

# Optimizing Content for the Web

*Joe Dolson and Lori Wagoner*

*This paper covers the essentials of organizing and preparing materials for publication on the Internet. We'll review ways to convey key concepts to your audience in multiple ways, including the incorporation of pictorial, graphical, and contextual navigation. We discuss how and when to team with a web designer and what to look for in a good HTML coding partner or web database developer. The best practices for publishing information online and methods for keeping web-based materials evergreen with less work are also covered.*

## Designing content

You are already an excellent technical writer and have mastered writing everything from advertising copy to end-user documentation. Now you are presented with a project to write content for a website. Is getting ready to publish content on the web any different than print? Absolutely! Let's face it! We have all been to bad web sites. These sites leave us feeling confused, overwhelmed, or irritated. Creating a site map, keeping your materials tightly organized, writing concisely, and taking advantage of visual aids for the reader all help you get your point across effectively on the web.

Optimizing content for the web isn't just about the cleanest copy and the clearest navigation. In this case, "content" can be defined as anything that affects the usability of the materials you plan to publish on the web including architecture, navigation writing, visual elements, and coding. It is also about optimized coding practices. If you decide to work with a developer, look for a solid commitment to optimized code so that your materials will function with no errors and be easy to update later.

## Architecture

Information architecture is the way that your content is organized and the navigational structure created to access it. Web site architecture parallels building architecture, to a degree. Just as with a house, great curb appeal is just not enough to make a website draw in customers. Just as we can get lost in poorly designed buildings, poorly architected websites confuse readers. Think about it. In a building, would we want to go upstairs to get to a room downstairs? Likewise, should we have to go back to the main page to get to the navigation menu? Creating a well-planned, user-centered architecture allows your readers to get in, find what they want, and exit in the most efficient way possible.

Just as a builder requires a blueprint, an important step in formulating your content design is to create a site map. Sketch some possible organizations using a hierarchical structure. Don't worry if you don't get it perfect the first time around. It may take several iterations for you to outline a structure that will best match the potential access paths to information for your intended readers. You might begin by listing the individual documents that will be

incorporated into your design. Group these into categories that make sense by subject and organize them alphabetically or by importance. Remember that you can overlap subject areas if content applies to more than one area. Clearly map out how readers will connect from concept to concept. Even if you are rushed for time, don't skip doing a site map! Time spent clearly laying out your design will pay back in the ease of site construction and maintenance later. There are also software packages to create site maps, if you don't like pencil and paper. For one example see <http://www.smartdraw.com/examples/websitemap>.

When writing content to be included in a larger site that already exists or is under development, try to be consistent with or compliment the dominant style on the site. Take the time to analyze the look, feel, and tone of the entire site. Plan where links will be required to new materials from existing content and make a map of how your content will link to existing content. Creating a site map of your portion is also relevant even if you are creating a limited amount of content to be included in an existing website.

Keep in mind that no page really stands alone on the web and must be connected to other pages in order to be found, read, and used! Plan to use logical linking labels and avoid the terms "click here" or "go." If a reader can't tell where a link will go, he or she may not try it at all! Provide logical, non-linear, and intuitive pathways to information to augment your hierarchical design. On the web it is easy to provide multiple pathways to information rather than just one! Lastly, make your navigation pathways clear and easy to follow. Avoid "marooning" your readers with pages that lead to a dead end. Make sure your readers know where they are, where they have been, and how to get back to the top-level of the site easily.

## ***Writing to Your Audience***

Web readers are fickle and their attention is easily lost. It is essential that you correctly anticipate your audience before publishing your information on the web. Consider who your target audiences will be and try to identify each type of reader. Think about the primary audience for the site as well as other types of readers that may use the site. For example, are you aiming to reach engineers, marketing executives, sales managers, business-to-business customers? Think about each group's principal information needs. What are they looking for and how might each type of reader approach finding this information using your site design? With this in mind, go back and readjust your preliminary site map to allow for differences in information seeking behavior of various groups and individuals.

Web audiences differ from print audiences in the amount that they skim more than read. According to a Sun Microsystems survey conducted in 1998, about 80 percent of online users skim rather than read content. As the reader skims through online materials, if a site doesn't seem relevant or doesn't get right to the point, the reader moves on! In the interest of keeping your readers,

eliminate as much jargon, slang, and non-essential text as possible and keep your writing clear and concise.

