ADA Web Standards and You: A "How-To"

Zach Bingaman

Follow this and additional works at: http://commons.pacificu.edu/inter09

Recommended Citation
ADA Web Standards and You: A "How-To"

Rights
Terms of use for work posted in CommonKnowledge.

This article is available at CommonKnowledge: http://commons.pacificu.edu/inter09/25
ADA Web Standards and You: A “How-To”

Posted on May 1, 2009 by Editor

By Zach W. Bingaman, Site Designer, Berglund Center for Internet Studies

Table of Contents:
Introduction
What is ADA?
What is Section 508 and Section 504?
Designing for Accessibility
Resolution Issues
Liquid Layout
Pixels v EM
Conclusion
Bibliography

Introduction

When learning to design websites, there are a few essentials that most people learn. For example: the “triple constraint” (time, money, and resources), HTML, CSS, client relations, and intellectual property. What most web designer don’t learn is how to design for ADA web standards (no not the American Dental Association). If, in a hypothetical project, you, as the web designer, arrange not to have any legal responsibility for the end project then you’re in the clear. But what if you have a client who asks, “Can you make this site in ADA standards?” Most clients who are aware of ADA standards are also aware are of the possible legal ramifications of not complying. Therefore, if you cannot design for ADA standards, you probably will lose that client. What share here are simple methods in web-design that avoid this issue. I will explain what ADA is, what Section 508 is, and how to design for ADA web standards.

What is ADA?

Short and simply put, ADA is an acronym for Americans with Disabilities Act of 1990. The intention of this act was initially to protect people with disabilities in the workplace by requiring
nondiscrimination standards. ADA prohibits:

“private employers, state and local governments, employment agencies and labor unions from discriminating against qualified individuals with disabilities in job application procedures, hiring, firing, advancement, compensation, job training, and other terms, conditions, and privileges of employment. The ADA covers employers with 15 or more employees, including state and local governments. It also applies to employment agencies and to labor organizations. The ADA’s nondiscrimination standards also apply to federal sector employees under section 501 of the Rehabilitation Act, as amended, and its implementing rules.” (eeoc.gov)

In short the act is directed at any employer who discriminates against a disabled individual by not providing what the individual needs. According to the law, the definitions of a disabled individual are rather broad:

An individual with a disability is a person who:

- Has a physical or mental impairment that substantially limits one or more major life activities;
- Has a record of such an impairment; or
- Is regarded as having such an impairment.

A qualified employee or applicant with a disability is an individual who, with or without reasonable accommodation, can perform the essential functions of the job in question.

Reasonable accommodation may include, but is not limited to:

- Making existing facilities used by employees readily accessible to and usable by persons with disabilities.
- Job restructuring, modifying work schedules, reassignment to a vacant position;
- Acquiring or modifying equipment or devices, adjusting or modifying examinations, training materials, or policies, and providing qualified readers or interpreters.

These are very vague terms, but the actual article lists disabilities as, “caring for oneself, performing manual tasks, seeing, hearing, eating, sleeping, walking, standing, lifting, bending, speaking, breathing, learning, reading, concentrating, thinking, communicating, and working...” (eeoc.gov). The act states that these are “…included but are not limited to…” and thus it puts any documented disability under ADA protection. For employers this can be very intimidating as nearly anyone, for many reasons, can file a lawsuit over discrimination.

In fact, many such suits have been filed: “In Fiscal Year 2008, EEOC received 19,453 charges of disability discrimination. EEOC resolved 15,708 disability discrimination charges in FY 2008 and recovered $57.2 million in monetary benefits for charging parties and other aggrieved individuals (not including monetary benefits obtained through litigation)” (eeoc.gov).
While there are a lot of cases with large payouts, not all of them are website related. In the next section you will see how Section 508 and ADA do go hand in hand to create ADA web standards.

What Are Sections 508 and Section 504?

Section 508 originated in 1998 as an amendment to the Rehabilitation Act (an act that dictates the use of governmental funding). It requires that “Federal agencies make their electronic and information technology accessible to people with disabilities” (section508.gov). This amendment is directed at Federal agencies, but Section 504 is significantly broader: “Section 504 prohibits discrimination based on disability in federally funded and federally conducted programs or activities in the United States, including employment programs” (section508.gov).

