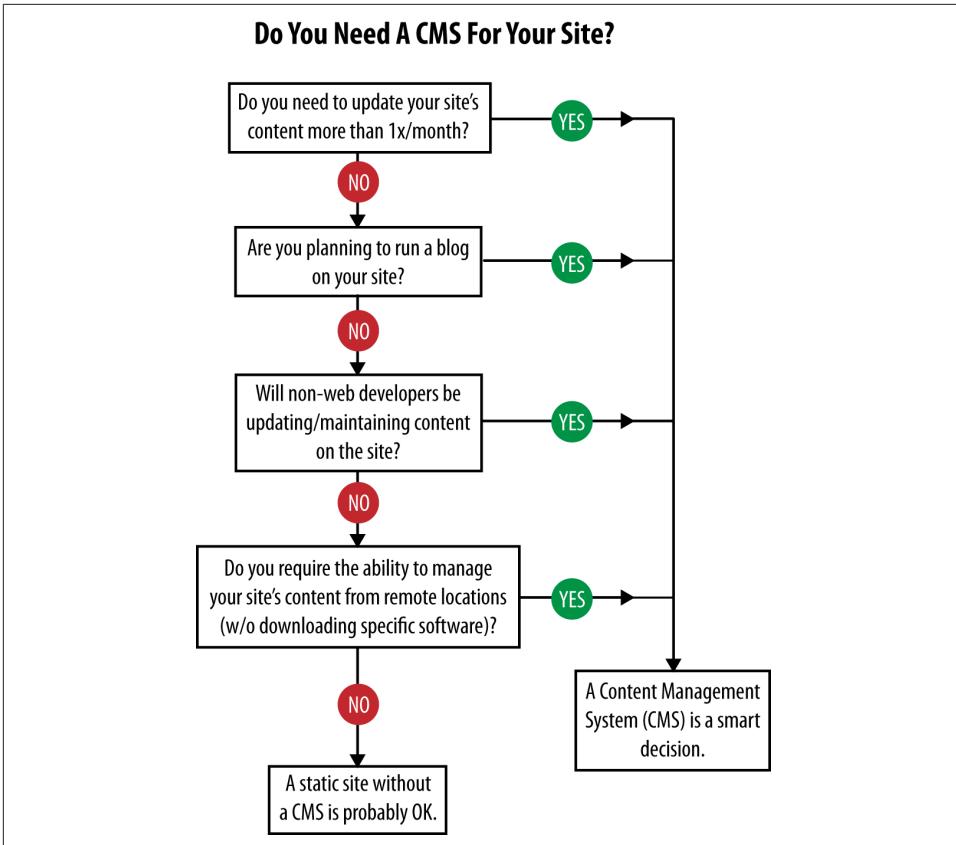


Figure 6-44. Flowchart to determine whether you need a CMS

Due to the inexpensiveness of customizable, free platforms such as **Drupal**, **Joomla**, **WordPress**, and **Weebly**, it is increasingly rare for a publisher to develop a static site, even when a CMS isn't required.

The next step involves understanding how to ensure that a CMS will be search engine-friendly. Here is a list of basic SEO issues that frequently plague a CMS (both prebuilt and custom-made systems). By dealing with these, you will ensure a relatively smooth platform for content delivery:

<title> tag customization and rules

A search engine-friendly CMS must allow for `<title>` tags not only to be customized on a page-specific level, but also to enable rules for particular sections of a website. For example, if the `<title>` tag always has to start with your site name followed by a colon followed by your article title, your on-page optimization efforts will be limited—at least as far as the powerful `<title>` tag is concerned. You

should be able to revise the formulas you use to generate the <title> tags across your site to make them more search-optimal.

Static, keyword-rich URLs

URLs have historically been the most problematic SEO issue for CMS platforms. Nowadays, a search-friendly CMS should feature custom URL creation. In WordPress, a custom URL is referred to as a *post slug*. Figure 6-45 is an example from WordPress.

Notice how the first line allows you to create the title of the post, and the second enables you to manually create the URL structure (and an automatic Generate button if you prefer to simply use the post title).

The image shows a screenshot of the 'Compose Entry' form in a CMS. At the top, the title 'Compose Entry' is displayed in blue. Below it, there are two main sections. The first section is labeled 'Title' and contains a text input field with the placeholder text 'lorem ipsum gort obonor'. The second section is labeled 'Title in URL' and includes a small explanatory text: 'For example, entering 'your-blog-entry' would make your post accessible via http://www.seomoz.org/blog/your-blog-entry'. Below this text is another text input field with the placeholder 'lorem-ipsum-gort-obonor' and a 'Generate' button to its right.

Figure 6-45. Example of custom URL creation

Meta tag customization

Being able to implement custom meta descriptions and meta robots tags is critical. Enabling editorial control is essential for a good CMS.

Enabling custom HTML tags

A good CMS has to offer extra functionality on HTML tags for features such as nofollow on links, or <hx> tags for headlines and subheadlines. These can be built in features accessible through menu options, or the CMS can simply allow for manual editing of HTML in the text editor window when required. Having no <h1> tags on a given page, having too many <h1> tags on the page, or marking up low-value content (such as the publication date) as an <h1> is not desirable. The article title is typically the best content to have wrapped in an <h1>.

Internal anchor text flexibility

For your site to be “optimized” rather than simply search-friendly, it’s critical to customize the anchor text on internal links. Rather than simply making all links in a site’s architecture the page’s title, a great CMS should be flexible enough to handle custom input from the administrators for the anchor text of category-level or global navigation links.

Intelligent categorization structure

Another common CMS problem is poor category structure. When designing an information architecture for a website, you should not place limits on how pages

are accessible due to the CMS's inflexibility. A CMS that offers customizable navigation panels will be the most successful in this respect.

Pagination controls

Pagination can be the bane of a website's search rankings, so controlling it by including more items per page, more contextually relevant anchor text (e.g., not "next," "prev," and page numbers), and careful use of meta `noindex` tags will make your important content get more link authority and crawl attention.

301-redirect functionality

Many content management systems sadly lack this critical feature, disallowing the proper redirection of content when necessary; 301s are valuable for expired content, for pages that have a newer version, and for dodging keyword cannibalization issues similar to those we discussed earlier in this chapter.

XML/RSS pinging

Although it is primarily useful for blogs, any content—from articles to products to press releases—can be issued in a feed. By utilizing quick, accurate pinging of the major feed services, you limit some of your exposure to duplicate content spammers who pick up your feeds and ping the major services quickly in the hopes of beating you to the punch.

Image handling and alt attributes

`alt` attributes are a clear must-have from an SEO perspective, serving as the "anchor text" when an image is used as a link (note that text links are much better than images with `alt` attributes, but if you must use image links you should implement the `alt` attribute) and providing relevant, indexable content for the search engines. Images in a CMS's navigational elements should preferably use CSS image replacement rather than mere `alt` attributes.

CSS exceptions

The application of CSS styles in a proper CMS should allow for manual exceptions so that a user can modify how a strong headline or list element appears visually. If the CMS does not offer this, writers may opt out of using proper semantic markup for presentation purposes, which would not be a good thing.

Static caching options

Many content management systems currently offer caching options, which are a particular boon if a page is receiving a high level of traffic from social media portals or news sites. A bulky CMS often makes dozens of extraneous database connections, which can overwhelm a server if caching is not in place, killing potential inbound links and media attention.

URLs free of tracking parameters and session IDs

Sticking session or tracking information such as the user's click path into the URL is deadly for SEO. It usually leads to incomplete indexation and duplicate content issues.

Customizable URL structure

If the default URL structure of the CMS doesn't suit your needs, you should be able to change it. For example, if you don't want */archives/* in the URLs of all your archived articles, you should be able to remove it. Or if you want to reference the article name instead of the article's database ID in the URL, you should be able to do it.

301 redirects to a canonical URL

Duplicate content is a major concern for the dynamic website owner. Automatic handling of this by the CMS through the use of 301 redirects is a must.

Static-looking URLs

The most palatable URLs to spiders are the ones that look like they lead to static pages—no query strings in the URL.

Keywords in URLs

Keywords in your URLs (used judiciously) can help your rankings.

RSS feeds

The CMS should autcreate RSS feeds to help your site rank in Google Blog Search and other feed engines.

Multilevel categorization structure

It is awfully limiting to your site structure and internal hierarchical linking structure to have a CMS that doesn't allow you to nest subcategories into categories, sub-subcategories into subcategories, and so on.

Paraphrasable excerpts

Duplicate content issues are exacerbated on dynamic sites such as blogs when the same content is displayed on permalink pages, category pages, archives-by-date pages, tag pages, and the home page. Crafting unique content for the excerpt, and having that content display on all locations except the permalink page, will help strengthen your permalink page as unique content.

Breadcrumb navigation

Breadcrumb (drill-down) navigation is great for SEO because it reinforces your internal hierarchical linking structure with keyword-rich text links.

Meta noindex tags for low-value pages

Even if you use `nofollow` attributes in links to these pages, other people may still link to them, which carries a risk of ranking those pages above some of your more valuable content.

Keyword-rich intro copy on category-level pages

Keyword-rich introductory copy helps set a stable keyword theme for the page, rather than relying on the latest article or blog post to be the most prominent text on the page.

nofollow links in comments

If you allow visitors to post comments and do not `nofollow` the links, your site will be a spam magnet. Heck, you'll probably be a spam magnet anyway, but you won't risk losing PageRank to spammers if you use `nofollow` attributes.

Customizable anchor text on navigational links

Contact, About Us, Read More, Full Article, and so on make for lousy anchor text—at least from an SEO standpoint. Hopefully, your CMS allows you to improve such links to make the anchor text more keyword-rich.

XML sitemap generator

Having your CMS generate your XML sitemap can save a lot of hassle, as opposed to trying to generate one with a third-party tool.

HTML4, HTML5, or XHTML validation

Although HTML validation is not a ranking signal, it is desirable to have the CMS automatically check for malformed HTML, as search engines may end up seeing a page differently from how it renders on the screen and accidentally consider navigation to be part of the content or vice versa.

Pingbacks, trackbacks, comments, and antispam mechanisms

The problem with comments/trackbacks/pingbacks is that they are vectors for spam, so if you have one or more of these features enabled, you will be spammed. Therefore, effective spam prevention in the form of Akismet, Mollom, or Defensio is a must.

If you want more information on picking a quality CMS, some great web resources are already out there—among them OpenSourceCMS.com and [CMSmatrix](http://CMSmatrix.com)—to help manage this task.

CMS Selection

There are many factors to consider when choosing an existing CMS. Many CMS platforms are free, but some of them are proprietary with a license cost per site. The majority were not designed with security, stability, search friendliness, and scalability in mind, though in recent years a few vendors have developed excellent systems that

have search friendliness as their primary focus. Many were developed to fit a certain market niche, but can be expanded to fit other purposes. Some are no longer maintained. Many are supported and developed primarily by hobbyists who don't particularly care if you're having trouble getting them installed and configured. Some are even intentionally made to be difficult to install and configure so that you'll be encouraged to pay the developers a consulting fee to do it all for you.

Popular CMS solutions that the authors have experience with include [Joomla](#), [Drupal](#), [concrete5](#), [Pixelsilk](#), [WordPress](#), [Magento](#), and [Sitecore](#). Each has strong support for SEO, but requires some configuration for optimal results. Make sure you get that help up front to get the SEO for your site off to a strong start.

Selecting a CMS is an important process. If you make the wrong choice, you will be faced with limited SEO options. Like most software, a CMS is a moving target—what's missing today may be a new feature tomorrow. In addition, just because a feature exists doesn't mean it is the default option, so in many instances the desired functionality will need to be enabled and possibly customized to work to your specifications.

Third-Party CMS Add-Ons

Many CMS platforms offer third-party plug-ins or add-ons that extend the core functionality of the CMS. In the WordPress plug-in directory alone, there are over 34,000 plug-ins, including the hugely popular [WordPress SEO by Yoast](#) and [All in One SEO Pack](#). Plug-ins provide a simple way to add new SEO features and functionality, making the CMS much more flexible and future-proof. It is particularly helpful when there is an active community developing plug-ins. An active community also comes in very handy in providing free technical support when things go wrong; and when bugs and security vulnerabilities crop up, it is important to have an active developer base to solve those issues quickly.

Many CMS add-ons—such as discussion forums, customer reviews, and user polls—come in the form of independent software installed on your web server, or hosted services. Discussion forums come in both of these forms: bbPress, which is installed software and is optimized for search, and vbulletin, which is a hosted solution and therefore more difficult to optimize for search.

The problem with hosted solutions is that you are helping to build the service providers' link authority and not your own, and you have much less control over optimizing the content.

As we referenced several times earlier in this chapter, Flash is popular on the Web, but presents challenges to the search engines in terms of indexing the related content. This creates a gap between the user experience with a site and what the search engines can find on that site.

In the past, search engines did not index Flash content at all. In June 2008, **Google announced that it was offering improved indexing of this content**. This announcement indicates that Google can index text content and find and follow links within Flash files. However, Google still cannot tell what is contained in images within the Flash file. Here are some reasons why Flash is still not fully SEO-friendly:

Different content is not on different URLs

This is the same problem you encounter with AJAX-based pages. You could have unique frames, movies within movies, and so on that appear to be completely unique portions of the Flash site, yet there's often no way to link to these individual elements.

The breakdown of text is not clean

Google can index the output files in the *.swf* file to see words and phrases, but in Flash a lot of your text is not inside clean `<h1>` or `<p>` tags; it is jumbled up into half-phrases for graphical effects and will often be output in the incorrect order. Worse still are text effects that often require "breaking" words apart into individual letters to animate them.

Flash gets embedded

A lot of Flash content is linked to only by other Flash content wrapped inside shell Flash pages. This line of links, where no other internal or external URLs are referencing the interior content, means some very low PageRank/link authority documents. Even if they manage to stay in the main index, they probably won't rank for anything.

Flash doesn't earn external links like HTML

An all-Flash site might get a large number of links to the home page, but interior pages almost always suffer. For embeddable Flash content, it is the HTML host page earning those links when they do come.

SEO basics are often missing

Anchor text, headlines, bold/strong text, `img alt` attributes, and even `<title>` tags are not simple elements to properly include in Flash. Developing Flash with SEO in mind is just more difficult than doing it in HTML. In addition, it is not part of the cultural lexicon of the Flash development world.

A lot of Flash isn't even crawlable

Google has indicated that it doesn't execute external JavaScript calls (which many Flash-based sites use) or index the content from external files called by Flash (which, again, a lot of Flash sites rely on). These limitations could severely impact what a visitor can see versus what Googlebot can index.