When writing for the web, keep it short! Ideally, sentences should be under 15 words in length and paragraphs at about 50 words. Design your documents with additional content that can be accessed using the “drill down” method. This allows some viewers to access more information if they choose, without overwhelming the majority of readers. Also, short web pages avoid the problem of readers who do not like to scroll through the entire document. Try to keep vertical scrolling to a minimum and do not use horizontal scrolling.

In terms of style, web readers are more receptive to humor and informal writing. Write casually when you need to draw your users in – but be exact! In fact, you can be more flexible on the web and can write each page for the user who is most likely to use it. Most importantly, proofread, proofread, and proofread again. And let somebody else do it, too! Sites that have typographical errors, especially at the top-level, tend to erode reader confidence in the accuracy of the materials presented.

One of the greatest advantages to the Internet is that it removes geographical boundaries to communication. When planning your content, give consideration to international and non-English speaking readers. Depending on your customer requirements, you may want to add alternative versions of text, forms, and other content. Also, avoid humor or graphics that are culturally specific and may be misinterpreted by international readers.

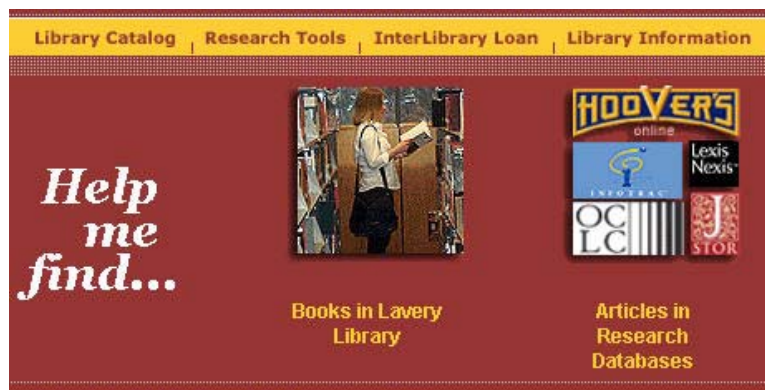
Information on the web becomes stale and out-of-date quickly. Nothing turns readers away like old or inaccurate information. Plan at the beginning of your project for how and when information will be reviewed and updated. Specifically who will be responsible for this and at what time intervals? It is a good idea to incorporate a modularized approach (including files and data to be refreshed) into your original design.

## ***Using Multimedia Elements***

Plain text is really boring! Multimedia elements used on a website might include drawings, graphics, photographs, animation, audio, or video. Adding media elements increases the interest of your readers. Traditional methods for organizing information (alphabetical, chronological, hierarchical, and classified schemes by subject) all work well on a website. The difference in writing for the web is that you can use these types of organizational strategies in combination with pictorial, graphical and contextual approaches to convey your ideas in more interactive manner. Here are a few examples of alternative approaches to navigating on a website.

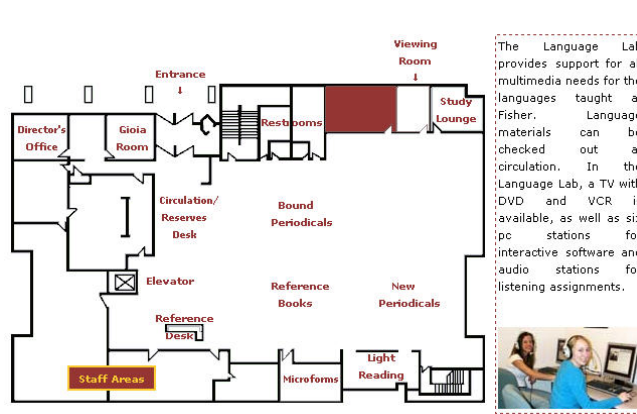
One approach to adding alternative navigation is to add pictures that help guide the reader to frequently accessed services. For example, we found at Lavery Library that users were confused by terms such as “Periodical Holdings” and “Catalog.” To counter this, we selected the five most popular activities performed by the students and created pictorial representations for

readers to click on. This was done in addition to the drop-down menu navigation already provided on the site. Two of these appear in the screenshot below. Students found that locating the correct locations to search for books, look for articles in databases, and other common activities took less time and were clearly visible from the top-level of the website. Of course, only a finite number of these elements can be added to any given page without risking a cluttered effect.