According to Section 508 all Federal agencies must make their information accessible to people with disabilities, and Section 504 requires any government-funded organization to do the same. The ADA requires any employer to accommodate disabled people or employees. “The Americans with Disabilities Act (ADA) and, if the government entities receive Federal funding, the Rehabilitation Act of 1973, generally require that state and local governments provide qualified individuals with disabilities equal access to their programs, services, or activities unless doing so would fundamentally alter the nature of their programs, services, or activities or would impose an undue burden” (ada.gov). Through Sections 504/508 and the ADA every aspect of a college campus website (the funding, the employee’s, etc) must be made accessible. But how must a big business with thousands of employees comply?

To better understand Sections 504/508 and ADA standards we need to look at them in application. Currently most non-governmental websites do not fully comply with these standards. The organization W3C attempts to aid web developers to create more accessible websites through their HTML/CSS, and validate programs found at www.w3c.org. But these validated programs do not account for two of the most common problems, resizing issues or resolution differences.

Using the W3C validate programs you can account for users who are blind via alt tags on images, but you can’t account for all of what might or might not be seen as each website is different. The content is generated by the designer and as such can’t be tested for yet.

Let’s return to a college website and a business website to see what this might mean:
I've taken screenshots of a college campus website (as seen above) and compared the navigation bar with itself at a larger font size (which a person with poor eyesight would do; this process will be explained in “Designing for Accessibility”). We’ll call these B.1, and B.2.

I’ve also repeated the experiment with a leading business website (as seen below). We’ll call these A.1, and A.2. Throughout the rest of this paper we’ll refer to these images for examples.
As you can see, the college campus navigation, or how you move through the site (image B.2) has entire navigation sections that are unreadable due to overlapping text and oddly reformatted bars. While it is a good thing that the page will scale vertically, the navigation items will not scale horizontally which can make the page difficult to read. According to the laws, because the college receives government funding and has employees, if someone needed a specific piece of information but cannot find it due to a poor design, that person can file a complaint or a lawsuit.

In A.2 you can see that full bodies of text disappear behind images placed there. Also you can see that full links are lost and unusable. If an employee or a potential customer needed that
piece of information (say a disclaimer which the person could not read) they could not obtain it due to a poor design. That business would thus be in violation of ADA and therefore vulnerable to lawsuits. When asked, “What’s so important about ADA Web Standards?” look at the facts. Designing for accessibility is a necessity.

**Designing for Accessibility**

We’ve defined what the ADA, Section 504, and Section 508 are but not yet their full meaning for accessibility. Joe Clark writes in his book *Building Accessible Websites*:

**Accessibility involves making allowances for characteristics a person cannot readily change.**

*It’s a simple, sweeping definition. Practical Examples of its application to websites:*

- A deaf person cannot stop being deaf when confronted with a soundtrack
- A blind person cannot stop being blind when confronted with visual words and images
- A learning-disabled person cannot reset the function of the brain when confronted with the same.
- A person with a mobility impairment cannot suddenly begin to move when confronted with a navigation task.
- A unilingual Anglophone cannot suddenly understand French when confronted with that language.

Clark is saying is that because a person is disabled we cannot expect them to be able to perform at the same level as a person who is not disabled. Our responsibility as designers is to accommodate our entire audience.

In Jakob Nielsen’s book *Designing Web Usability*, he opens his chapter on accessibility with, “The Americans with Disabilities Act and similar laws and regulations in other countries often mandate equal access to computer systems for users with disabilities. In particular, there will often be a legal obligation to facilitate intranet use by disabled employees who may not be able to do their job if they cannot access their company’s internal website” (297).

What Nielsen has outlined in this statement reinforces the information provided in the previous sections of this paper. The ADA and Sections 504/508 are part of the law, and as such there are real consequences to not complying with them.

The truly big question yet to be asked is how to start complying with these laws. Organizations such as the W3C have launched programs such as the WAI (Web Accessibility Initiative; found at http://www.w3.org/WAI/). These sites provide guidelines as to how things should be done to accommodate disabled audiences. However the truth is that **not every** guideline can be perfectly followed and even the **best** designed sights **cannot accommodate** everyone perfectly.