Note that in the past you could not test the crawlability of Flash, but the Adobe Search Engine SDK now gives you an idea of how the search engines will see your Flash file.

Flash Coding Best Practices

If Flash is a requirement for whatever reason, there are best practices you can implement to make your site more accessible to search engine spiders. What follows are some guidelines on how to obtain the best possible results.

Flash meta tags

Beginning with Adobe/Macromedia Flash version 8, there has been support for the addition of title and description meta tags to any *.swf* file. Not all search engines are able to read these tags yet, but it is likely that they will soon. Get into the habit of adding accurate, keyword-rich `<title>` tags and meta tags to files now so that as search engines begin accessing them, your existing *.swf* files will already have them in place.

Adobe Flash search engine SDK

Flash developers may find the SDK useful for server-based text and link extraction and conversion purposes, or for client-side testing of their Flash content against the basic Adobe (formerly Macromedia) Flash Search Engine SDK code.

Tests have shown that Google and other major search engines now extract some textual content from Flash *.swf* files. It is unknown whether Google and others have implemented Adobe's specific Search Engine SDK technology into their spiders, or whether they are using some other code to extract the textual content. Again, tests suggest that what Google is parsing from a given *.swf* file is very close to what can be extracted manually using the Search Engine SDK.

The primary application of Adobe's Search Engine SDK is desktop testing *.swf* files to see what search engines are extracting from a given file. The program cannot extract files directly from the Web; the *.swf* file must be saved to a local hard drive. The program is DOS-based and must be run in the DOS Command Prompt using DOS commands.

By running a *.swf* file through the Flash SDK `swf2html` program during development, you can edit or augment the textual assets of the file to address the best possible SEO practices—homing in primarily on keywords and phrases along with high-quality links. Because of the nature of Flash and the way in which it deals with both text and animation, it is challenging to get exacting, quality SEO results. The goal is to create the best possible SEO results within the limitations of the Flash program and the individual Flash animation rather than to attempt the creation of an all-encompassing SEO campaign. Extracted content from Flash should be seen as one tool among many in a larger SEO campaign.

Internal Flash coding

There are several things to keep in mind when preparing Flash files for SEO:

- Search engines currently do not read traced text (using the `trace()` function) or text that has been transformed into a shape in Flash (as opposed to actual characters). Only character-based text that is active in the Flash stage will be read (see [Figure 6-46](#)).
- Animated or affected text often creates duplicate content. Static text in Flash movies is not read as the duplicate instances that “tweening” and other effects can create. Use static text, especially with important content, so that search engines do not perceive the output as spam (see [Figure 6-47](#)).
- Search engine spiders do not see dynamically loaded content (text added from an external source, such as an XML file).
- The font size of text does not affect search engines; they read any size font.
- Special characters such as `<`, `>`, `&`, and `“` are converted to HTML character references (`<`, `>`, `&`, and `"`;) and should be avoided.
- Search engines find and extract all URLs stored within the `getURL()` command.
- Search engines have the ability to follow links in Flash, though it is an “iffy” proposition at best. They will not, however, follow links to other Flash `.swf` files. (This is different from loading child `.swf` files into a parent `.swf` file.) Therefore, links in Flash should always point to HTML pages, not other `.swf` files.

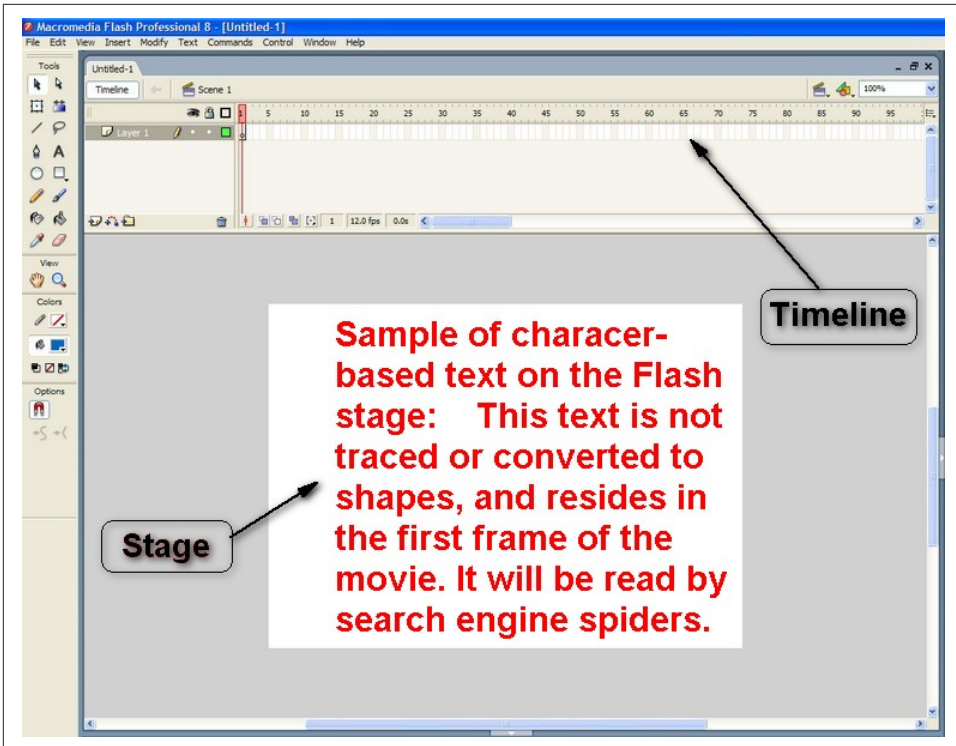


Figure 6-46. Example of spider-readable text inside a Flash program

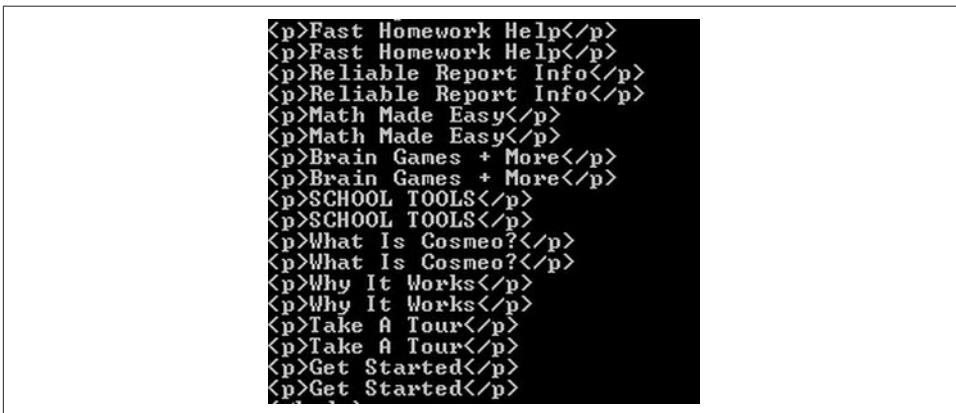


Figure 6-47. Animated text results in Flash source; can be seen as duplicate content

SWFObject library and <noscript> tag

Because “alternative content” workarounds for SEO of Flash files have historically been abused by spammers, it is challenging to recommend these tactics to optimize your Flash files without a critical disclaimer.

Both the SWFObject library and <noscript> tag were originally designed to be legitimate, graceful degradation techniques readily accepted by the search engines as a way to accommodate older browsers or people with special needs. But many unscrupulous sites have used the code to trick search engine spiders. In other words, these methods are used in such a way that browsers display one thing to users, but something completely different to search engine spiders. As you’ve learned in this chapter, all of the major search engines disapprove of such tactics.

Websites using such methods today are often penalized or removed from search engine indexes altogether. This makes graceful degradation risky on some level, but if the methods are used clearly within the boundaries for which they were intended, getting penalized or banned is highly unlikely.

As we discussed earlier in this chapter, intent is an essential element search engines take into consideration. If your intent is to provide *all* users with a positive experience while they’re visiting your site, you should be fine. If your intent is to game the search engines, all it takes is one online rival to report your site for spam to incur the wrath of the search engines.

Google and other search engines do not algorithmically ban sites for using SWFObject and <noscript>; it usually requires human intervention to evoke a penalty or outright ban.

SWFObject. SWFObject is the better of the two Flash optimization options because it is JavaScript code designed specifically for Flash *.swf* purposes, and it has been abused to a lesser extent than the <noscript> tag.

SWFObject is a Flash detection code library written in JavaScript that checks whether a browser has the Flash plug-in. If the browser does have the Flash plug-in, the *.swf* file is displayed secondary to that detection. If the browser does not have the Flash plug-in or the JavaScript to detect it, the primary, alternative content contained within <div> tags is displayed instead. The key here is that search engine spiders do not render the JavaScript. They read the primary content in the <div> tags.

The opportunity for abuse is obvious when you view the code. This small piece of code is placed within the <head> tags:

```
<script type="text/javascript" src="swfobject.js"></script>
```

In the body of the text, the code looks something like [Figure 6-48](#).

```

<script type="text/javascript" src="swfobject.js"></script>

<div id="flashcontent">
  Text, links, and graphics placed here are replaced by the Flash movie. Search
  engine spiders will read this information, but the browser with an active Flash
  plugin will show the Flash movie instead.
</div>

<script type="text/javascript">
  var so = new SWFObject("whatever.swf", "themovie", "200", "100", "7", "#336699");
  so.write("flashcontent");
</script>

```

Figure 6-48. Information between the `<div>` HTML tags read by search engine spiders

Search engine spiders will read text, links, and even `alt` attributes within the `<div>` tags, but the browser will not display them unless the Flash plug-in isn't installed (about 95% of browsers now have the plug-in) or JavaScript isn't available.

Once again, the key to successfully implementing SWFObject is to use it to the letter of the law: leverage it to mirror the content of your Flash `.swf` file *exactly*. Do not use it to add content, keywords, graphics, or links that are not contained in the file. Remember, a human being will be making the call as to whether your use of SWFObject is proper and in accordance with that search engine's guidelines. If you design the outcome to provide the best possible user experience, and your intent is *not* to game the search engines, you are probably OK.

You can download [the SWFObject JavaScript library](#) free of charge. Included in this download is the `flashobject.js` file, which is placed in the same directory as the web pages on which the corresponding calling code resides.

<noscript>. The `<noscript>` tag has been abused in "black hat" SEO attempts so frequently that you should be cautious when using it. Just as SWFObject and `<div>` tags can be misused for link and keyword stuffing, so too can the `<noscript>` tag. Certain companies have promoted the misuse of the `<noscript>` tag widely; consequently, there have been many more problems with its use.

With that being said, conservative and proper use of the `<noscript>` tag specifically with Flash `.swf` files can be an acceptable and good way to get content mirrored to a Flash file read by search engine spiders. As is the case with SWFObject and corresponding `<div>` tags, content must echo that of the Flash `.swf` movie exactly. Do not use `<noscript>` to add content, keywords, graphics, or links that are not in the movie. Again, it is a human call as to whether a site or individual page is banned for the use or misuse of the `<noscript>` tag.

You use `<noscript>` with Flash `.swf` files in the following manner:

```
<script type="text/javascript" src="yourflashfile.swf"></script>
```

Followed at some point by:

```
<noscript>
<h1>Mirror content in Flash file here.</h1>
<p>Any content within the noscript tags will be read by the search engine
spiders, including links
http://www.mirroredlink.com, graphics, and corresponding alt attributes.
</noscript>
```

For browsers that do not have JavaScript installed or functioning, content alternatives to JavaScript-required entities are displayed. So, for use with Flash *.swf* files, if a browser does not have JavaScript and therefore cannot display Flash, it displays instead the content inside the `<noscript>` tags. This is a legitimate, graceful degradation design. For SEO purposes, as is true with SWFObject, the search engine spiders do not render the JavaScript but do read the content contained in the HTML. Here, it is the content between the `<noscript>` tags.

Scalable Inman Flash Replacement

Scalable Inman Flash Replacement (sIFR) is a technique that uses JavaScript to read in HTML text and render it in Flash instead. The essential fact to focus on here is that the method guarantees that the HTML content and the Flash content are identical. One great use for this technique is to render headline text in an anti-aliased font (this is the purpose for which sIFR was designed). This can greatly improve the presentation of your site.

Dan Crow, head of Google's Crawl team, said that as long as this technique is used in moderation, it is OK. However, extensive use of sIFR could be interpreted as a signal of poor site quality. Because sIFR was not designed for large-scale use, such extensive use would not be wise in any event.

It is worth noting that there are comparable technologies available to web designers for improved type presentation, which provide similar search engine friendliness. FaceLift Image Replacement (FLIR) is an image replacement script similar to sIFR in its use of JavaScript, but without the Flash element, and there is [a handy WordPress plug-in for implementation on WordPress-based websites](#). Google also offers [its own set of fonts optimized for use on websites](#).

Best Practices for Multilanguage/Country Targeting

Many businesses target multiple countries with their websites and need answers to questions such as: Do you put the information for your products or services all on the same domain? Do you obtain multiple domains? Where do you host the site(s)? As it turns out, there are SEO factors, as well as basic marketing questions, that affect the answers. There are also non-SEO factors, such as the tax implications of what you do;

you can get some TLDs only by having a local physical presence (e.g., France requires this to issue a *.fr* domain).

How to Target a Specific Country

Starting with the basics of international targeting, it is important to let the search engines know where your business is based in as many ways as possible. These might include:

- Using a country code TLD (ccTLD) for your domain (e.g., *.uk*)
- Hosting your site locally (more for content delivery speed than for the “localness” factor)
- Displaying the physical local address in plain text on every page of your site
- Setting Google Search Console geotargeting to your country of interest
- Verifying your address with Google Maps
- Including links from in-country websites
- Using the local language on the website

If you are starting from scratch, getting these components all lined up will give you the best possible chance of ranking in the local country you are targeting.

Problems with Using Your Existing Domain

You may ask why you cannot leverage your domain weight to target the new territory rather than starting from scratch—in other words, why can’t you create multiple versions of your site and determine the user’s location before either delivering the appropriate content or redirecting him to the appropriate place in the site (or even to a sub-domain hosted in the target country)?

The problem with this approach is that the search engines spider from the United States, meaning their IP addresses will be in the United States in your lookup and thus they will be delivered only U.S. content from your site. This problem is exacerbated if you are going even further and geodelivering content in different languages, as only your English language content will be spidered unless you cloak for the search engine bots.

This kind of IP delivery is therefore a bad idea. You should make sure you do not blindly geodeliver content based on IP address, as you will ignore many of your markets in the search engines’ eyes.

