

## **Accessibility Primer**

© 2010

Envision Foundation  
610 N Main, Wichita KS 67203  
[www.EnvisionUs.com](http://www.EnvisionUs.com)



# Table of Contents

---

Introduction .....	1
What Is Accessibility?.....	2
Section 508 vs. WCAG .....	2
Which one should I use? .....	2
Accessibility Pros and Cons .....	3
Steps to Accessibility.....	4
The Basics.....	6
Language .....	6
Color.....	6
Headlines .....	7
Acronyms and Abbreviations.....	7
Links .....	7
Menus .....	8
Tables .....	8
Forms and Input.....	9
Sound and Animation .....	10
Images.....	10
PDF .....	11
Video .....	11
Resources.....	12
Books.....	12
Links .....	12
Index.....	13



# Introduction

---

While creating and maintaining accessible online content is can be cumbersome, and somewhat confining, it does not mean you can't have a great web design with appealing, interactive content. In fact, the real implication of accessible web content is what web designers should be striving for: *content with semantic markup and presentation that strictly conforms to established design standards.*

If you build your web site on a foundation of just three principles, your web site will be accessible to everyone.

- 1) Everyone, regardless of ability, has a right to information.
- 2) Assume only that your users are capable of receiving and sending text only.
- 3) Accessible content is content presented within the precepts of proper web design.

In the following pages you'll learn about the evolution of online accessibility, why it is important, and how accessibility principles should be applied to your web site.



# What Is Accessibility?

---

Accessibility is a comprehensive term used to describe the level at which something can be accessed by those with disabilities, or the process used to achieve access for those with disabilities. Accessibility most commonly refers to requirements of Section 508 (1998 amendment to United States Workforce Rehabilitation Act of 1973) or the Web Content Accessibility Guidelines (WCAG) of the World Wide Web Consortium (W3C).

## Section 508 vs. WCAG

Section 508 is a US federal law mandating that all federal agencies make electronic information accessible to people with disabilities. The law also requires organizations that work with federal agencies to meet the same accessibility requirement.

WCAG is a set of recommendations for making web content more accessible to people with disabilities. The guidelines were first released by the W3C in May 1999. It is important to note that the WCAG states that following these guidelines will make content more accessible to users in general (see principle #3).

## Which one should I use?

The WCAG document was developed by the W3C, an organization responsible for developing myriad web standards. Much of the software we use today is built to meet one or more W3C standards. Section 508 was written by legislators.

The advantage of WCAG is what the W3C refers to as “success criteria written as testable statements that are not technology-specific.” Essentially that means that each of the guidelines in the WCAG is written as an easily understood statement resembling “if content is this, and you provide that, then it is accessible.” The WCAG was developed by a group of industry leaders knowledgeable in web and application development and web technologies.

If, however, you work for a government agency or your content is part of any business relationship with a government agency, it is recommended that you follow Section 508 requirements.



# Accessibility Pros and Cons

---

## Pros

- Content conforms to web standards which provides usability, longevity, consistency
- Content is open and available to a larger population
- Content and presentation are developed separately

## Cons

- Some web technologies (e.g., DHTML, client-side scripting, AJAX) demand more developer time to implement, can be implemented only with constraints, or cannot be utilized
- Limited implementation of technologies that mass audiences have come to expect
- Content must be tested manually in some cases



# Steps to Accessibility

---

- 1) Familiarize yourself with the standards – Section 508 and WCAG.

You will have no problem finding reference materials on the web for both Section 508 and WCAG. Regardless of your chosen method of meeting accessibility, it's recommended that you research both. A thorough understanding of the spirit of accessibility guides and laws will be of great benefit to you.

- 2) Anyone writing content should learn the accessible content basics.

Those other than yourself that will contribute content to your web site, possibly through a company blog or content management system, should be trained in the basics of writing for accessibility.

- 3) Invite outside testers, preferably those with disabilities or those using assistive technology.