Any experienced business will tell you that while you can’t have the entire audience, you can
target a specific one. To apply this theory to web design start being accessible by designing your site for one disability at a time. In Jakob Nielsen’s chapter on accessibility, he outlines each relevant disability and how to consider them for design. The Internet as we know it is a visual media. To follow Neilson’s model consider visual, auditory, speech, motor, and cognitive disabilities. The key to each of these is simplicity. Simple is always better when it comes to accommodating people with disabilities.

Designing for visual disabilities should be the first priority when making a website accessible. Because the Internet is a visual medium, a website should be able to accommodate a user if they have poor eyesight or even no eyesight at all. In the situation of poor eyesight the website should be tested using the method I used to test websites A and B.

In Mozilla Firefox, this means resizing your text by going to the “Tools” option then to “Options” on Windows, and on a Mac, to “File” and then to “Properties”. Under the content tab click the “Advanced” button in the “Font & Colors” section. Once there you can change your minimum font size to the highest setting (the worst case scenario) and view your website.

But in Internet Explorer (IE) 8 you can increase your font size via “View” and “Font Size.” However, as most designers know, IE in any form leaves out necessary functionality, so test the text-resizing in as many web browsers as possible. You may notice that there will be overlapping text and/or broken items on the page.

To fix these problems look to the “Liquid Layout” section. Text on a website should be spaced evenly through CSS. Use style rules such as line-height and readable fonts such as Verdana. Avoid using letter-spacing if possible as the resizing of characters will push the spacing, making it unreadable. Also avoid high contrast backgrounds, and, if possible, make sure your font color is easy on the eyes but still in contrast with the background. Fonts are normally black on a white background for good reason.

In the case that a member of your audience cannot see at all they will most likely be using an audio screen reader. The program/device will read the HTML excluding style rules and code, like one would read a book. However, the reader will not be able to interpret images or flash items. This is where you must use the Alt attribute for images. The Alt attribute outputs text enabling a screen reader to output a description of the image. A proper image tag with the Alt attribute might look like this:

```html
<img src="image/bl.jpg" alt="Image of a website's navigation">
```

Be sure in the Alt tag to be terse but specific about what the image is, but if notice above that there is no height or width attribute applied to the image. In some images there may be text used that must be resized for audiences who can see but have poor eyesight.
Through CSS you can resize images through the use of EM’s. To view how to resize images using CSS look to the section “Pixels v EM”. Returning to Alt tags, as long as a proper alt is applied (it must state something about what the image is) it should pass through a W3C validation, thus meeting ADA web standards. Jakob Nielson argues that not all images need a description in an alt tag, “Even though the rule of thumb is to provide ALT text for all images, there are in fact some images that are best annotated with empty string. If an image is purely decorative and has no meaning other than to make the page look better, then there is no reason to slow blind readers with having to hear an explanation” (Nielson 306). It is up to the designer to decide which is best, but even if the tag says it’s a “Decorative Image” the user will know in general terms what is on the page.

Auditory and speech disabilities are the second and third issues to consider when designing a website. With auditory disabilities the user would be unable to hear sound, and thus a transcript should always be produced when sound is played. Avoid playing music on the page unless necessary as it tends to lower the quality of the design, and the site would work the same without it.

Speech disabilities should be an issue only in the case that speech is used in the design such as in the exchanging of messages. Programs such as Skype carry speech capabilities but also allow for text input. For a web design, if you are including a speech form of input, be sure to include an option for text input.

Cognitive and motor disabilities are related in that with motor disabilities the person is unable to perform complex button clicks and with cognitive disabilities the person is unable to comprehend a complex design. To accommodate these disabilities you only need to use the KISS rule: Keep It Simple, Stupid. A good but simple design can still make almost any website amazing.

The website should be appealing to the eye, but not too contrasting. Simplify any items on your site as possible. For example, if you have a news feed make sure your titles are 2-4 words maximum with 2-3 line descriptions. If making an additional page will free up a cluttered item then this will improve design. Also note that it should take no more than three clicks to reach any item in the website.

Finally, make sure that your website looks the same across all browsers. Implementing a browser CSS reset code (such as the one provided by Eric Meyer at meyerweb.com) would be the first step to do so.