The Two Major Approaches

The best practice remains one of two approaches, depending on the size and scale of your operations in the new countries and how powerful and established your *.com* domain is.

If you have strong local teams and/or (relatively speaking) less power in your main domain, launching independent local websites geotargeted as described earlier is a smart move in the long run.

If, on the other hand, you have only centralized marketing and PR and/or a strong main domain, you may want to create localized versions of your content either on country-specific subdomains (*http://uk.yourdomain.com*, *http://au.yourdomain.com*, etc.) or in subfolders (*/uk/*, */au/*, etc.), with the preference being for the use of subdomains so that you can set up local hosting.

Both the subdomain and the subfolder approach allow you to set your geotargeting option in Google Search Console, and with either method, you have to be equally careful of duplicate content across regions. In the subdomain example, you can host the subdomain locally, while in the subfolder case, more of the power of the domain filters down.

Unfortunately, the Search Console's geotargeting option doesn't work nearly as well as you'd hope to geotarget subfolders. The engines will consider hosting and ccTLDs, along with the geographic location of your external link sources, to be stronger signals than the manual country targeting in the tools. In addition, people in other countries (e.g., France) don't like to click on *.com* or *.org* TLDs; they prefer *.fr*. This extends to branding and conversion rates too—web users in France like to buy from websites in France that end in *.fr*.

Multiple-Language Issues

An entire treatise could be written on handling multilanguage content as the search engines themselves are rapidly evolving in this field, and tactics are likely to change dramatically in the near future. Therefore, this section will focus on providing you with the fundamental components of successful multilanguage content management.

Here are best practices for targeting the search engines as of this writing, using Spanish and English content examples:

- Content in Spanish and English serving the same country:
 - Create a single website with language options that change the URL by folder structure; for example, *http://www.yourdomain.com* versus *http://www.yourdomain.com/esp/*.

- Build links from Spanish and English language sites to the respective content areas on the site.
- Host the site in the country being served.
- Register the appropriate country domain name (for the United States, *.com*, *.net*, and *.org* are appropriate, whereas in Canada using *.ca* or in the United Kingdom using *.uk* is preferable).
- Mark up your HTML code using `hreflang` tags for multiple languages. See “[hreflang for multiple languages/no specific location](#)” on page 379.
- Content in Spanish and English targeting multiple countries:
 - Create two separate websites, one in English targeting the United States (or the relevant country) and one in Spanish targeting the relevant Spanish-speaking countries.
 - Host one site in the United States (for English) and the other in the relevant countries for the Spanish version.
 - Register different domains, one using U.S.-targeted domain extensions and one using the Spanish-speaking countries’ extension.
 - Acquire links from the United States to the English site and links from the Spanish-speaking countries to that site.
 - Mark up your HTML code using `hreflang` tags for multiple languages/locations. See “[hreflang for multiple languages/regions](#)” on page 380.
- Content in Spanish targeting multiple countries:
 - Create multiple websites (as mentioned earlier) targeting each specific country.
 - Register domains using the appropriate country TLD and host in each country separately.
 - When possible, have native speakers fluent in the specific region’s dialect write the site content for each specific country.
 - Obtain in-country links to your domains.
 - Mark up your HTML code using `hreflang` tags for multiple languages/locations. See “[hreflang for one language/multiple regions](#)” on page 380.

Although some of these approaches may seem counterintuitive, the joint issues of search engines preferring to show content hosted in and on a country-specific domain name combined with duplicate content problems make for these seemingly illogical suggestions.

hreflang markup

There are several options for serving multiregion and multilanguage content to the search engines. You'll need to use at least one of these solutions to encourage Google to rank the appropriate version of your content in the appropriate version of the Google search engine (*google.com*, *google.co.uk*, *google.ca.*, etc.). These solutions are also necessary to prevent duplicate content issues both within a single language—such as American English and UK English, where the copy is likely to be virtually identical—and also across languages that are more unique.

There are three main options available to serve up multilanguage or multiregion content:

- Code within the server header section of a page
- Code within the `<head>` section of the HTML on a page
- Special directives within the site's XML sitemap, or a specific multiregion/multilanguage sitemap

It's recommended that you use only one of these solutions at a time. While redundancy, if accurate, will cause no negative effects, there's the possibility of disagreement between multiple solutions if they are working simultaneously, which can confuse the search engines about which version to "count."

We will focus on the second option: code within the `<head>` section of the HTML on a page.

hreflang for multiple languages/no specific location

Each page that has alternate language versions, but not alternate country versions, should contain markup specifying the language only. It is acceptable for pages to contain language-only markup but never region-only markup. Once pages are built for specific regions, they must be marked up with a combination of both language and region markup. An example of this markup for a home page presented in both English and Spanish follows.

If the home page of a site, in this case *example.com*, is translated into both English and Spanish, both versions of the page should include code such as:

```
<link rel="alternate" hreflang="x-default" href="example.com" />
<link rel="alternate" href="example.com/es/" hreflang="es" />
```

Each language will have its own unique hreflang code. Note that there is no accommodation within the language markup for the difference between Spanish for Spain and Spanish for Latin America. Similarly, there is no difference in the language

markup between Portuguese for Portugal and Portuguese for Brazil, or Canadian French versus the version spoken in France, and so on.

A full list of the two-character language codes can be found at http://www.loc.gov/standards/iso639-2/php/code_list.php under the ISO 639-1 standard.

hreflang for multiple languages/regions

If you wanted to have a default version of the page (English language, no region assigned), a version for Spanish from Mexico, and a version for Spanish from Spain, the markup would look similar to the following. Please note that each region/language combination would need its own unique URL/domain:

```
<link rel="alternate" hreflang="x-default" href="example.com" />
<link rel="alternate" href="example.es/" hreflang="es-es" />
<link rel="alternate" href="example.com.mx/" hreflang="es-mx" />
```

Another consideration in your geotargeting of URLs is that there are no provisions for markup associated with “regions” such as Latin America, APAC, the EU, and so on. Each country within these regions must be treated individually.

hreflang for one language/multiple regions

If you wished to have versions in Spanish for Spain and versions for Spanish for Latin America, you would use markup similar to the following:

```
<link rel="alternate" hreflang="x-default" href="example.es" />
<link rel="alternate" href="example.es/" hreflang="es-es" />
<link rel="alternate" href="example.com.mx/" hreflang="es-mx" />
<link rel="alternate" href="example.com.cr/" hreflang="es-cr" />
<link rel="alternate" href="example.com.com.ar/" hreflang="es-ar" />
```

...and so on for each Latin American country.

A full list of country-level TLDs can be found at <http://www.mcanerin.com/EN/articles/ccTLD.asp>.

It pays to plan ahead when adding hreflang markup for alternate language/country versions of your site. Each alternate version of a page needs to reference every other alternate version. If you have a version of a page in English for the United States and another in Spanish for Mexico, both of those pages need the markup referencing the other version. If you were to then add a version in Spanish for Spain, not only does this new version need to reference both the English/U.S. version and the Spanish/Mexican version, but both of those pages now also need to reference this new Spanish/Spain version.

This level of complexity is why it is crucial that before you proceed with creating alternate language/region versions of your content, you have a comprehensive interna-

tional strategy. Before you embark on any site changes that are detectable by the search engines, you should be planning ahead several years as to what regions and languages you will optimize for. If you choose not to do this level of planning, you may face numerous code changes across all alternate-version pages as new countries/languages are added in the future.

For more information on hreflang markup, see [“Use hreflang for language and regional URLs - Search Console Help”](#) and the blog post [“Using the Correct Hreflang Tag: A New Generator Tool”](#) by [Aleyda Solis](#). Aleyda also created the [“hreflang Tags Generator Tool”](#).

NOTE

A special thanks to [Rob Woods](#) for his contributions to this portion of the chapter.

Semantic Search

There is a lot of confusion over the definition of semantic search. Some of this confusion comes from the formal definition of *semantics* commonly associated with linguistics, and some of it comes from the misunderstanding that arises the moment the words “structured data” are mentioned.

In truth, semantic search has a little to do with both, and a lot to do with the four vectors that drive Big Data across the Web:

- Volume is about processing massive amounts of data and extracting unique meaning from it.
- Velocity refers to the speed at which critical data comes in and how quickly it must be analyzed and processed.
- Variety is required as well, as many different types of data must be handled, such as audio, video, and text.
- Veracity is about the need to validate the accuracy of the data being processed.

To help you understand this concept better, it helps to take things from the beginning, and the true beginning for semantic search was August 30, 2013, when Google quietly rolled out Hummingbird.

The change, which was announced almost a month later on the eve of Google’s 15th birthday, completed Google’s long journey to turn search into more than a blind fishing expedition where those who created content and those who looked for it continually strove to guess each other’s keywords and connect.

Google's Hummingbird

To understand how much search (and SEO) has changed, consider just how far voice search has come. When we use voice, we tend to speak in sentences instead of keywords, and in order for Google to return meaningful answers it has to be able to understand our search query. To handle spoken queries well, Google also needs to understand intent, which requires it to be able to understand context.

The same technology that was applied to Google voice search before August 30, 2013, is now applied to the regular text search with which everyone is familiar. Hummingbird (which Google said was so named because it was precise and fast) does a number of things no search engine had done before.

First, it takes the entire search query into account—not just the keywords, but every word. Second, it looks at who is carrying out the search. Suddenly, variables such as past search history and search patterns are important in delivering the right results, at the right time, to the right person. Thirdly, it also factors in how the search itself is being conducted. Device type, time of day, and location now are also important parameters affecting the search results.

With linguistic sensitivity (i.e., the ability to better process natural language) Google's Hummingbird is also better at understanding the relationships between queries and between bits of data. It is in this space that the real magic happens.

Semantic search, really, is about relational connections and contextual content. In order to deliver “the right results, at the right time, to the right person,” semantic search needs to understand the importance of the query to that person and the importance of the query in relation to the data it already holds in its index and the data it is currently indexing.

Every single item of data that is in the visible Web needs to be crawled, indexed, and evaluated against all the other items of data and then weighed against a particular search query. The net result of this approach is that the traditional first page of Google everyone strove to rank for in the past has now largely disappeared.

While all of this might make it sound like SEO as we've known it is dead, nothing could be further from the truth (though if this were the case, it might have made for a much shorter book).

Semantic Search and SEO

Basic SEO factors are still in play. Links are still important. Keywords still play a role. Content still needs to be created. But as more and more variables are added to the picture, the value ascribed to each one decreases. This makes it difficult to pick specific aspects of search engine optimization, focus on them to the exclusion of everything else, and expect that to be enough.

This is an important change. In the past, you could get away with thin content, for instance, if you had a large number of links coming in that would boost it in search. You could get away with poor-quality design if you had sufficient keywords to draw in the “crowds” to your page through search. You could get away with links in some suspect neighborhoods if you had a sufficiently large number of links for the relatively small percentage of bad links to be overlooked.

You could, in short, take some shortcuts that might have been “bad” when you planned to be “good” once you got where you needed to in terms of search ranking. There was an expectation that the end result justified the risks and that things would balance out in your favor, eventually.

This is no longer the case. Because Google now needs to deliver high-quality, high-confidence results in search it has to have confidence in the content presented. The veracity factor becomes critical. This makes every activity intended to optimize a website—design, content, website structure, keywords, traffic, traffic behavior, social network footprint, links, comments, and citations—crucial. The list is far from exhaustive; everything that helps build a data-driven impression of what a website is all about and the quality of its content now becomes an element or activity you need to consider.

The reason this has happened has to do with two things that are synonymous with semantic search: entities and structured data.

Entities and Semantic Search

An entity is something that exists in itself. It can be real, like a car, or fictional, like a film or book character like Harry Potter. In either case, it possesses properties, qualities, and attributes that make up the “thing” it represents. All of these are language-independent, though obviously when an entity is described we do need a language with which to describe it. The properties, qualities, and attributes, along with associated entities, form the Knowledge Graph that is used to define new entities.

Entities are at the heart of what Google calls the transition from “strings to things.” While all the information on the Web is data, entities allow Google to understand how that information fits in and how accurate it is (the veracity aspect of semantic search). In order for an entity to be created in Google’s index, Google needs to index all the properties, qualities, and attributes about it and understand the relational connections between them. This is exactly why the Web, with semantic search, is becoming more transparent. Data is now portable. Its origin is every bit as important as the data itself. The connections between different data pieces are being indexed, and the importance of the data itself is becoming better understood.

The concept of entities has a large impact on how to pursue SEO. While tasks such as keyword research and getting links to your site remain important, you must also pur-

sue holistic strategies to build the reputation and visibility of your business to create positive associations across the Web.

This brings SEO and marketing a lot closer than they have ever been and makes SEO, as a whole, something that should be part of the DNA of a business rather than a bolt-on activity that can be picked up and dropped as the need arises.

So, how do we begin? What are the guiding principles that you need to have in mind in this new world of SEO? Funnily enough, the concept is as simple to plan as it is difficult to apply. It starts off from the very basic questions of: Who? Why? How?

If you cannot answer these three questions successfully—that is, in a way that expresses a distinct and unique identity for your business—then chances are good that neither can Google or your prospective customers. Your SEO, then, is governed by activities that make sense at a technical level but not at a brand identity one. Semantic search is all about establishing that identity, even—*especially*, one might argue—from a business point of view. This is what helps with the formation of entities in the Google search index.

Entities then become high-trust points that help Google’s semantic search understand the value of information better. As you engage in your overall digital marketing strategy, keep these three areas of concern in focus:

- Trust
- Authority
- Reputation

These three aspects, more than anything else, will help your business find its audience, keep it, and grow.

Structured Data

Structured data is the label applied to a number of markup formats that allow Google to better understand the data it is indexing. Structured data, then, is simply *metadata* (data about data) implemented for search engines rather than people.

Google, Microsoft (Bing), and Yahoo! worked to establish Schema.org, which is an independent, W3C-approved way of implementing structured data across the Web (see the section “[Schema.org](#)” on page 386 for more information). Unfortunately, many SEOs believe that this is a shortcut to better rankings, which it’s not. Semantic search is *all* about structured data. The entire effort that Google has undertaken involves indexing the unstructured data that is found across the Web and then placing it in structured data format within its index.

That does not mean, however, that structured data on a website is a ranking signal. It helps in better indexing, but ranking depends upon other factors including the quality of the content, its value, uniqueness, and even freshness. Despite the fact that Google is one of the founding organizations behind the Schema.org structured data markup initiative, Google will also attempt to extract entity information from unstructured data through its own efforts.