**Adding pictorial navigational elements helps users find information fast.**  
<http://library.sjfc.edu>

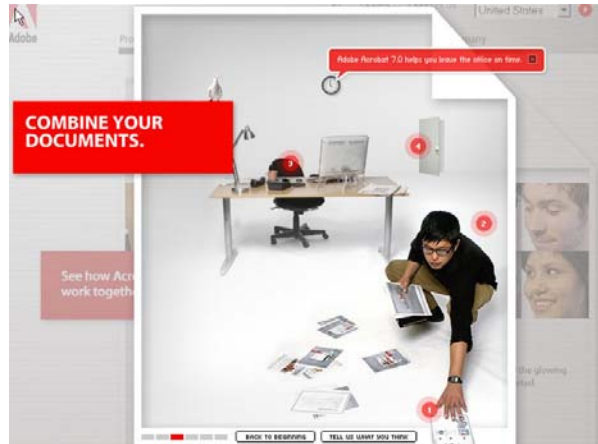
Another method of creating multimedia pathways for readers to follow is to incorporate an interactive graphical approach to your design. This is particularly helpful when conveying directional information. For instance, to help users find their way around the building and access resources, we put together an interactive “tour” of the library’s three levels which was based on an architectural floor plan of the building. We then added descriptions, photographs, and navigational links to all areas in the library.



**Using an interactive graphical approach helps convey complex information.**  
<http://library.sjfc.edu/tour/tour1.asp>

Adding multimedia elements can generate interest when you make it responsive to your customer’s contextual movements on the site. Also, when materials that you are presenting are highly technical or contain subject-

specific terminology that may be unfamiliar to your readers, it is useful to add contextual help for them to explore. This can be accomplished by adding an interactive resource, such as an online dictionary to assist readers, pop-ups to describe features, or content that appears on “mouse over.”



**Contextual aids persuade readers to spend more time at the site.**  
<http://www.adobe.com/special/acrobat/tour/worktogetherbetter.html>

Remember, when planning to incorporate multimedia elements to make sure that each item relates in a meaningful fashion with written content. Also, allow users to bypass items that are high-bandwidth (such as Flash animation) if they need or desire to.

Now that you have some ideas about the design of your content, it is time to move on to the next phase – publication on the web. This involves either preparing the content in an appropriate format or hiring a developer to help you. Some issues you will need to consider are the types of technologies that you would like to employ and the kind of styles you want to use. Also, thinking about accessibility is helpful for all sites and legally required for government sites. And, you are probably wondering about cost too. Don't feel discouraged. Just read on!

## Building the content

You've decided to build your website. You've thought about content, architecture, and the needs of your users. Maybe you've even started writing your content! What's next? Well, unless you're pretty experienced in the realm of Internet technologies – HTML, XHTML, CSS, JPG, PNG, PHP, SQL, ASP – then you're probably looking to hire an experienced web developer. When you're preparing your web development, you don't need to know the ins and outs of every web technology. But every technology serves a different purpose – and what you need to know is how a specific technology can help you! So it's time to start throwing around some terminology.

<b>ABBREVIATIONS:</b>	
<b>HTML:</b> Hyper text Markup Language	<b>PHP:</b> Hypertext Preprocessor
<b>XHTML:</b> Extensible HTML	<b>ASP:</b> Active Server Pages
<b>CSS:</b> Cascading Style Sheet	<b>JSP:</b> Java Server Pages
<b>XML:</b> Extensible Markup Language	<b>PERL:</b> Practical Extraction & Report Language
<b>SQL:</b> Structured Query Language	<b>SSI:</b> Server Side Include

So what are all of these things? What do you need? Wait - this isn't even everything! Well, you're right. And we're not going to explain everything to you. What you need to know is what kind of technologies these abbreviations represent, and what use you can get from them.

The first major category is the "Markup Languages." HTML, XHTML, and XML are the big names here. Regardless of what other technologies you end up using – you will definitely use one of these. This is the language of structure. These languages are the foundation of the web, because they tell the browser what purpose each element of content serves. But they don't do anything else – there's nothing pretty about them (we'll talk about THAT later), and what's more – there's nothing interactive about them.