Testing is extremely important. Many of the criteria of the WCAG require manual validation, i.e. verifying actual HTML code. Human verification is fallible, especially when testing to verify content works with technology you do not have.

- 4) Develop content validation and approval procedures.

All content must be approved before being released. All design and code changes should be thoroughly tested before release. To ensure accessibility of design and content, everything should pass through an approval chain. The validation procedures for all approvers should be documented.

- 5) Familiarize yourself with the accessibility functionality of the software used for web design and content submission.

Whether you're using a simple HTML editor, an advanced rapid application development platform, a simple blogging platform, or content management system, familiarize yourself with what, if any, accessibility functionality is available.

Does the software perform HTML validation? Are there specific validation settings for accessibility?

Remember, even if your software can be used as an aid in accessibility validation, no software can check every aspect of accessibility. Ultimately it will be up to you to confirm accessibility requirements are met.



- 6) Build your project from the start with accessibility principles.

Once you realize that your message or purpose is meaningless without an audience, you can make accessibility a priority in every aspect of development and maintenance.

- 7) Methodically add newer technologies only after initially meeting accessibility requirements.

After initial testing is complete and accessibility has been confirmed, you can then begin adding other technologies to the project verifying accessibility as you go.



# The Basics

---

This section uses some HTML terminology as it relates to web accessibility. If you are not familiar with HTML or XHTML you should consider a course in web design or at least online research about the topics included in this section.

The general rule for writing accessible web content is "use semantic markup." Use appropriate (meaningful) HTML tags to markup your content and design the presentation (layout, color, etc) of your content separately. If content is written separately from presentation, we can be assured of its accessibility without images, without Cascading Style Sheets (CSS), without audio or video, and in all font sizes. Examples of semantic HTML markup are:

- Headlines should be enclosed in *h1*, *h2*, etc, tags
- Underline (*u*) should not be used as it can be confused with links; instead use *em* or *strong*
- Paragraphs should be enclosed in *p* tags

## Language

In the spirit of accessibility we want to get our message to the broadest audience. This includes making content available in multiple languages and writing in such a way that our users can comprehend our message.

Always supply a default language for the web document using the *lang* attribute of the HTML, or preferably the Extensible Hyper-Text Markup Language (XHTML), tag. If any content within the document uses a language different from the default then use the *lang* attribute for the markup of that section.

Your writing should also target your audience's comprehension level. Reading level tests can indicate how difficult content is to comprehend. The Flesch-Kincaid Grade Level score uses an algorithm to determine the reading level of your content. If you are writing for college graduates, you should edit for a Flesch-Kincaid Grade Level score of 14-20. If you are writing for a high school audience, target a score of 9-12. It is recommended that when writing for general audiences you should target middle school.

There are several methods for testing content for readability. If you are using Microsoft Word, you can get the Flesch-Kincaid Grade Level score after a spell check. You first must turn on this option using Office button > Word Options > Proofing > Show Readability Statistics. You can also use any of the many readability tools on the internet.

## Color

There are many things to consider when choosing colors for web content. You'll want to maintain your branding and use colors effectively, but consider those that cannot see color well



and those that cannot see color at all. To maintain accessible content, never use color to convey meaning. When you do use color in text, you must consider how the text color and the background color work together:

- Colors should have a color difference of 500 or more. (Black/White is 765)
- Colors should have a brightness difference of 125 or more. (Black/White is 255)
- Colors should have a contrast difference of 7.1 or more. (Black/White is 21)

Aside from the requirements above, also consider that any color you choose may be seen differently by those who are color blind. The calculations to determine whether colors you've chosen meet accessibility requirements are complicated. There are many online validators that will do those calculations for you. Some are listed in the Resources section below.

## Headlines

One of the most misused of HTML tags is the headline (*h1*, *h2*, *h3*, ...). Because we are writing for content and not presentation, headlines serve an important purpose: they create an outline of your web page. Every page on your web site should have one *h1* tag, and it should be the first headline used. This will usually be the title of your content. All headlines that follow should be *h2* or higher.