There are several good reasons for this:

- Adoption (structured data markup is notoriously difficult to implement if you do not know any coding)
- Accuracy (the moment human agents are involved in the markup of data, mistakes happen)
- Consistency (even when structured data is applied without errors, there are still differences in the categorization of content and confusion over how to best apply semantic identifiers)
- Reliability (there will always be a temptation to game search by implementing structured data markup in ways intended to boost ranking; Google has already had a number of manual action penalties in search designed to remove such spammy results)

The million-dollar question is: is there anything you can do to help Google index your site better if you do not implement structured data markup?

The answer is yes: implement all the search engine optimization tools you have in your arsenal in a way that makes sense for a human user first, and a search engine second.

Namely, your on-page SEO should help a reader better navigate your content and make sense of it at a glance. The keywords, synonyms, and entities you use in your content should do the same. Any links you include, and the anchor text of those links, must similarly fill in those blanks.

If you're running a brick-and-mortar business, all the relevant information should be included on your pages, such as your name, address, and phone number. You should interlink your web properties (such as your site and social media accounts) with your Google+ presence.

Where possible, on your site, make use of [Google's structured data highlighter tool](#). Finally, make use of [Google My Business](#) and ensure you have a cohesive presence on the Web, whose effectiveness you can measure. Use Schema.org to help search engines better understand the content of your pages.

In addition, build lots of positive relationships on the Web that help drive signals of trust and authority back to your website and business.

NOTE

A special thanks to **David Amerland** for his contributions to this portion of the chapter.

Schema.org

Schema.org is best viewed as part of a much larger idea, one that traces its origins back to the foundational concepts of the Web itself, and its progenitor, Tim Berners-Lee. In their seminal article in *Scientific American* in 2001, Berners-Lee, James Hendler, and Ora Lassila described a semantic web that “will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page...will know not just that [a] page has keywords such as ‘treatment, medicine, physical, therapy’...but also that Dr. Hartman *works* at this *clinic* on *Mondays, Wednesdays* and *Fridays*.”¹⁸

Schema.org is arguably one of the most practical, accessible, and successful outcomes of the semantic web movement to date. With the marketing prowess of Google, Yahoo!, Bing, and Yandex behind it, and with the powerful incentive of gaining additional, more inviting shelf space in the SERPs, it’s no surprise that webmasters are adopting Schema.org at a rapid pace. And Berners-Lee et al.’s words now read like a prophetic description of the search engine spiders crawling the Web and extracting meaning for display in enhanced search results.

At its core, Schema.org is about standardizing and simplifying the process of adding semantic markup to your web pages, and providing tangible benefits for doing so. The most visible such benefits come in the form of *rich snippets*, such as the star ratings and price range shown in **Figure 6-49**.

However, it’s clear that Schema.org markup plays a larger, and perhaps expanding, role in how the SERPs are constructed. Other benefits now attributed to Schema.org include local SEO ranking benefits received from clearly communicating a business’s so-called NAP (name, address, phone number) information by marking it up with Schema.org, and even supplying Google with information that can appear in the knowledge panel and “answer box” results (see **Figure 6-50** and **Figure 6-51**).

¹⁸ Tim Berners-Lee, James Hendler, and Ora Lassila, “The Semantic Web,” *Scientific American*, May 2001, <http://www.scientificamerican.com/article/the-semantic-web/>.

panCoast Pizza - Walnut Creek, CA | Yelp
 www.yelp.com > Restaurants > Pizza > Yelp, Inc. >
 ★★★★★ Rating: 4 - 134 reviews - Price range: \$\$
 134 Reviews of panCoast Pizza "Great pizza!! They know what they're doing, for sure. Hand stretched dough, fresh toppings, and baked to a crispy finish."

Figure 6-49. Rich snippets from Google SERPs

Tim Berners-Lee

Computer Scientist

Sir Timothy John "Tim" Berners-Lee, OM, KBE, FRS, FREng, FRSA, DFBCS, also known as "TimBL", is a British computer scientist, best known as the inventor of the World Wide Web. [Wikipedia](#)

Born: June 8, 1955 (age 59), London, United Kingdom

Nationality: British

Parents: [Mary Lee Woods](#), [Conway Berners-Lee](#)

Awards: MacArthur Fellowship, Marconi Prize, Charles Stark Draper Prize, Mountbatten Medal, President's Medal

Books: [Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor](#)


Education: [The Queen's College, Oxford \(1973–1976\)](#), [Emanuel School \(1969–1973\)](#)

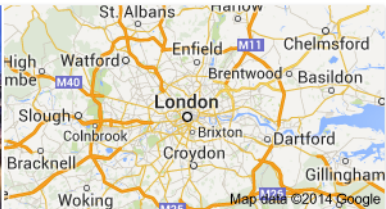


Figure 6-50. Google SERPs knowledge panel on Tim Berners-Lee

Web
News
Shopping
Images
Videos
More ▾
Search tools

About 892,000 results (0.37 seconds)





London, United Kingdom

Tim Berners-Lee, Place of birth

Figure 6-51. Google answer box for Tim Berners-Lee query

Before Schema.org, semantic markup was largely the province of academia, research and development, specialty niche businesses, and others with specific requirements to exchange and understand data in a deeply meaningful way. With Schema.org, the local pizza joint can hope to have “5 star reviews” jump off the search results page; local governments can publicize civic events and have that information re-presented in the SERPs, providing “instant answers” to searchers; and the list goes on. With such practical benefits in mind, and with the simplified approach of Schema.org over its big brothers like RDFa, many people responsible for building web pages are making the effort to incorporate this markup into their sites.

Overview

Schema.org markup communicates the meaning of web pages to computer programs that read them, like search engine spiders. While humans can often infer the meaning of words on a page through a number of contextual clues, computer programs often need help to extract such meaning. Let’s walk through a simple example. Imagine you have a page that displays information about the book *20,000 Leagues Under the Sea*. You might create such a page with the following HTML code:

```
<div id="book">
<h3>20,000 Leagues Under the Sea</h2>

<h3>Author: Jules Verne</h3>
<h3>Rating: 5 stars, based on 1374 reviews</h3>
<h3>ISBN: 978-1-904808-28-2</h3>
</div> +
```

After being marked up, the source code might look like [Figure 6-52](#). The Schema.org microdata markup is highlighted, and explained after the figure.



Figure 6-52. Annotated Schema.org markup

Line 1: itemscope

Adding this to a container element, in this case a `<div>`, is the way to begin defining an entity. This attribute makes the `<div>` element the outermost, enclosing type definition for the entire book entity. The `itemtype=http://schema.org/Book`

attribute, also added to the `<div>` element, declares the type of this entity. Together, this makes the entire `<div>` a container for a single book type entity.

Line 2: `itemprop="name"`

Adding `itemprop` to an HTML element defines it as the container for a property. In this case, the property is the name of the book, and the value is the inner text of the `<h3>` tags, `20,000 Leagues Under the Sea`.

Line 3: `itemprop="image"`

Similar to the name `itemprop`, but the value of this property is the URL referenced in the `src` attribute of the `` tag.

Line 4

Compare this to line 2. In line 2, the inner text of the `h3` element was our exact title. Here, we also have a label ("Author:"), which is not part of the actual author property. To keep our browser display looking the same as the original but omit the "Author:" label from our author property, we use this construct.

Lines 5 and 6

Our item property in this case is not a simple text string or URL, but rather itself another item—a `schema.org/AggregateRating`. It is simultaneously a property of the book (so it uses the `itemprop` attribute) as well as a type itself (so it uses `item` scope and `itemtype`, as we saw in line 1 for our outermost book type).

Lines 7 and 8

These lines add properties for the `aggregateRating`, in much the same way we defined `name` and `author` in lines 2 and 4. Note the careful enclosure of the data with `` tags so as to include only the data itself, not the surrounding labels, in our property. This is the same technique we used in line 4.

Lines 9 and 10

These `itemprops` contain information needed to provide a context for the item rating (namely, that our scale is 0 to 5, with 0 being worst and 5 being best), but which is not displayed on the page. In the previous examples, the values of the properties came from the inner text of the HTML. In this case, there is no text to display in the browser, so we use the `value` attribute on the `` element.

Line 12

This code defines the ISBN property with an `itemprop`, again using a `` element to keep the display and the data cleanly separated.

Schema.org is a simple idea, with a mostly simple implementation, but getting it right can be tricky. Fortunately, the operating environment is pretty forgiving, and there are a few tools that help ease the task. Search engines understand that most webmasters aren't structured data gurus with a deep understanding of ontologies and advanced

notions of relations, entities, and other such concepts. Thus, they are generally quite adept at figuring out what you mean by your Schema.org markup, even if there are errors or ambiguities in how you say it. Clearly you should strive to be accurate, but you should approach this exercise knowing that you don't have to understand every single nuance of structured data markup, or strive for perfection in order to succeed.

How to Use Schema.org

Let's first talk about the best way to approach using Schema.org. Semantic markup is designed to help you provide meaning and clarity about what your website and each web page on it are about, so you should be clear about this before attempting to implement Schema. Think real-world tangible objects, or in semantic markup parlance, *entities*.

For example, if you're a purveyor of fine linen, your site may have lots of pages related to pillowcases, bed sheets, duvet covers, and so on. Your pages are "about" these entities. If you're willing to make the common conceptual leap here, you could say these entities "live on" your web pages. Job one is to figure out how to map these entities to Schema.org's catalog of "types."

At this level of thinking, Schema.org is a large and ever-growing and evolving catalog of "types" (Schema.org documentation sometimes uses the word *items* in place of *types* here) that attempts to classify everything that can be represented on web pages. Let's take a look at the Schema.org page for a Book type, shown in [Figure 6-53](#). The idea is straightforward. The type definition identifies the key attributes that you would use to uniquely describe an "instance" (that is, a single, real-world example) of this type.

schema.org Search

Home Schemas Documentation

Thing > CreativeWork > Book

A book.

Property	Expected Type	Description
Properties from Book		
bookEdition	Text	The edition of the book.
bookFormat	BookFormatType	The format of the book.
illustrator	Person	The illustrator of the book.
isbn	Text	The ISBN of the book.
numberOfPages	Integer	The number of pages in the book.
Properties from CreativeWork		
about	Thing	The subject matter of the content.
accessibilityAPI	Text	Indicates that the resource is compatible with the referenced accessibility API (WebSchemas wiki lists possible values).
accessibilityControl	Text	Identifies input methods that are sufficient to fully control the described resource (WebSchemas wiki lists possible values).
accessibilityFeature	Text	Content features of the resource, such as accessible media, alternatives and supported enhancements for accessibility (WebSchemas wiki lists possible values).
accessibilityHazard	Text	A characteristic of the described resource that is physiologically dangerous to some users. Related to WCAG 2.0 guideline 2.3. (WebSchemas wiki lists possible values)
accountablePerson	Person	Specifies the Person that is legally accountable for the CreativeWork.
aggregateRating	AggregateRating	The overall rating, based on a collection of reviews or ratings, of the item.
alternativeHeadline	Text	A secondary title of the CreativeWork.
associatedMedia	MediaObject	The media objects that encode this creative work. This property is a synonym for encodings .
audience	Audience	The intended audience of the item, i.e. the group for whom the item was created.
audio	AudioObject	An embedded audio object.
author	Person or Organization	The author of this content. Please note that author is special in that HTML 5 provides a special mechanism for indicating authorship via the rel tag . That is equivalent to this and may be used interchangeably.
award	Text	An award won by this person or for this creative work. Supercedes awards .
citation	Text or CreativeWork	A citation or reference to another creative work, such as another publication, web page, scholarly article, etc.
comment	Comment or UserComments	Comments, typically from users, on this CreativeWork.
commentCount	Integer	The number of comments this CreativeWork (e.g. Article, Question or Answer) has received. This is most applicable to works published in Web sites with commenting system; additional comments may exist elsewhere.

Figure 6-53. Schema.org definition of Book

It may help if you open this page in your web browser as we discuss it. Note that the Schema.org definitions are frequently reviewed and updated based on active user feedback, so you may even see minor variations on the current page. But the overall structure will likely remain very similar, and the major elements of the page are central to Schema.org. First, note the simple description, confirming that this is, indeed, the model for a book. Let's ignore the Thing > CreativeWork > Book breadcrumb for now; we'll come back to that later.

Next comes a table of *properties*—what we might think of as the attributes that uniquely describe our individual entity—which, in this example, are the things that describe the book *20,000 Leagues Under the Sea*. Each property has a name (the Property column), an Expected Type, and a Description. The Expected Type tells us whether this property is simply a text value (like a name), or something more complex—that is, a type itself. For example, the *illustrator* property should contain not the name of the

illustrator, but a full person entity, using the <http://schema.org/Person> type definition (which, as you would expect, itself contains a `name` property, and that's where you include the illustrator's name).

As you begin examining a possible mapping of your entities to Schema.org types, you'll often encounter this nesting of types within types. While many of the properties of an entity are simple descriptions (text strings like "blue", "extra large", or even "Jan 17, 2015"), others are more complex and entities in their own right. This is the concept of composing larger-scale things from a collection of smaller ones, as in describing a car (a single entity in its own right) as being made up of an engine, a chassis, wheels, interior trim, and so on (all entities themselves).

Extending this idea further: to an auto mechanic, an engine—a component of our car—is itself the composite thing (the big entity). To understand the engine in more detail, it's important to break it down into its own component entities, like carburetors, spark plugs, filters, and so forth.

Schema.org, then, is a set of conventions for modeling complex things in the real world, and marking them up in a way that search engines can consume, leading them to a deeper understanding of web pages. This deeper understanding in turn leads to many current and future benefits when the search engines subsequently present that data back to users in compelling, contextually relevant ways.

There's one more preliminary concept we should cover; it seems complicated at first but isn't once we break it down. One thing you'll notice as you browse Schema.org's types is that each one lives within a hierarchical family tree. We saw this earlier with the breadcrumb on the Books page, shown again in [Figure 6-54](#).