This is where we start using a language like Javascript. I'm sure you've heard of Javascript – providing all sorts of nifty tricks to give a bit more zip to your website. Javascript is unquestionably a useful language. It's not quite perfect, though – it misses out in one key manner. Javascript is a "Client-side" language. This means that the script is carried out by the user's computer. That is, it's carried out if the user's computer is set up to allow it to. Surprise, surprise – there's nothing at all you can do to enable your users to use your nifty new tricks. Some people will simply not see it.

But there's an alternative – and there are many flavors of it! The alternative is "Server-side" programming. This includes PHP, Perl, PYTHON, ASP, JSP, and no shortage of other less well-known breeds of scripting. "Server-side" programming does exactly what it sounds like – executes code on the server. You control the server, and therefore you have no worries whether the browser will be able to interpret your website.

And the best thing about server-side programming is the way it interacts with databases. This is where we get to mention a favorite rule of web development - update once. Using good server-side tools gives you the opportunity to use the same code over and over – update it once, use it throughout the site. A database can store your entire site, code free – and a server-side scripted site

can take all those bits and pieces of information and assemble them for you. This can allow all of the site maintenance to be done through a customized and simple interface - just type in your plain text content, hit submit, and VOILA! Your site is updated. Not all server-side scripting will offer this level of functionality, granted, but at the most client-friendly level you have the potential.

So, what's the point here? These technologies allow you a choice – once you've figured out your project requirements for a site, you can figure out what you think you might need. Server side programming is more expensive than pure HTML – it's more complicated, and requires more programming expertise and more time. However, it means less maintenance. If you have a small site with a fairly straightforward organizational structure, you probably don't need complicated server technologies. But if you're going to be organizing thousands of documents that need to be updated on a weekly basis, go for a dynamic database-driven site.

So this is your first step – figure out the scope of your project, and identify the type of technologies you might want. You don't need to specify the language – just know the options.

Now we've got things moving along. We have our static HTML or our database driven project – let's take a look.



**joedolson.com – without styling!**

Okay! Now I'd call that pretty boring. What have we forgotten, here? Well, of course – style! Style is a perfect example of a web technique that has changed drastically since the earliest days of HTML. Back in the 'bad old days' of web design, HTML was a confusion of tags that were *presentational* and other tags that were *structural*.

An example of structural code is the Heading tag – H1, H2, H3, etc. These codes inform you of the importance of a particular group of text. Structural code tells you that this piece of information describes the block of information that follows, and it tells you what the order of importance of the blocks of text is.

An example of presentational code is the Font tag – which tells you only how this particular group of text will look, and conveys no information about its value.

A common error in web design is to use a piece of presentational code in an HTML page instead of a piece of structural code simply because it gives you the look you want. Fine for the eyes, but not so great to a search engine!



**The exact same site – can you believe it?**

Today, all presentational HTML has been *deprecated* by the World Wide Web Consortium ([www.w3c.org](http://www.w3c.org)), the international watch dog for standards in web coding. HTML should be written with an informational architecture framework in mind. Instead, Cascading Style Sheets (CSS) are intended to control the appearance of a web site. This has many advantages, including the ability to locate all of the stylistic design of your page in one place. Instead of changing colors on 200 individual pages – you only do it once.

<b>Definitions:</b>
<b>Structural Code:</b>
In HTML, markup refers to the identification of the contents of a web page. An HTML tag should identify that an item is a ‘first-level heading’ or a ‘table’ or a ‘paragraph.’
<b>Presentational Code:</b>
HTML markup includes many tags that don’t tell you what the content is – instead, they tell you how it looks. ‘Font’ specifies a particular face. ‘Center’ tells you where on the page it might be.
<b>Deprecation:</b>
Code is deprecated when it is judged to be in contradiction with the intended separation between Structural and Presentational coding. Although deprecated code will continue to work for now, it is not guaranteed to be supported in the future.

When you're planning your web development, you don't need to know the ins and outs of every web technology. But every technology serves a different purpose – and what you need to know is how a specific technology can help you. So here's the run down so far:

### **HTML, XHTML and XML**

- Different flavors of the same thing. Every site will use one or more of them!