Headlines should be used in order so the page reads like an outline (*h1* for the most important topic (only one), then *h2* for sub-topics, then *h3* for sub-topics of *h2* headlines, and so on).

## Acronyms and Abbreviations

When using an acronym, always spell it out and use the ACRONYM tag with the title attribute on its first use.

Example:

International Business Machines (<acronym title="International Business Machines">IBM</acronym>) manufactures many products in addition to computers.

Always enclose abbreviations in the ABBR tag with the title attribute to expand the abbreviations.

Example:

The phrase Rock and Roll was first used <abbr title="circa">ca.</abbr> 1950.

## Links

Links to other pages, on your site or others, are common and necessary. To maintain accessibility compliance using *a* tags:



- The destination (*href*) of the link should never be a script function. To use client scripting in links use event attributes such as *onclick*.
- The destination should never be #. All *a* links should have a *href* that points to a valid anchor within the document or a valid internet Unified Resource Locator (URL).
- The text of the link should always inform the user of the link destination and should not use phrases such as "Click here."
- You should not use the *target* attribute. Targets are used for *frames* and opening links in new windows. Since we cannot know what software is being used to view our site, or the capabilities of our visitor, we do not presume to take over his/her computer. If the user wants to open a link in a new window, that is his/her prerogative. Do not use frames and do not pop-up new windows.
- The *title* attribute of the *a* tag should always be used to describe the link. However, when the *title* would just be a duplication of the link text, use a blank *title*. A blank title lets screen reader software know that the *title* is not necessary. If the *title* attribute is not used the screen reader software may read the entire address of the link in its place.

Example of a link with empty title:

```
<a href="registration.html" title="">Register a new account</a>
```

For lists of links, such as menus, see Menus below.

## Menus

Lists of 3 or more links, especially link lists repeated on many pages like navigation menus, need "skip links." A skip link is just a link before a menu that points to a position on the page just after the menu. It allows screen reader software and keyboard users to skip past long lists of links.

Example:

```
<a href="#afterNavigation">Skip navigation menu</a>
```

```
<a href="index.html">Home page</a>
```

```
<a href="login.html">Login</a>
```

```
<a href="account.html">My account</a>
```

```
<a name="afterNavigation"></a>
```

It is acceptable to hide these links using CSS. However, if the link is hidden, it should reappear when it receives focus.

## Tables

The practice of using *tables* for web page layout is no longer acceptable. Remember, our content is to have semantic markup and created separate from presentation. Therefore, *table* is only to be used for long lists of multi-dimensional (multiple columns) data, i.e. spreadsheet.



Simple one-dimensional lists should be tagged using an appropriate list tag (*ul*, *ol*, *dl*) and not *table*.

If a *table* is necessary, remember:

- You must use the *summary* attribute of the *table* tag
- You must use the *caption* tag immediately after the *table* tag
- You must use the *thead* tag immediately after the *caption* tag
- You may optionally use the *tfoot* immediately after the *thead* tag
- You must use the *tbody* tag immediately after the *thead* or *tfoot* tag

Example:

```
<table summary="Example of the proper use of table">
<caption>TABLE Example</caption>
<thead>
<th>Column 1</th>
<th>Column 2</th>
</thead>
<tfoot>
<th>Column 1</th>
<th>Column 2</th>
</tfoot>
<tbody>
<tr>
<td>Cell 11</td>
<td>Cell 21</td>
</tr>
</tbody>
</table>
```

Skip links can also be used for long tables. See Menus above for information on skip links.

## Forms and Input

Using proper markup in *forms* actually makes communication easier for users and maintenance easier for you. There are many tags and attributes used in the example below that are not commonly used but they are necessary to maintain semantic markup.

The more common attributes for *input* fields usually used for accessibility are not recommended. The *accesskey* attribute can interfere with keyboard shortcuts in the browser and other software. The *tabindex* attribute should not be used because it is hard to manage when updating web pages and if the web page is designed in reading order, *tabindex* is unnecessary.