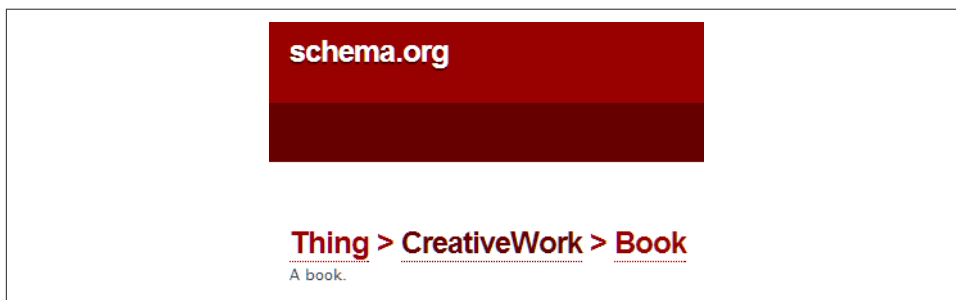


Figure 6-54. Showing the “inheritance” hierarchy for the *Book* type

It's important to note that this kind of hierarchy, referred to among computer scientists as *inheritance*, is different than the composition hierarchy (a car made up of an engine) example we discussed earlier. The Schema.org type hierarchy is a way of categorizing things from most generic to most specific—what we call an *ontology*. Its form is similar to the well-known animal kingdom charts we've all seen, or the myriad other classifi-

cation schemes we all tend to take for granted—often represented on web pages with features like breadcrumbs, navigation menus, and faceted navigation filters.

The key point to remember here is that when choosing the Schema.org type to model your entities, it's always best to choose the most specific type you can. That is, choose `Restaurant` over `LocalBusiness` (if, indeed, you're operating a restaurant!). Choose `Book` over `CreativeWork` for books, and `HighSchool` over `EducationalOrganization` for high schools. Doing so ensures you are giving the most specific information possible to the search engines, rather than settling for generic descriptions.

With that background covered, let's run through the general plan for adding Schema.org markup to your website. Here are the six major steps:

1. Determine the Schema.org types that best describe the entities represented on your web pages, which may be different for each of your different page archetypes.
2. For each page archetype you're modeling, perform a detailed mapping of the information elements displayed on the page to the Schema.org type properties.
3. Choose the approach you will use to express the Schema.org markup.
4. Edit the HTML document templates, or update the CMS settings, or modify the scripts—whatever best describes how your pages are generated—to incorporate the Schema.org markup.
5. Test the markup to see if your syntax is accurate, and if you've properly modeled complex entities.
6. Monitor how well the search engines are consuming your structured data, and whether and how that data is being presented in the SERPs.

Let's take these one at a time in more detail.

Step 1: Determine Schema.org types

In this step, you think carefully about which web pages to mark up while simultaneously browsing the Schema.org website (actually, the browsing capability is fairly limited as of the time of this writing, so you might be better off searching for types; see [Figure 6-55](#)).

For example, if your website is about community theater groups, and displays one page for each theater group along with the upcoming list of their performances, you would begin by searching at <http://schema.org> for something like *theater*. The resulting page looks like what's shown in [Figure 6-56](#). Scanning the results, we quickly spot `TheaterGroup` as a likely candidate for the type of our main entities.

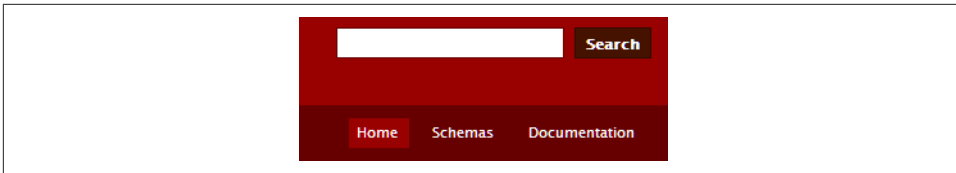


Figure 6-55. Schema.org includes a search box at the top of each page of on the site

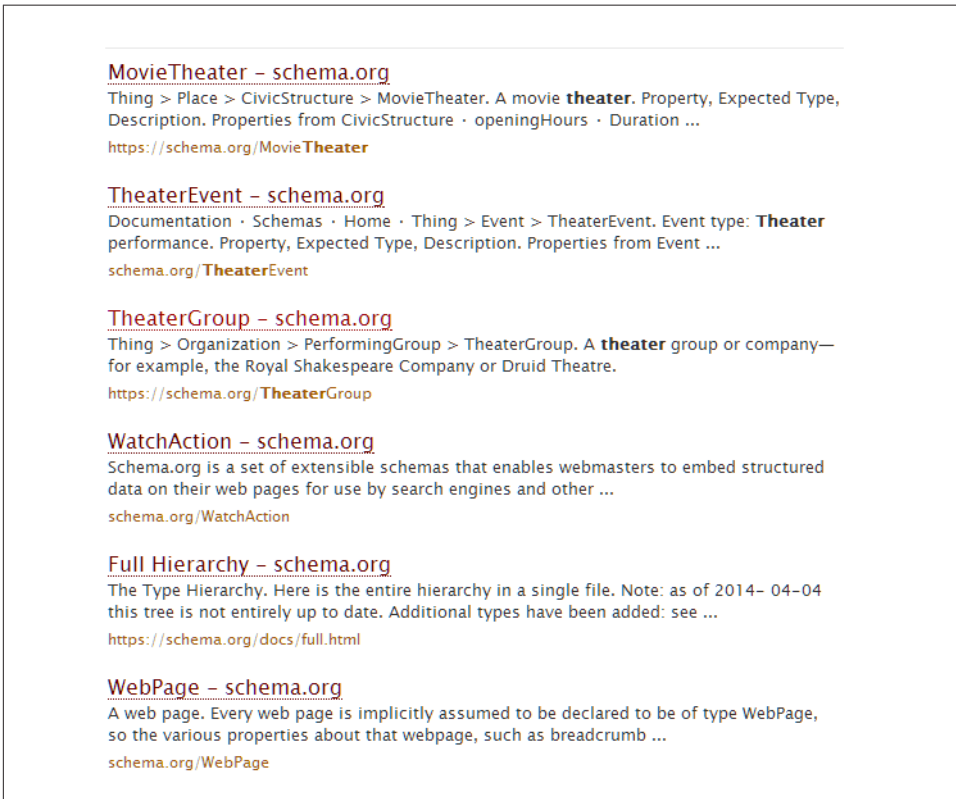


Figure 6-56. Search results for “theater” on *Schema.org*

Taking a closer look at the TheaterGroup page at <http://schema.org/TheaterGroup> (Figure 6-57), we can see a few of our core concepts at work:

- TheaterGroup is part of a logical hierarchy, starting with the most generic type (Thing—actually the topmost ancestor of all Schema.org types), then proceeding to more and more refined types: Organization, PerformingGroup, TheaterGroup.
- A TheaterGroup is composed of many elements (called properties), some of them simple like the name of the group, and some of them actual types in their own right (such as address, aggregateRating, employee, etc.). Examining the list of prop-

erties confirms our belief that this is the best type for describing our local theater entities on our web pages.

Property	Expected Type	Description
Properties from Organization		
address	PostalAddress	Physical address of the item.
aggregateRating	AggregateRating	The overall rating, based on a collection of reviews or ratings, of the item.
brand	Brand or Organization	The brand(s) associated with a product or service, or the brand(s) maintained by an organization or business person.
contactPoint	ContactPoint	A contact point for a person or organization. Supercedes contactPoints .
department	Organization	A relationship between an organization and a department of that organization, also described as an organization (allowing different urls, logos, opening hours). For example: a store with a pharmacy, or a bakery with a cafe.
dissolutionDate	Date	The date that this organization was dissolved.
duns	Text	The Dun & Bradstreet DUNS number for identifying an organization or business person.
email	Text	Email address.
employee	Person	Someone working for this organization. Supercedes employees .
event	Event	Upcoming or past event associated with this place or organization. Supercedes events .
faxNumber	Text	The fax number.
founder	Person	A person who founded this organization. Supercedes founders .
foundingDate	Date	The date that this organization was founded.
globalLocationNumber	Text	The Global Location Number (GLN, sometimes also referred to as International Location Number or ILN) of the respective organization, person, or place. The GLN is a 13-digit number used to identify parties and physical locations.

Figure 6-57. TheaterGroup type from Schema.org

It’s during this step that you want to deal with the question “What is this page about?” and choose the Schema.org type that best describes the overall contents of the page. Often this choice is obvious, but at times it can be tricky. For example, on a page with a product for sale, should you choose *schema.org/Offer* or *schema.org/Product* to model the page?

Examining both pages on the Schema.org site, you can see that an Offer has a property called `itemOffered`, with an expected value of Product. This means that you can describe the contents of the page as an Offer (Schema.org’s concept of something for sale), where the item for sale (the Product) is contained within the Offer, using the `itemOffered` property.

Alternatively, you could use the Product type, which has a property called `offers` that can, as you might expect, contain one or more Offer types. The decision probably depends on the overall purpose of the page. If the page is a detailed product page, describing many attributes of the product, and the offer information is just a piece of that, it probably makes sense to model the page as a Product and include the offer

information in the `itemOffered` property. However, it's not out of the question that you could invert this model.

Either of the approaches to the `Product/Offer` model is valid, as both convey the meaning that you want. But take another look at `Offer`. You can see that it is a complex concept, and has many properties that are themselves types (for example, `aggregateRating`). Other complex nesting of types and attributes can easily arise, and it's important to model these out in a way that best matches the meaning of the page. The best approach often won't be obvious at this stage of analysis, so you may need to revisit your thinking after you complete step 2 of the process.

Step 2: Map Schema.org properties to elements on the web page

The first step here is to survey the various data elements displayed on the web page, and match them up with the appropriate Schema.org types and properties you selected in step 1. In this step, you may discover relationships that resolve some of the potential ambiguities from step 1.

For example, continuing the `Product/Offer` discussion, let's assume that one of the items displayed on the page is an overall rating—say a value on a scale of 1 to 5—representing user evaluations of the product. We notice that both `Product` and `Offer` have a property called `aggregateRating`, so this hasn't quite settled our debate on which type to model the page on.

Let's also assume that we display several different prices—perhaps for new or used versions of the product, or with different shipping options or different currencies. It now starts to become obvious that we should model the entire page as a `Product` that contains multiple `Offers` and a single `aggregateRating` that applies to the `Product` itself. Finally, this is starting to take shape!

You might notice that there are properties defined on the Schema.org type that you're not currently displaying to browsers, but which you have access to. Continuing with our `Product` example, perhaps your web application's database stores the MPN (manufacturer's part number), but you don't choose to display that on the page. What should you do?

Ideally, you want a very high degree of consistency between what you mark up and what's visible to "normal users" via web browsers. Technically, there are mechanisms that allow you to communicate to the search engines metadata about your entities that shouldn't be displayed to users (we saw this earlier in our `aggregateRating` example, and we'll explore that example a bit more momentarily).

However, it's important to use these mechanisms sparingly, and not be tempted to stuff a lot of extra data into the Schema.org markup that is not visible to human users. In the MPN case, our choice should be between adding this as a visible element on the

page (and then of course adding it to our Schema.org markup), or forgoing it entirely. As you think about this, it should become clear that marking up a lot of data that is not displayed to the user is conceptually something a spammer might do, and for that reason the search engines frown on it.

What are the valid reasons for marking up nondisplayed data? Usually it's because you need to convey some different context that is obvious to people, but not to search engine spiders. For example, when you mark up an `aggregateRating`, you're strongly encouraged to specify the scale; that is, if you display 4 stars for a review on a scale of 0 to 5, this is usually quite clear in the visual representation, but it needs to be stated explicitly in the Schema.org markup. Thus, `aggregateRating` entities have `worstRating` and `bestRating` properties, and we want to supply the values 0 and 5, respectively, corresponding to our star rating scale. We saw this in the sample code for our book at the beginning of the chapter.

Upon completing this step, you should have a complete mapping between the data displayed on the page and the various Schema.org types and properties that make up your model. Your model may be simple or complex with multiple levels of nesting. It's best to make all these decisions before you begin actually implementing Schema.org on the page.

Step 3: Choose your implementation technique

For most people, this step means “go mark up the page.” Sounds simple, right? And for some pages, especially those that are template-driven with mostly static data, it should be fairly straightforward. Or, if you're lucky enough to be using a content management system or publishing platform that has built-in support for Schema.org, you can do most of the actual implementation by setting a few configuration parameters.

For other, more dynamic sites that generate their pages through a complex pipeline of page generation programs, tweaking things to insert the right tags in the right place can be far more difficult. And for these types of sites, validating that the generated schema is correct is also challenging, as the Schema.org markup may be sporadically injected into kilobytes of code.

The primary implementation technique is to edit templates and/or modify page generation programs to insert the microdata markup as needed to produce the desired final output. The key thing is to have a clear mapping of the model from step 2 showing the final desired HTML with microdata markup inserted, and use this to validate that the final page produced by the web server matches the model. As we'll see in step 5, there are some tools that can help with this verification as well.

For those who don't have access to the code or backend systems, or who want a simpler approach, Google offers the Structured Data Markup Helper, as part of Google

Search Console. This is a proprietary Google tool that allows you to annotate a page, using a point-and-click editor (see [Figure 6-58](#)). It's actually just an alternative way of providing the same data you provide via Schema.org microdata markup, but you are instead feeding it directly to Google and do not change the page source code at all.

So why doesn't everyone just do this? There are two good reasons why this isn't the best fit for everyone. First, the information is available only to Google, not to other Schema.org-aware search engines (or other applications that may make use of Schema.org markup). Second, as is often the case with this kind of tool, the visual editor is more limited than the markup syntax in its ability to express rich and complex information.

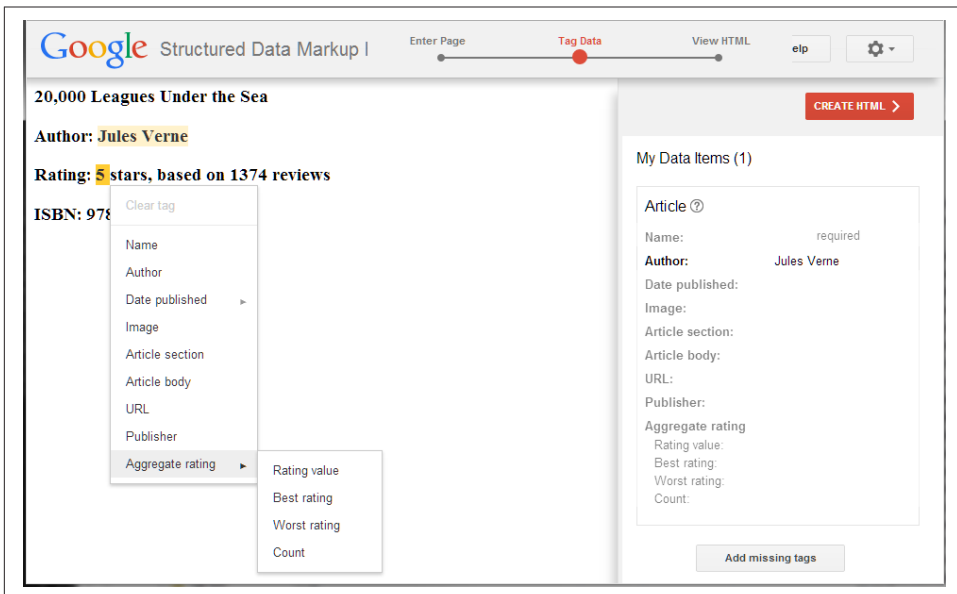


Figure 6-58. *Google Structured Data Markup Helper*

Looking to the future, another alternative may be on the horizon, and is something to keep an eye on. Google and others are already beginning to make use of a format known as JSON-LD for expressing Schema.org markup. This format is showing up in limited, specialized circumstances, but it seems apparent that JSON-LD may soon become a full-fledged alternative to microdata for all Schema.org expression.