### **Web Programming:**

- PHP, ASP, Java, PERL – Server side technologies. Run programs before the user receives the page.
- Javascript, CSS – Client Side technologies. Very useful, but depend on the user choosing to run them. CSS is only for presentation!
- SQL and Access – Databases technologies.

### **Style:**

- CSS for layout.
- Graphics! We haven't talked about this – but to make a really stellar website, you'll need to consider graphics. We'll get to these later on.

The full scope of project management for a major web project is well beyond this paper – but suffice it to say that a good project manager will need to become familiar with all the above technologies. It's not necessary to become a great Java authority, but it's crucial to know what a great Java coder can do!

So, if you've sat down and figured out the scope of your project, and what kind of technology you want to work with, you'll need to add the people to do the work. It's time to pick a developer.

The core commitment you'll need from your developer is optimized code. Optimizing content for the web isn't just about the cleanest copy and the clearest navigation – it's also about optimized coding practices. And checking code is a very easy thing to do!

You don't need to learn any code to inspect a site for optimization. In fact, you don't even need to look at it. The wonders of the web supply many free tools to examine the underbelly of a site in many different ways. Largely used by developers to check for mistakes, they can also serve other purposes – for you!

You've looked at good and bad navigation already, so you'll have that in mind from the beginning of your search. You know what you like from the look and feel of a site. But studying the code is a wholly different playground!

## Step 1:

**Validation.** “Valid” code is code that conforms exactly to the standards set out by the World Wide Web consortium for the language specification. You may find a lot of sites that look just fine but contain hundreds of validation errors – such as Yahoo!

<b>Address:</b>	<input type="text" value="http://www.yahoo.com"/>
<b>Encoding:</b>	utf-8 <input type="text" value="(detect automatically)"/>
<b>Doctype:</b>	<a href="#">HTML 4.01 Transitional</a> <input type="text" value="(detect automatically)"/>
<b>Errors:</b>	286

*As you can see, some major sites have a LOT of invalid code.*

The simple fact is that browsers are very forgiving – they’re designed to be able to read pages that were written before the HTML specifications were actually set. But while browsers are forgiving of HTML errors, they also interpret them very differently!

Following the rules helps your site be more consistent across the browsers. You can check the validity of a site’s HTML at <http://validator.w3c.org>. However, the only sure way to consistency is to look at your site in multiple browsers. If you don’t test in Internet Explorer 5.5, it probably won’t look right in Internet Explorer 5.5.

## Step 2:

**Accessibility.** Accessibility is the degree to which a website can be used by individuals with disabilities. Accessibility frequently goes hand in hand with usability – although they’re not at all the same thing. When you go to the effort to make a site accessible, you will always make the site a little bit more user friendly.

The two biggest elements of accessibility are building in keyboard shortcuts to navigation and creating page navigation that can be used by non-visual web browsers. The biggest group of disabled web users is the blind. Here's an example of what a blind web browser might find if looking for the hours at the St. John Fisher Library:

Page has two headings and fifty-five links Lavery Library colon St. John Fisher College dash Mozilla Firefox Link Search the Catalog Link Search Reserves Link What's New! Link Other Library Catalogs Link List of Library DVD's Link Research Databases Link Periodical Holdings Link Subject Guides Link Guides and Tutorials Link Go to Interlibrary loan Link Frequently Asked Questions Link Hours Link General Information Link Library Services Link Library Policies Link Student Employment Link Angle Link Byrne Papers Link Cardinal Courier Link Collegium Link Decker Papers Link Documentary Photos Link Douglass Collection Link Jo Roffs Link Pioneer Link Rochester Radio History Link Slide Collection Link A slash V Booking Request Form Link College Teaching Resources Link Faculty Library Resources Link Grants Resource Site Link Information Literacy Link Reserve Information Link Ready Reference Link Selected Websites Link Blackboard Link Academic Calendar Link Web E dash Mail Link SJFC Intranet Link Graphic slash shim.gif Table with one column and three rows Table with two columns and one row Link Graphic St. John Fisher College Link Graphic Lavery Library Table end Table with one column and one row Link Graphic College Link Graphic slash shim.gif Table end Table end Table with nine columns and one row Link Graphic slash shim.gif Link Graphic slash dbscbone .gif Link Graphic slash dbscbtwo .gif Link Graphic slash dbscbthree .gif Link Graphic slash dbscbfour .gif Link Graphic slash dbscbseven .gif Link Graphic slash dbscbsix .gif Link Graphic slash dbscbfive .gif Link Graphic slash shim.gif Table end Table with one column and one row Graphic column spacer Table end **Lavery Library Library Hours**

What I've included here is everything a blind web user would have to hear until they get to the content they're looking for, and wow! That's not very friendly. Can you imagine having to listen to 5 minutes of navigation links on every page before you can even start to find what you want? This site is particularly unfriendly to a blind user.