*Fieldset*, along with the *legend* tag, are required for semantic and accessible *forms*.

Example:

```
<a href="#afterLogin">Skip login</a>
<form action="results.aspx" method="get">
<fieldset>
<legend>Login</legend>
<label for="IUsername">Username:</label><input type="text" id="IUsername"
name="username" />
<label for="IPassword">Password:</label><input type="password" id="IPassword"
name="password" />
<input type="submit" value="Login" title="Login" />
</fieldset>
<fieldset>
<legend>Create an Account</legend>
<label for="cUsername">Username:</label><input type="text" id="cUsername"
name="username" />
<input type="submit" value="Next >" title="Next step" />
</fieldset>
</form>
<a name="afterLogin"></a>
```

Skip links can also be used for long or complicated forms. See Menus above for information on skip links.

## Sound and Animation

Accessibility guidelines prohibit us from taking control of the user's system or using technology that is potentially harmful. The following is a list of HTML tags that should be avoided:

- *bgsound*: This tag plays background sound. It is often difficult to control the volume of the sound and it does not offer a way for the user to turn off the sound.
- *blink*: This tag causes the enclosed text to flash. Strobing content is known to cause seizures in those who have photosensitive epilepsy.
- *marquee*: This tag causes the enclosed text to scroll. This animation also causes a strobing effect that is potentially harmful for those who have photosensitive epilepsy.

## Images

All *img* tags must be accompanied by the *alt* attribute. The *alt* attribute supplies textual descriptions for images. Proper use of *alt* will ensure that sighted users do not have access to information that vision impaired users cannot access. While *alt* must always be used, there is one



caveat: If the *img* is used for decorative purposes only (not information), then the *alt* should be left empty. Also remember:

- The *alt* attribute should not contain phrases such as "image of" or "picture of"
- Avoid *alt* text that is more than 80 characters in length
- If you need more than 80 characters then use a shorter *alt* description and include the *longdesc* attribute

## PDF

Most Portable Document Format (PDF) files are comprised of images. Images in PDFs are not readable so we must tag our PDFs to maintain accessibility.

- 1) Set the default language first. File > Properties > Advanced Tab
- 2) If there is text in the images, try Optical Character Recognition (OCR) first. Document > OCR Text Recognition > Recognize Text Using OCR
- 3) If there is a mixture of text and images then you should "Tag" the PDF. Tagging adds AltText to PDF images just like HTML IMGs and also allows you to specify the order in which the text should be read. Advanced > Accessibility > Add Tags to Document
- 4) After Tags are applied, click the Show Order Panel button. Now sections can be tagged, added, or deleted and AltText can be edited, and you can change the order of the Tagged sections to change the read order.
- 5) After the above is complete, try the built-in screen reader. View > Read Aloud
- 6) Now run accessibility tests in Acrobat. Advanced > Accessibility > Full Check
- 7) Lastly, have an experienced screen reader software user test your PDF.

Additionally, if the PDF will be linked from your web site, be sure to include a link to the Adobe Acrobat Reader download page.

## Video

Video is one of the most difficult formats to adapt to accessibility standards. As noted in Sound and Animation above, screen flicker is particularly risky for those with photosensitive epilepsy. Flicker or strobing in the range of 2 to 55 times per second can induce seizures. Most online video plays at 30 frames per second. The Photosensitive Epilepsy Analysis Tool (PEAT) tool from the Trace Research & Development Center can determine whether your animations or video are likely to cause seizures.

For the visually impaired, auditory descriptions or textual descriptions should be available for each video. For the hearing impaired, open captioning should be available for each video.



# Resources

---

## Books

Mueller, John Paul. [Accessibility for Everybody: Understanding Section 508 Accessibility Requirements](#). New York: Springer, 2003.