The beauty of JSON-LD is that it provides a way to isolate all of the Schema.org information into a single string of code, rather than expressing it by embedding markup within the HTML document itself. This has the possibility of solving many of the more complex issues associated with implementing Schema.org on complex, dynamic sites.

Step 4: Implement the changes to generate the target Schema.org code

This step is really just saying, “Now it’s time for your web developers to go breathe some life into your creation”; that is, go get these pages served up by your web server! This is where the content management system is tweaked, the templates are updated, the page production programs are modified, and so on.

Step 5: Test

When you reach this stage, your web server is shooting out bundles of HTML with tidy little microdata tags embedded in it that add meaning and structure to your data. At this point, the generated Schema.org markup code should be syntactically correct, and should express the right model—that is, the whole composition of smaller properties and types into larger properties and types needed to accurately model the information displayed on our pages. Of course it’s important to verify this.

The hard way to do that is to examine the generated code by hand, looking for the opening and closing tags, and ensuring that all the data is there, nested properly. Fortunately, there’s an easier way (though you should still be prepared to roll up your sleeves and dig into the code to debug potential problems).

The easier way is to use one or more of the tools available to verify your Schema.org microdata markup. Perhaps the best known of these tools is [Google’s Structured Data Testing Tool](#), which is an elegant utility that examines your page (either directly by supplying a URL, or alternatively by cutting/pasting HTML source code) and gives you feedback on the structured data it finds. [Figure 6-59](#) shows such a result.

```
1 <div itemscope itemtype="http://schema.org/Restaurant">
2 <h1 itemprop="name">Fondue for Fun and Fantasy</h1>
3 <p itemprop="description">Fantastic and fun for all your cheesy
  occasions.</p>
4 <p>Open: <time itemprop="openingHours"
  datetimes="Mo,Tu,We,Th,Fr,Sa,Su 11:30-23:00">Daily from 11:30am till
  11pm</time></p>
5 <p>Phone: <span itemprop="telephone" content="+155501003333">555-
  0100-3333</span></p>
6 <p>View <a itemprop="menu" href="http://example.com/menu">our
  menu</a>.</p>
7 </div>
```

Restaurant	
name:	Fondue for Fun and Fantasy
description:	Fantastic and fun for all your cheesy occasions.
openingHours:	Mo,Tu,We,Th,Fr,Sa,Su 11:30-23:00
telephone:	+155501003333
menu:	http://example.com/menu

Figure 6-59. Google’s Structured Data Testing Tool output

The output of this tool has a bit of an arcane formatting convention. Note that our book shows up as the first item. The book has a number of properties, among which is

our `aggregateRating`; recall that this is itself another type. When this composition or nesting occurs properly, you see the output shown in [Figure 6-59](#).

Here, the nesting relationship is shown by Item 1 in the item value field for the `aggregateRating` property of the `Book`, followed immediately by the Item 1 output. So the value Item 1 in the first field ties together with the name of the second item shown, and shows that Item 1—the rating—is properly contained within the book entity, as specified by the Schema.org definition for a `Book`. Google will keep incrementing these numbers for as many embedded types as you have on the page.

Summary

We have seen that Schema.org is a standard for providing search engines (and potentially other applications) with structured data describing the meaning of website content. The notion of data structuring is actually quite intuitive, and maps well to the way we commonly categorize things like product catalogs, biology, library card catalogs, and many other collections of related items. This intuitive, webmaster-friendly approach has led to rapid adoption of Schema.org by the webmaster and content production communities. Currently, the most common way to structure data with Schema.org is to add microdata markup to HTML documents. Search engines use this data to extract meaning, and enrich SERPs with rich snippets, answer boxes, and knowledge panels, providing a more relevant and deeper search result. Implementing Schema.org can bring these benefits to both users and publishers today, and can help set the stage for publishers to gradually delve more deeply into the emerging world of semantic search in the coming years.

NOTE

A special thanks to John Biundo for his contributions to the Schema.org portion of this chapter.

Google Authorship and Author Authority

One of the most interesting insights into the mind of Google, as it were, was the three-year experiment known as Google Authorship.

Google Authorship was a program that allowed online authors to identify and verify their content with Google. This was accomplished by a two-way link between the author's content across the Web and his Google+ profile.

While the Authorship program was officially discontinued by Google on August 28, 2014, it is likely that Google's interest in the value of the authority, trust, and reputation of an author in a given topical area is undiminished.

A Brief History of Google Authorship

The roots of Google Authorship lie in a patent originally granted to Google in 2007, called **Agent Rank**. The patent described methods whereby a search engine could identify distinct “agents” (one of which could be the author or authors of a web document) and assign a score to each agent that could then be used as a factor in search rankings.

Google didn’t appear to do anything with this patent until June 2011, when Google’s Othar Hansson announced in a blog post that it would begin to support the use of the HTML5 standard `rel="author"` and the XFN standard `rel="me"`, and that webmasters could use that markup to identify authors and author profiles on their sites.¹⁹

The next major step in Authorship came just 21 days later, when Google unveiled its new social network, Google+. Google+ provided personal profiles that Google could use to verify authors using the `rel="author"` markup.

This intention was confirmed in a YouTube video by Othar Hansson and Matt Cutts published on August 9, 2011, titled “**Authorship Markup**”. In the video, Hansson and Cutts explained that Google wanted web authors to have Google+ profiles, and that they should link from the “Contributor To” link sections of those profiles to each domain where they publish content. Over time, Google offered several options by which the publisher could confirm the relationship by linking back to the author’s Google+ profile.

In that video, Google confirmed that there could be rewards to authors who implemented Authorship markup; the immediate possible benefit was the potential for an author’s profile image and byline to be shown with search results for her content.

Figure 6-60 is typical of such results.



Figure 6-60. Rich snippet authorship result

Additional potential benefits mentioned by Hansson and Cutts were increased search rankings and the fact that Google might be able to use Authorship to identify the original author or a piece of web content, thus giving that author’s copy precedence in search over scraped copies.

¹⁹ Othar Hansson, “Authorship Markup and Web Search,” Webmaster Central Blog, June 7, 2011, <http://googlewebmastercentral.blogspot.com/2011/06/authorship-markup-and-web-search.html>.

Over time, Google added several tools and features to make Authorship easier to implement and more useful for authors and publishers. This was probably the result of the problems the company saw with a lack of adoption of this markup.

The first major hint that Google might be pulling back on its Authorship experiment came in October 2013 when AJ Kohn revealed that Othar Hansson had left the Authorship team and was not being replaced.²⁰ In that same month, Matt Cutts revealed that Google would soon be cutting back on the amount of Authorship rich snippets shown in search, as it had shown in tests that doing so improved the quality of those results.

Cutts's words proved true in December 2013, when observers noticed a 15% reduction in the amount of author photos being shown for most queries.²¹

In June 2014 Authorship was further reduced in search as Google announced that it would no longer show author photos in results, just bylines. The only announced reason for this was to bring its mobile and desktop user experiences more into sync.

However, only two months later, as previously noted, Google announced the complete removal of Authorship data from search, and stated that it would no longer be tracking any data from `rel="author"` links. The Google Authorship program, or at least any program based on `rel="author"` links and showing rich snippets in search results, was now over.

Why Did Google End Support for `rel="author"`?

In his official announcement of the end of the Authorship program, John Mueller of Google Webmaster Central said, "Unfortunately, we've also observed that [Authorship] information isn't as useful to our users as we'd hoped, and can even distract from those results. With this in mind, we've made the difficult decision to stop showing authorship in search results."

He went on to elaborate, saying that this decision was based on user experience concerns. After three years of testing, Google was no longer seeing any particular user benefits from showing Authorship results. Mueller said that removing the Authorship results "did not seem to reduce traffic to sites." It would seem, then, that searchers were no longer viewing these results as anything special.

20 AJ Kohn, "Authorship Is Dead, Long Live Authorship," Blind Five Year Old, October 24, 2013, <http://www.blindfiveyearold.com/authorship-is-dead-long-live-authorship>.

21 Barry Schwartz, "Confirmed: Google Reduces Authorship Rich Snippets in Search Results," Search Engine Land, December 19, 2013, <http://searchengineland.com/confirmed-google-reduces-authorship-rich-snippets-in-search-results-180313>.

What else may have factored into the decision to stop showing Authorship results? In his post Mueller mentioned that he knew that Authorship “wasn’t always easy to implement.” Could it be that low implementation rates by most sites fed Google’s decision? If Google were ever going to rely on Authorship as a signal for search, it would need to have data from a wide variety of sites.

In a study published just after the ending of Authorship, Eric Enge confirmed from a sampling of 150 top publishing sites that Authorship implementation was indeed low.²² He found that 72% of these sites had attempted Authorship markup in some way, but out of those *nearly three-fourths had errors in their implementation*. But even worse, 71% of the 500 authors sampled from those sites had done nothing from their side to implement Authorship.

It would seem that low participation might be another reason behind Google’s decision. Google may have learned that data you want to use as a ranking factor can’t be dependent upon voluntary actions by webmasters and authors.

Is Author Authority Dead for Google?

Does the end of `rel="author"`-based Authorship mean Google has lost all interest in understanding, tracking, and making use of data concerning the authority levels of online authors? Most likely not.

For one thing, on September 2, 2014, Google was granted a patent for a system that would retrieve, rank, and display in search authors considered authoritative for a topic based on their relationship (in social networks) to the searcher.²³

Also, Google spokesperson Matt Cutts often spoke during the last year of Google Authorship about his interest in and support for Google eventually being able to use author reputation as a means of surfacing worthwhile content in search results, but noted that he sees it as a long-term project.²⁴ While Cutts seems to be voicing his personal opinion in such statements, it is doubtful that he would speak so frequently and positively about the topic if it weren’t actually active at Google.

Another area that seems to support the notion that Google will only increase its interest in author authority is semantic search. Semantic search involves, in part, a dependence upon the identification of various entities and the ability to understand and eval-

²² Eric Enge, “Authorship Adoption Fail – Detailed Stats,” Stone Temple Consulting, September 9, 2014, <https://www.stonetemple.com/authorship-adoption-fail-detailed-stats/>.

²³ Bill Slawski, “Has Google Decided That You Are Authoritative for a Query?,” SEO by the Sea, September 7, 2014, <http://www.seobythesea.com/2014/09/google-decided-authoritative-query/>.

²⁴ Mark Traphagen, “Does Google Use Facebook & Twitter as Ranking Signals? Matt Cutts Answers,” Stone Temple Consulting, January 23, 2014, <https://www.stonetemple.com/googles-matt-cutts-understanding-social-identity-on-the-web-is-hard/>.

uate the relationships between them. As the original Agent Rank patent makes clear, authors of web content are certainly a useful type of entity.

Google understands that real people often evaluate authority and trustworthiness not just by a document's contents or what links to it, but by the reputation of the author. Semantic search at its simplest is a quest to enable Google's search algorithm to evaluate the world more closely to the way people do. So it makes sense that Google would continue to pursue the ability to evaluate and rank authors by the trust and authority real people place in them for a given topic.

Google+ Authors in Personalized Search

At the time of this writing there remained one large and very interesting exception to Google's elimination of Authorship rich snippets from search. Author photos and bylines can still appear for Google+ content authored by people in a searcher's Google network (Google+ circles and Gmail contacts) when that searcher is logged in to his Google+ account while searching. Figure 6-61 shows such a result.



Figure 6-61. Personalized search rich author snippet

Notice that the URLs are both from plus.google.com (Google+). The person performing this search has both Ana Hoffman and Mark Traphagen in his Google+ circles, and is searching while logged in to his Google+ account. For the search query "Google authorrank" Google found that these two people in the searcher's network had relevant content on Google+ and so included it in the searcher's results.

Note two points about these results:

- These results are uniquely ranked higher for this individual searcher. If he searches for the same query while logged out of his Google+ account, these results will not show up in the top results.
- The personal connection of the authors to the searcher is being emphasized by the photo and byline.

The Future of Author Authority at Google

It appears that Google remains interested in the concept of author authority as a factor in search rankings. Google is likely working on methods to identify and evaluate

authors and their content that are not dependent on human publishers and authors placing links and attribution tags. When those methods are providing reliable data, Google might make these signals a ranking factor.

However, given the lessons of the first Google Authorship experiment, we might expect the following possible differences:

Author authority might be more personalized.

That is, Google may give a greater boost to content by authoritative authors relevant to your search *if* you have some connection to or relationship with those authors.

Author authority in search will probably be less obvious.

Google may not return to the practice of displaying rich snippet profile photos for top authors, in part because it is moving away from flashier rich snippets in general as part of its Mobile First initiative. It is therefore likely that any future author authority factor will simply be folded into the many factors that determine search rankings and may not be apparent to the searcher.

Author Authority

Here are some tips on how to build author authority:

Publish with real names.

In order to build author authority search engines have to be able to recognize that multiple pieces of content are connected with a particular individual. Several of the following tips relate to building your personal authority both on and offline, so using your real name with your content is important.

Keep your name consistent.

In parallel with the previous tip, it is important that you use exactly the same name as the byline on all your content as well as in all your social profiles. That will help search engines gain confidence about your identity, and make it more likely that all of your online content will be used to evaluate your authority.

Cross-link your profiles.

Wherever possible, create links between all your online profiles. This is another way to help search engines have more confidence in your unique identity.

Link your social profiles to your content.

Wherever possible, create links from your social and site profiles to the sites on which you publish content. Of course, in the case of Google, it is most important to make sure that all sites on which you publish are linked from the “Contributor to” section of your profile links. Even though Google says it no longer tracks data based on `rel="author"` links to Google+ profiles, we still recommend creating links

from your content or site author profiles back to your Google+ profile, as these still may give Google confidence about content that should be identified with you.

Produce content about all aspects of your field.

More and more we see indications that Google is including in measures of its topical authority how complete and well rounded the content is. It's no longer effective to merely hammer away at certain long-tail keywords. You need to build contextually rich content that looks at your subject from all sides. That doesn't just apply to individual content pieces, but also to the content across an entire site or across your profile as an author on many sites.