Try this one:

Page has nine headings and sixty-six links HTML and CSS Tutorials, References, Articles and News dash HTML Dog dash Mozilla Firefox htmldog.com dash A Good Practice Guide to XHTML and CSS . Search colon Edit Search button This page link **Skip navigation. alt plus s . . .**

Seems pretty obvious what advantages we've got here! We have access keys, so you can easily use the navigation without having to listen through it all – and we've also got these hidden links that navigate between content and the navigation. If you know that you're on the right page and just want to listen to what it says – just hit **Alt + s** and you'll go right there!

Of course, these aren't the only rules for a fully accessible site. The visual capabilities of the color blind should be taken into consideration, as should the problems of cognitive disabilities and even hearing impairment. Audio examples should provide transcripts, for example!

There are many places to check for compliance with accessibility. There are also multiple levels of accessibility to check for! Here are some options:

**Section 508:** Section 508 is the compliance level required for government and government funded organizations. Part of the American Disabilities Act, there has been precedent for this law in application to commercial sites in the state of New York.<sup>1</sup> Section 508 compliance can be tested at <http://bobby.watchfire.com> or at <http://www.contentquality.com>.

**WCAG:** The web content accessibility guidelines are a separate set of guidelines set by the World Wide Web Consortium's Web Accessibility Initiative (WAI.) They're not any less strict – and in many ways are stricter. They're a slightly different interpretation of accessibility. It is, however, perfectly possible to comply with both sets of guidelines. WCAG offers multiple levels of compliance, having three groups of guidelines ranked by priority. Both the accessibility checkers above will also give you the option of verifying WCAG accessibility.

**Colorblindness:** Testing for color accessibility is largely an issue of contrast. The major online resource available is <http://www.vischeck.com>. This company provides a service that will interpret the appearance of your site as if you were viewing it with each of three different types of color blindness.

### Step 3:

**Speed.** The definition of this should be pretty clear - the speed of your website is simply a factor of how long it takes to download. How patient is a potential customer when it comes to your website? Well, the generally not so patient web reader will wait a whole eight seconds. On a standard 56k dial up connection, that means a total page weight of around 30k.

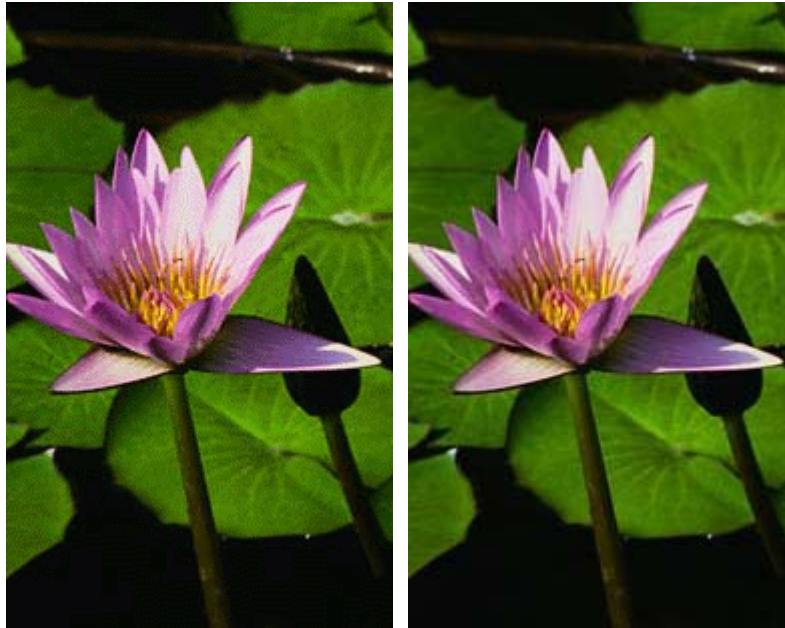
How do you check up on this? Well, the hard way is to look at the file size of every referenced file on a given page and add them all up. The easy way is to run the page in question through a good speed testing engine, such as what you'll find at <http://www.websiteoptimization.com/services/analyze.com>, which will do the whole thing for you.