Sydik, Jeremy J. [Design Accessible Web Sites](#). Pragmatic Bookshelf, 2007.

## Links

### [Color Contrast Check](#)

The Color Contrast Check Tool allows you to specify a foreground and a background color and determine if they provide enough of contrast.

### [Juicy Studio Readability Test](#)

This service analyses the readability of all rendered content.

### [HiSoftware Cynthia Says](#)

Designed to identify errors in your content related to accessibility.

### [Etre Accessibility Check](#)

Evaluates your page against a subset of the WCAG guidelines. These guidelines form the basis of most global legislation relating to accessibility.

### [W3C](#)

Not only will it compare your style sheets to the CSS specifications, helping you find errors, typos, or incorrect uses of CSS, it will also tell you when your CSS poses some risks in terms of usability.

### [W3C](#)

Most Web documents are written using markup languages, such as HTML or XHTML. These languages are defined by technical specifications, which usually include a machine-readable formal grammar (and vocabulary). The act of checking a document against these constraints is called validation, and this is what the Markup Validator does.

### [Total Validator](#)

Free one-stop all-in-one validator comprising a HTML validator, an accessibility validator, a spelling validator, a broken links validator, and the ability to take screenshots with different browsers to see what your web pages really look like.

### [Web Accessibility Evaluation Tool](#)

Used to aid humans in the web accessibility evaluation process. Rather than providing a complex technical report, WAVE shows the original web page with embedded icons and indicators that reveal the accessibility of that page.

### [Functional Accessibility Evaluator](#)

Analyzes web pages for markup that is consistent with the use of Illinois Center for Information Technology and Web Accessibility (iCITA) HTML Best Practices for the development of functionally accessible web resources that also support interoperability.

### [WebAIM Section 508](#)

An easy to use checklist of Section 508 web standards. It includes pass/fail criteria for each item.



# Index

---

- #, 8
- abbr, 8
- accesskey, 10
- Acrobat, 12
- acronym, 7
- AJAX, 3
- alt, 11
- assistive technology, 4
- bgsound, 11
- blink, 11
- blog, 4
- brightness difference, 7
- caption, 9
- color difference, 7
- colors, 7
- comprehension level, 6
- content, 1, 2, 4, 6, 7, 9, 11, 13
- Content, 2, 3
- content management system, 4
- contrast difference, 7
- DHTML, 3
- disabilities, 2, 4
- em, 6
- Fieldset, 10
- Flesch-Kincaid Grade Level, 6
- flicker, 12
- frames, 8, 12
- government agency, 2
- h1, h2, etc, 6
- Headlines, 1, 6, 7
- href, 8, 10
- HTML, 4, 6, 7, 11, 13
- img, 11
- input, 10
- lang attribute, 6
- languages, 6, 13
- legend, 10
- link, 8, 9, 12
- longdesc, 11
- marquee, 11
- Microsoft Word, 6
- OCR, 11
- onclick, 8
- outline, 7
- p, 6
- photosensitive epilepsy, 11, 12
- Photosensitive Epilepsy Analysis Tool (PEAT), 12
- Portable Document Format (PDF), 11
- presentation, 1, 3, 6, 7, 9
- reading level, 6
- screen reader, 8, 12
- Section 508, 1, 2, 4, 12, 13
- seizures, 11, 12
- semantic markup, 1, 6, 9, 10
- skip links, 8, 10, 11
- strobing, 11, 12
- strong, 6
- tabindex, 10
- tables, 9, 10
- target, 6, 8
- tbody, 9
- tfoot, 9
- thead, 9
- title attribute, 7, 8
- ul, ol, dl, 9
- Underline (u), 6
- Unified Resource Locator (URL), 8
- usability, 3, 13
- validation, 4, 5, 13
- Video, 1, 12
- W3C, 2, 13
- WCAG, 1, 2, 4, 13
- Web Content Accessibility Guidelines (WCAG), 2
- World Wide Web Consortium (W3C), 2
- XHTML, 6, 13