Produce content that goes in depth on specifics of your field.

As well as covering all aspects of your area of expertise, your content also needs to explore those areas deeply. That doesn't mean every piece of content needs to be an academic paper, or even long form. But you should be seeking as often as possible to produce content that gives a unique perspective on a topic, or that goes into more depth and detail than most other similar pieces on the Web.

Cultivate an audience.

Every content producer has to be as concerned with building a loyal audience as she is with producing quality content. That means being active on social networks, for one. Seek to build good relationships with those who might be interested in your expertise and likely to share it with their networks.

Participate in relevant conversations.

Go beyond just broadcasting your content to participating in relevant online conversations and communities. Doing that can have multiple benefits. As you contribute to such communities, you get a chance to display your expertise before a broader audience, some of whom may start to follow you. That means you are growing your audience (see above), but doing it in places where you are more likely to pick up followers with high interest in what you do.

Don't forget real-world opportunities.

Attending conferences and networking events in your field can lead to online connections that help reinforce your online authority. This is especially true if you are a speaker or panelist at such events, or get interviewed by a media outlet. You can accelerate this effect by actively inviting people at these events to connect with you online. For example, always place your primary social profiles prominently in any presentations you do.

Incubate and promote brand subject matter experts.

Publishers should not ignore the power of individual topical authority used in conjunction with their brands. Many companies are reluctant to empower individual employees or representatives to build their own authority, but they miss a

real opportunity by not doing so. People identify with, trust, and connect with a real individual long before they do with a faceless brand. Therefore, wise brands will cultivate subject matter experts (SMEs) who have a direct connection with their brand, knowing that the audience and authority those SMEs build will ultimately reflect back on the brand.

Google's Publisher Tag

Although Google has ceased to track author data using `rel="author"` links, the related `rel="publisher"` link tag is still very much supported, and can convey a number of benefits to brands using it.

Similar to Facebook and some other social networks, Google+ allows the creation of pages as distinct from profiles. While profiles are intended solely for individuals, pages allow nonpersonal entities (brands, companies, organizations, bands, etc.) to have a presence on Google+.

Google+ brand pages deserve particular attention, however, because of the way Google uses them, particularly in conjunction with Google search. Stated succinctly, a Google+ page linked from a brand's official site using `rel="publisher"` is the easiest and most direct way for Google to verify the unique identify of a particular brand. Once Google has such verification, it can begin to make use of data related to the brand in various ways.

Brands with verified Google+ pages can be eligible for three special search features:

Knowledge panel with link to Google+ page

As shown in [Figure 6-62](#), when people search for a brand name, brands with verified Google+ pages will show a Google+ logo in their knowledge panel that links to their Google+ page.



Figure 6-62. A search for a brand name brings up the brand's verified Google+ page

The Google+ knowledge panel box shows the brand name and logo, a Google+ follow button (shows only if the searcher is logged in to Google+), the page's follower count, and a recent Google+ post.

Elevated rich snippet Google+ post results

When a searcher has a brand circled on Google+ and searches while logged in to her Google+ account, she may have a relevant Google+ post from that brand elevated to page one and highlighted with a brand logo and brand name rich snippet. An example is shown in **Figure 6-63**.

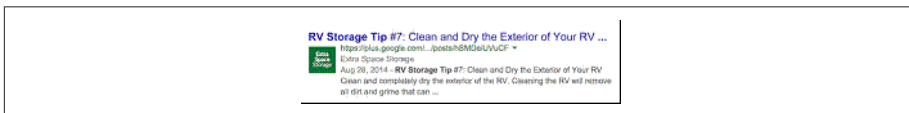


Figure 6-63. Rich snippet authorship result

AdWords Social Extensions

If a brand connects its Google+ Page to its AdWords account and enables **Social Extensions**, Google may add a Google+ annotation to the brand's ads in search. **Figure 6-64** shows how this appears in the search results.

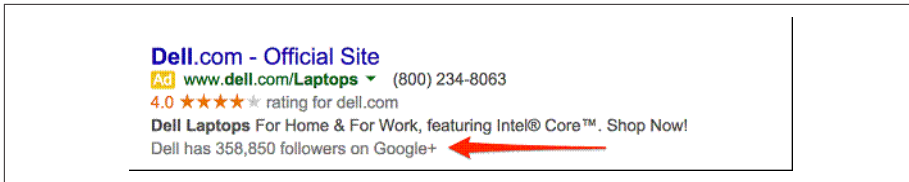


Figure 6-64. AdWords Social Extensions example

Gmail Google+ related pages widget

A brand with a verified brand page that meets certain qualifications can have a widget show in the right sidebar of Gmail when customers open an email from the brand (see [Figure 6-65](#) for an example). The widget displays a thumbnail of a recent post from the brand's Google+ page and, if the user has a Google+ account, a follow button. For details, see <https://support.google.com/business/answer/4569086?hl=en>.

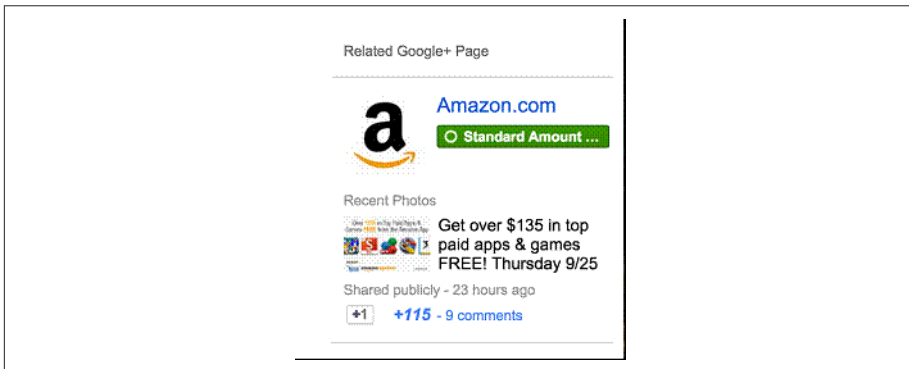


Figure 6-65. Gmail Google+ related pages widget

Verifying a Google+ brand page

For nonlocal business pages, you can verify a page by simply making the brand's official website the main URL of the page, and then linking back from the home page of that site to the Google+ page with a `rel="publisher"` attribute. A local business page must verify via [Google My Business](#). If your business has 10 or more locations to verify, use <https://www.google.com/local/manage/?hl=en#>.

NOTE

A special thanks to [Mark Traphagen](#) for his contributions to the Authorship and publisher tag portions of the chapter.

Google's Knowledge Graph and the Knowledge Vault

The face of search is changing in significant ways. The latest incarnation of that evolution is the Knowledge Graph. Google has also begun to communicate about the Knowledge Vault, though that is just a research project as of October 2014. To get some perspective on why these are important, it is useful to review how search has evolved.

Overview of Changes in Search Complexity

Search engines used to build results by analyzing the text strings they found on the pages of the Web. The resulting pages were quite useful, but the presentation of the results was quite simple, as shown in [Figure 6-66](#).

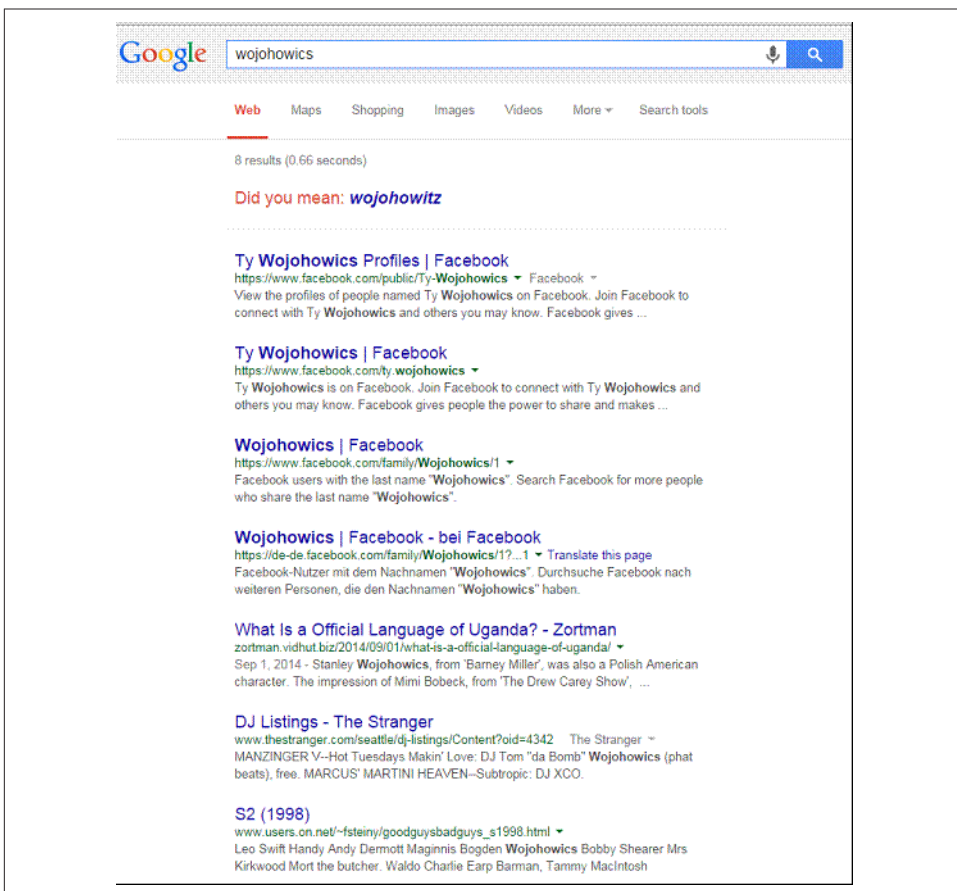


Figure 6-66. Simple text-only search results

Over time, these types of search results became known as “10 blue links” due to their simplicity. As they became more sophisticated, the engines figured out how to incorporate more types of media into the results, including videos, images, news stories, shopping results, and more.

These are typically referred to as “blended results,” and you can see an example in [Figure 6-67](#). You can also read more about these types of results in [Chapter 10](#).

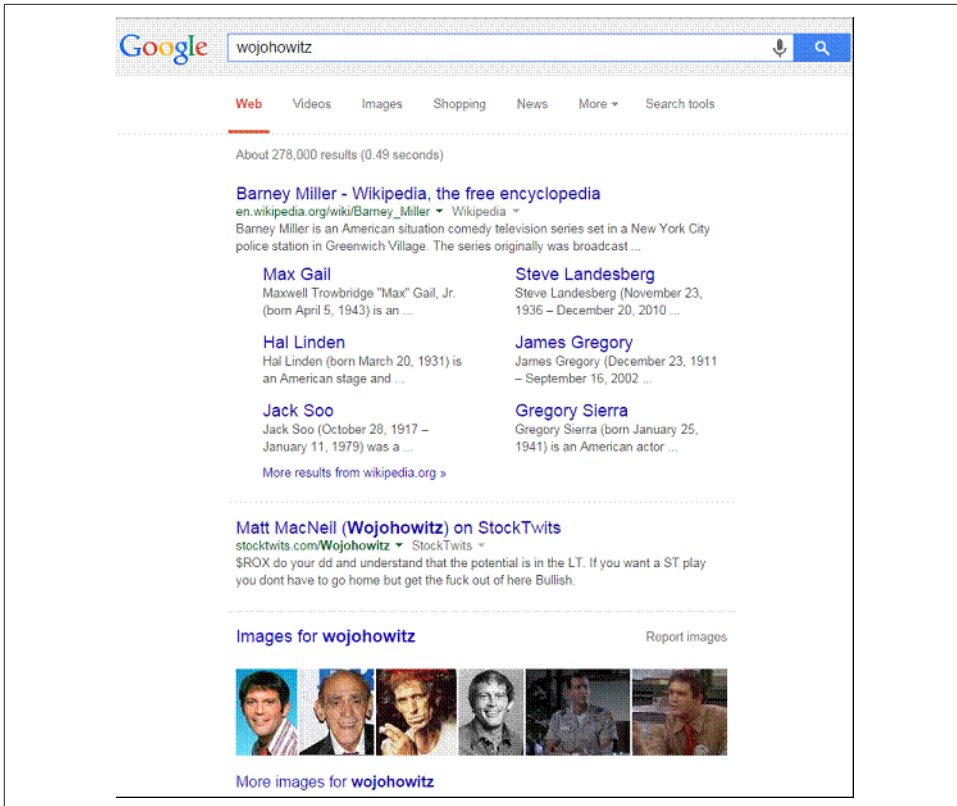


Figure 6-67. Example of blended search results

The emergence of blended search was a big step forward for search engines, but it was only one step on a longer journey. The commitment to that journey is well defined in [Google’s mission statement](#): “Google’s mission is to organize the world’s information and make it universally accessible and useful.”

The next step in that journey for Google was [the Knowledge Graph](#). This was a Google initiative designed to allow it to leverage structured databases to enhance the search results. This initiative allowed Google to further enhance the presentation of its results. An example is shown in [Figure 6-68](#).

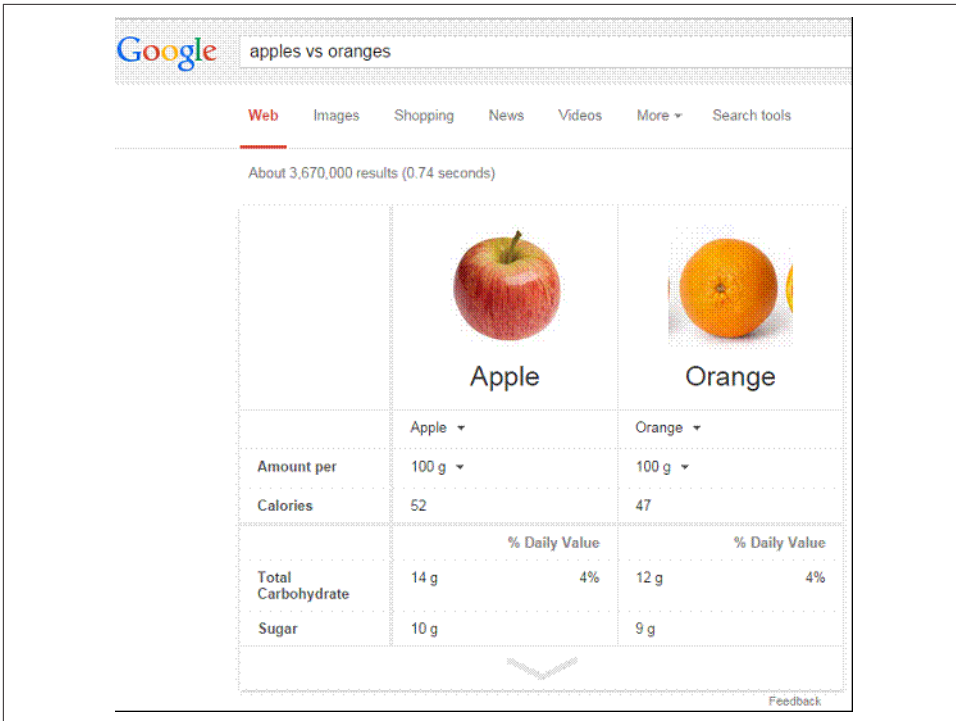


Figure 6-68. Sample Knowledge Graph result

In essence, the Knowledge Graph was another major step forward by Google that allowed it to start showing complete answers in the search results. The information for these results is typically retrieved from **Freebase**, a community-edited database of information.