Your existing customers may know exactly what they want and can get from you, and be willing to wait for it – but new customers want their information quickly! The most common issue in overweight web pages is the graphics – poorly compressed graphics, or graphics using the wrong compression type.

---

<sup>1</sup> Eliot Spitzer, NY state Attorney General, Agreement of August 19, 2004 states "[t]he Attorney General opined that the Americans With Disabilities Act requires that private web sites be accessible to blind and visually impaired Internet users. The ADA generally dictates that all "places of public accommodation" and all "goods, services, facilities, privileges, advantages, or accommodations" of places of public accommodation, must be made accessible to disabled citizens, absent undue hardship. New York law provides similar civil rights protections."

You may not have realized that there's a right and a wrong file type, and it's not exactly accurate to put it that black and white. But compare the following two versions of one image:



**gif, 37.7k**

**jpg, 9.6k**

Simply put – the jpg format is intended for photographic images, and is ideal for them. The gif format is intended for simpler images with fewer colors – and will give you much better results in a drawing or other non-photographic image. The visible difference between the above images is very slight – but the load time in a website is significantly different!

There are a lot of other issues at play with graphics. Resolution should be kept low. The optimal dpi is 72 – any higher just means longer download times! The graphic types are limited – just gif, jpg and png are eligible. Sorry, no BMP images! Just keep it firmly in mind – images can be reduced in size without losing a significant degree of image clarity.

## Anticipating costs

A good website may be expensive – but so might a bad one. Even if you save a bundle by hiring the bottom-line developer, the loss of web traffic from a poorly designed site can have an enormous long-term impact on your business. Keep this in mind when you're trying to anticipate your costs. The cost of development is far from the only expense you can incur.

Think about your website in two discrete blocks – the long term cost, and the short term cost. In the short-term, there will be a fairly simple cost evaluation. The developer or developer's fees, the web hosting, various software costs. For the long term, you need to think about maintenance and continuing fees for web hosting and bandwidth.

There is one other consideration – the backup. The last thing you want to do is call up the developer one day and say:

*“The server crashed and wiped everything. Do you have a copy of our site?”*

Maybe your developer does – but there’s no guarantee. You need to plan on a way to save your hard work! On a frequently updated site, you may need to back everything up every day, either onto a portable medium, such as a DVD-R, or keep everything duplicated on a development server.

Once again, the size of your project dictates most of these considerations. Investing in something like a content management system from the beginning will increase your initial costs, but if you have a large site that requires a lot of upkeep – it’ll definitely reduce your long term costs. Just keep one thing firmly in mind – if you’re paying more for a service, will you get the money back later in added value or saved maintenance costs? It’s a tough decision.

Even in a small site, there are things that can be done to make the design modular. A database isn’t the only way to keep things easy to update. The navigation, which should be the same on every page, can be maintained in a separate file. So can the header information. For that matter, anything at all that will be used more than once can be stored in what’s called a *server-side include*. Another useful server-side method, server-side includes allow a page to be assembled by the server before being sent off to the reader.

There’s a simple rule – *update once*. If you will use something a hundred times – you only want to update it once. Even if you only use it twice – you only want to update it once.

## What to prepare for or with your developer

Before anybody signs a contract – make sure you’re on the same page. Have you discussed the size of the project? Has maintenance been considered? Does the developer know where the project will be hosted and what technologies it will support? All of these things need to be taken care of in advance.

If you want to make the developer really happy right from the beginning, let him or her know the following things:

1. The key project deadlines.
2. The server access codes.
  - a. Is this a development only server, or is it production?
  - b. If it’s development – make sure the platform is identical to the production environment!
3. How much content will there be?
4. What categories does it need to be divided into?
5. When will the content be delivered, and how?

When it comes right down to it, if you've planned well and picked the right person to work with – you're ready to go. After all, by this point it all comes down to your specialty – communicating!