The basic concept is sometimes referred to as “moving from strings to things.” The search engines that returned nothing but 10 blue links were comparatively quite simple, as they relied on scanning the text on a web page to figure out what it was about, and did not understand relationships.

In comparison, the Knowledge Graph can understand that apples and oranges are both fruits, and they have properties, such as calories, carbohydrate levels, and grams of sugar. Or that the Empire State Building has height, a construction date, and initial architect, and that Google has access to pictures of it.

These types of data sources provide a rich array of information. It is estimated that they allow Google access to information on 500 million entities and 3.5 billion pieces

of information. Stone Temple Consulting performed an extensive study on what types of queries generate Knowledge Graph results, and which don't.²⁵

While 3.5 billion pieces of information may seem like a very large number, in the grand scheme of things, it represents a very small portion of all the types of user queries. As a result, Google is pursuing other avenues to expand its ability to further enhance the information in the search results.

For example, it has started experimenting with extracting information from websites which it in turn has started to use for displaying step-by-step instructions, as shown in [Figure 6-69](#).

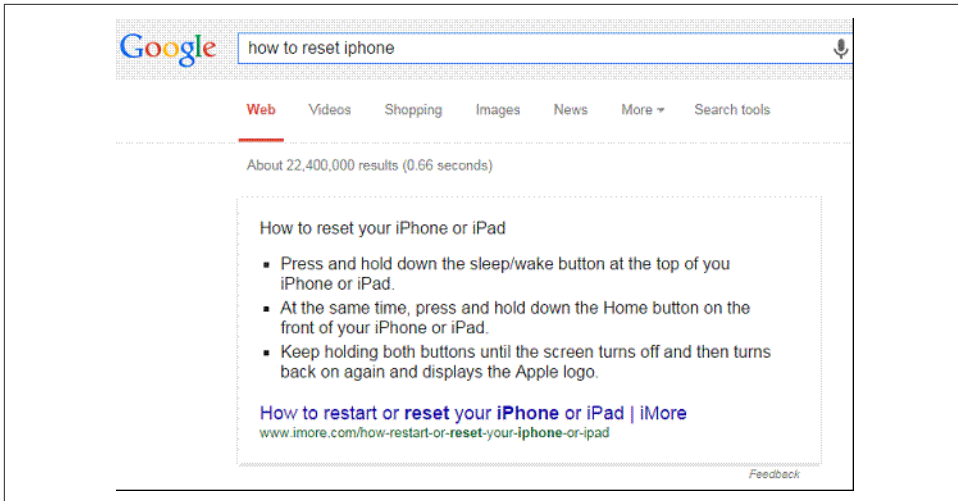


Figure 6-69. Example of step-by-step instructions in Google's SERPs

Barry Schwartz reached out to Google, and got the following response when he asked about the use of step-by-step instructions:²⁶

We started experimenting with this in early June. We hope it draws attention to webpages that provide a useful series of steps to help people complete their task. In these cases we focus attention on the snippet because it's likely to be more helpful for deciding whether the webpage is going to be the most useful for the task.

²⁵ Eric Enge, "The Great Knowledge Box Showdown: Google Now vs. Siri vs. Cortana," Stone Temple Consulting, October 7, 2014. <https://www.stonetemple.com/great-knowledge-box-showdown/>.

²⁶ Barry Schwartz, "Google's Knowledge Graph Is Showing Step By Step Instructions: Here Are Some Examples," Search Engine Land, June 24, 2014, <http://searchengineland.com/googles-knowledge-graph-showing-step-step-instructions-examples-194923>.

Other examples exist where Google is extracting knowledge from websites and showing it in the SERPs. Figure 6-70 shows an example of historical information being found on a website and displayed directly in the results.

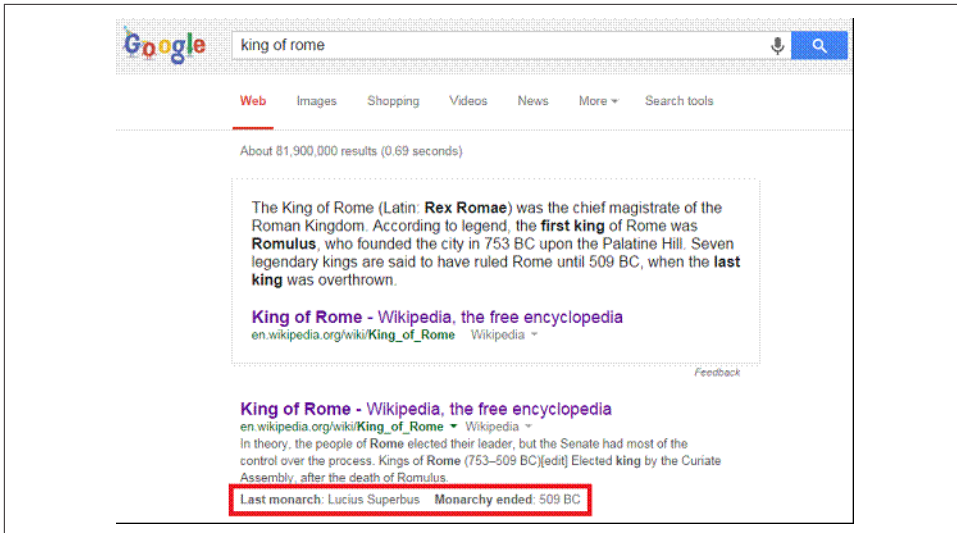


Figure 6-70. Knowledge extraction example

The examples shown in Figure 6-69 and Figure 6-70 are a clear step beyond the simple use of structured data. These represent early examples of what Google refers to as the Knowledge Vault.

Fair Use?

As Google presents more and more of these types of search results, many of the impacted publishers feel that Google is stealing their content and profiting from it. The question becomes whether or not Google's usage can be considered *fair use as defined by the U.S. Copyright Office*. There are four factors involved in determining fair use, as follows:

- The purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes
- The nature of the copyrighted work
- The amount and substantiality of the portion used in relation to the copyrighted work as a whole
- The effect of the use upon the potential market for, or value of, the copyrighted work

There is actually no clear definition of fair use, but it is clear that the substance you take from the third party is a factor. It is common practice among those who quote others, or who attempt to make fair use of someone else's copyrighted material, to provide attribution. However, the U.S. Copyright Office indicates that this might not be enough: "Acknowledging the source of the copyrighted material does not substitute for obtaining permission."

In addition, this is more than a U.S.-only issue, and the laws differ from country to country.

Whether this becomes an issue for Google or not is yet to be determined, but the scale of what it's trying to do makes it likely that it will be subject to legal challenges, and that the way that various legal systems will respond will differ.

One additional aspect to consider is that public domain information is not copyrightable. For example, the fact that Olympia is the capital of the state of Washington is not copyrightable info. If Google is able to extract some common knowledge from third-party sites, it would not be subject to this discussion.

How the Knowledge Vault Works

The first public acknowledgment that Google had a concept it called the Knowledge Vault was in a presentation by Google's Kevin Murphy that took place on October 31, 2013.²⁷ As of October 2014, the Knowledge Vault is just a research project within Google, but it is still useful to learn what this project is about. The core concepts being studied are:

Machine reading

This is the process of extracting facts from a large text corpora. This is similar to methods developed by Carnegie-Mellon, the University of Washington, and others, but Google is working on a much larger-scale version. In addition, it is researching methods for using other prior knowledge to help reduce the error rate.

As information is assembled, it becomes possible to infer, or even determine, other facts. **Figure 6-71** (slide 16 of the presentation) shows an example of this in action. For example, if we know that Barack Obama and Michelle Obama are both parents of Sasha Obama, then we can infer that it is likely that they are married (though that is not necessarily true).

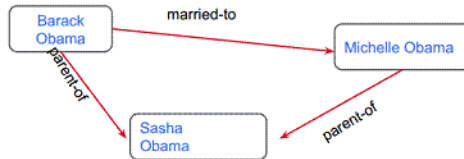
²⁷ Kevin Murphy, "From Big Data to Big Knowledge," October 31, 2013, <http://cikm2013.org/slides/kevin.pdf>.

Predicting facts given prior knowledge



- Perform association rule mining* on Freebase graph, to find noisy rules (features passed to a learned classifier).

$$\forall x, y. \exists z. \text{parent-of}(x, z) \wedge \text{parent-of}(y, z) \Rightarrow \text{married}(x, y)$$



* "Random Walk Inference and Learning in A Large Scale Knowledge Base", Ni Lao et al, 2011

Figure 6-71. Inferring new information

Asking the Web

Web-based question and answers can be used to further supplement the available information. Learning how to ask the right questions (as shown in Figure 6-72) and how to frame those questions is by itself a very difficult process. Verification of the accuracy of the responses is important as well.

The importance of asking the right question



Who is the mother of Frank Zappa

[The Mothers of Invention - Wikipedia, the free encyclopedia](#)
en.wikipedia.org/wiki/The_Mothers_of_Invention
The Mothers of Invention were an American rock band from California that served as the backing musicians for Frank Zappa, a self-taught composer and ...
History · Personnel · Discography · References

[Frank Zappa - Wikipedia, the free encyclopedia](#)
en.wikipedia.org/wiki/Frank_Zappa

Jump to: 1970: Rebirth of The Mothers and filmmaking - [edit] Frank Zappa in Paris, early 1970s. Later in 1970, Zappa formed a new version of The ...
Discography · Moon Zappa · Diva Zappa · Gail Zappa



Who is the mother of Frank Zappa Baltimore Maryland

[Frank Zappa - Wikipedia, the free encyclopedia](#)
en.wikipedia.org/wiki/Frank_Zappa
Frank Vincent Zappa was born in Baltimore, Maryland, on December 21, 1940. His mother, Rose Marie (née Colimore), was of Italian and French ancestry; his ...

Kevin Murphy, CIMM Industry talk, San Francisco, CA, October 31, 2013

Figure 6-72. The importance of asking the right question

Asking people

Freebase is itself an example of this, as it is community edited. Other sources can be used as well. For example, each knowledge panel that Google shows contains a feedback link, allowing it to collect information on accuracy problems (Figure 6-73). This feedback can also be flawed, and Google is investigating algorithms to predict the possibility that the information received is correct.

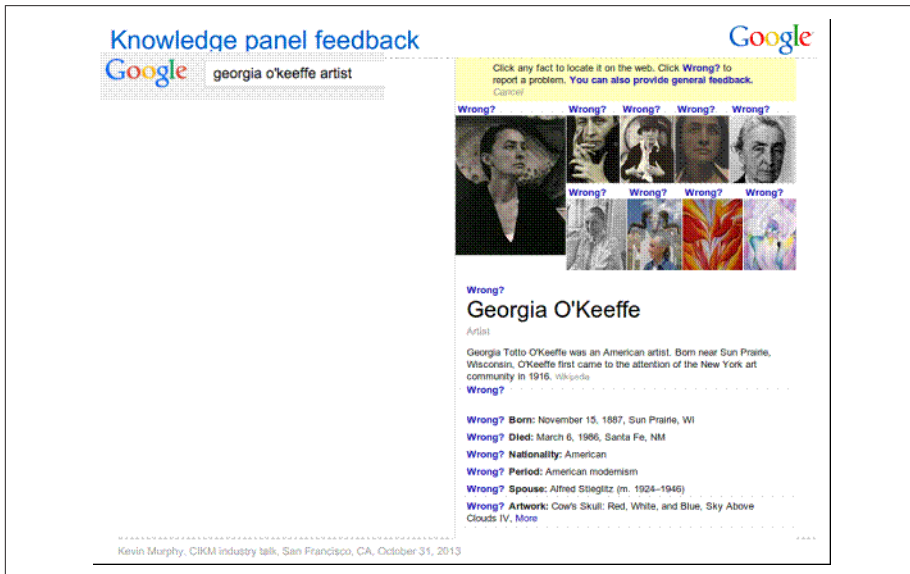


Figure 6-73. Knowledge panel feedback

Google has many patents that could potentially pertain to the Knowledge Graph and/or the Knowledge Vault. Each covers a particular aspect of how to extract information into a knowledge base. Here are a few examples for further reading:

- Knowledge Graph–Based Search System
- Extracting Patterns and Relations from the World Wide Web
- Determining Geographic Locations for Place Names in Fact Repository

There are many more that apply to this complex topic, and it will remain an area of investigation for some time to come.

The Future of the Knowledge Vault

As of October 2014, the Knowledge Vault is still a concept in its infancy. As noted earlier, it is just a research project. The algorithms are in primitive states of definition, and the required processing power is quite substantial. Google will keep investing in these types of technologies as it tries to find more and more ways to provide better and bet-

ter results. Its goal, and that of other search engines, remains to “organize the world’s information,” and it will keep investing in that goal until it succeeds.

However, this process may take a decade or more. This means we will see gradual changes continuing over time. Even if a single breakthrough provides Google with access to 1 billion facts, which sounds like a large number, it will still only impact a very small percentage of search results.

However, understanding the concepts of semantic search and the Knowledge Vault can in turn help you understand a bit more about where search engines are going.

Conclusion

By now you should be aware that a search engine–friendly website is the first step toward SEO success. In the next chapter, we will demonstrate how links are also a critical piece of the SEO puzzle—particularly when targeting highly competitive terms. However, if you have not made your site crawler-friendly and optimized, all of your other efforts—whether they’re link development, social media promotion, or other tactics to improve search visibility and increase search traffic—will be wasted. A website built from the ground up to optimal specifications for crawler accessibility and top organic exposure is the foundation from which you will build all SEO initiatives. From this solid foundation, even the loftiest of goals are within reach.