## Conclusion

Optimizing content for the web may seem overwhelming at first. Keeping your work tightly focused on the reader, to the point, and organized are essential! Learning more about web technologies can help you to tackle projects yourself or to team with a developer. Remember that there is a learning curve for everyone who is publishing on the Internet. Even people who have been at this for some time need to continually update their knowledge as web technologies change, improve, and become more complex. You too can do it!

## Resources

### Books

Garrand, Timothy. Writing for Multimedia and the Web. Second Edition. Woburn: Focal Press. 2001.

Hammerich, Irene and Claire Harrison. Developing Online Content: The Principles of Writing and Editing for the Web. New York: Wiley Computer Publishing. 2002.

Krug, Stephen. Don't Make Me Think! A Common Sense Approach to Web Usability. Indianapolis: New Riders Publishing. 2000.

McGovern, Gerry and Rob Norton. Content Critical: Gaining Competitive Advantage Through High-Quality Web Content. Harlow: Financial Times Prentice Hall. 2002.

Nielsen, Jakob. Designing web usability. Indianapolis: New Riders. 2000.

Price, Jonathan and Lisa. Hot Text: Web Writing That Works. Indianapolis: New Riders. 2002

Usborne, Nick. Net words: Creating High-Impact Online Copy. New York: McGraw-Hill. 2002.

Veloso, Maria. Web Copy That Sells: The Revolutionary Formula for Creating Killer Copy Every Time. New York: AMACOM. 2005.

### Web

#### Accessibility

<http://www.w3.org/WAI/> - Web Accessibility Initiative

WAI, in coordination with organizations around the world, pursues accessibility of the Web through five primary areas of work: technology, guidelines, tools, education and outreach, and research and development.

<http://www.bobby.watchfire.org> - Bobby web page analyzer

This free service will allow you to test web pages and help expose and repair barriers to accessibility and encourage compliance with existing accessibility guidelines, such as Section 508 and the W3C's WCAG.

<http://www.diveintoaccessibility.org> - Dive Into Accessibility

The online version of a book of the same name, this structured tutorial guides readers towards an understanding of the Why's and How's of accessible web site design.

<http://www.section508.gov> – Government info on section 508

This site, maintained by the Center for Information Technology (CITA), is the primary resource for Federal employees and the public to access resources for understanding and implementing Section 508 requirements.

<http://www.webaim.org> – Web Accessibility in Mind

WebAIM has provided comprehensive Web accessibility solutions since 1999. Their website provides extensive resources about Web accessibility. WebAIM is a non-profit organization within the Center for Persons with Disabilities at Utah State University.

<http://www.contentquality.com> – Cynthia Says Portal

Cynthia is a web content accessibility validation service, designed to identify errors in design related to Section 508 standards and the WCAG guidelines. The main purpose of the portal is to educate web site developers in the development of Web Based content that is accessible to all.

## Usability and Web Content Guides

<http://www.alistapart.com> – Programmer's resources and articles.

A List Apart Magazine (ISSN: 1534-0295) explores the design, development, and meaning of web content, with a special focus on techniques and benefits of designing with web standards.

<http://www.useit.com> – Jakob Nielsen's personal website.

Jakob Nielsen, author of "Designing Web Usability: The Practice of Simplicity" and numerous other usability texts, maintains this site as a reference for good information on usability.

<http://usability.gov/> - Usability from The U.S. Dept of Health and Human Services

Extensive collection of usability resources maintained by the U.S. Department of Health and Human Services. Includes basics on what usability is, accessibility resources, market research, guidelines and checklists for many issues. Mostly links to other sites.

<http://www.usabilityfirst.com/> - from Diamond Bullet Design.

Resource providing usability suggestions from a Michigan-based consulting firm.

## Author information

**Joe Dolson** is a free-lance web designer who provides a wide range of web-related services including new site development, updating and redesign of existing sites, logo design, custom graphics, and sound editing. He specializes in web usability issues with a particular focus on making websites accessible for users with disabilities. His website is <http://www.joedolson.com>

**Lori Wagoner** is the owner of Wagoner Research. She has a Masters in Library and Information Science and broad experience with government, industry, and higher education. Over the past 13 years, Wagoner Research has specialized in project management, business and international research, end-user training, knowledge management, and organization of information. Her website is <http://www.wagonerresearch.com>

**Return to  
Session Grid**