Resolution Issues

During the development of my first site, I was initially concerned with cross-browser compatibility. To do this I simply used two computers. One was a Mac loaded with every browser I could find, and the other was a PC also loaded with browsers. When I first tested the site, I did it on the Mac because I designed the site on that computer. But when I viewed it on the other computer, I noticed that some items shifted and moved.
Websites also tend to look different based on a person’s screen resolution. You could build your website in Adobe Flash and write a line of code for a resolution check, but for those who are not very familiar with Flash, there is the additional option of Liquid Layouts and a lot of testing. If you can, get two computers like I did and test them at three different resolutions: a low (800×600 minimum), middle (1024×768 minimum), and as high as you can get (2800 x2100). Check DIV placements to make sure everything is aligned as it should be then check images to see if they overlap. When using liquid layouts, a high resolution will sometimes make a font smaller. If the font size is read as being smaller, the sizing of the DIV can change and thus images may overlap.

The key issue with different resolutions is being able to keep the majority of content on the smallest resolution without requiring major scrolling, while also being able to limit the amount of blank space on a high resolution screen. Blank space is not a bad thing as long as the site is still appealing. But, with a site that looks tiny, a professional may have problems differentiating the blank space from a lack of content or design.

**Liquid Layout**

Liquid layouts are both a godsend and a nightmare because they require a balance between being fully liquid and being static. What you, as a designer, need to do is to build aspects of your site to be scalable and recognize what elements needs to compliment the scaling. When using HTML and CSS, your best tools are auto values, percent values, and EM values. The simpler a website’s design is, the easier it is to make its layout liquid. There are many different methods to creating what is commonly known as a liquid layout, but the most effective method I’ve found is to begin with your DIV layout. The Internet is a visual media, so rather than looking at code created by professionals, consider the box model for DIV’s as a physical box which you are building in.

Remember that DIV items organize HTML items in boxes. You must use your space wisely. The methods for cognitive/motor disabilities dictate that you should keep the layout simple. Attempt to limit the design to a single navigation item and create secondary navigations as necessary. The previous section “Resolution Issues” explains that the design should have the majority of its content fit in a minimum resolution (about 800×600), but still fill a medium to high resolution screen. If properly laid out, the DIV items should feel like building blocks that push and pull on each other. Using CSS, your placement of each DIV should be able to scale with the resizing of DIV next to it.

With every DIV nicely placed, the next step is to consider the content inside each DIV. In your prototype note where text will be, where images will be, and finally to what extent you need those items to scale. Consider what will be visually unreadable at a large scale and, once you input content into your site, test resizing it to see where it does break. Do this because you need to decide what images must be resized via CSS, and what images must be resized via DIV, and how. Does a DIV need to be able to expand horizontally and/or vertically? If this DIV expands, then what DIVs around it need to move and how? It is these changes that truly create a liquid layout.
Here is an example of CSS designed for a static layout:

```css
#nav {
    background-color:#FFFFFF;
    width: 181px;
    height: 910px;
    float: left;
    position: absolute;
    margin-left: 35px;
    margin-top: -200px;
}

#subject {
    background-color:#FFFFFF;
    width: 480px;
    min-height: 548px;
    line-height: 1.25;
    /* height: ; */
    float: left;
    margin-left: 253px;
    padding-top: 25px;
    padding-left: 40px;
    padding-right: 40px;
    padding-bottom: 30px;
    position: absolute;
}
```

This is an Example of CSS altered to become a liquid layout:

```css
#nav {
    background-color:#FFFFFF;
    width: 11.31em;
    height: 910px;
    float: left;
    position: absolute;
    margin-left: 35px;
    margin-top: -200px;
}

#subject {
    background-color:#FFFFFF;
    width: 30.00em;
    min-height: 548px;
    line-height: 1.25;
    /* height: ; */
    float: left;
}
This code was taken from my most recent project which can be viewed either at its home site (http://bcis.pacificu.edu/) or my web design blog at (http://zachwb.blogspot.com/). Be sure to reference this code as an example for the remainder of this section.

But before you go too far, let’s discuss what the “position:absolute” attribute in the code. Most web designers will notice that this value makes the item line up inside the DIV properly. You can use the absolute or the relative values, but it comes down to trial and error in the code. Notice that certain aspects of the static layout have to stay the same to keep the layout from breaking the design. When resizing text you can allow for a DIV to expand horizontally. However, if there are important items to the right of that DIV you may want to apply a max-width. Also it’s a good practice to test and see what the minimum width needed is, based on screen resolution and font size. Sometimes you might find that an image will overlap the boundaries of the DIV which leaves you with a few options. If the image is one you inventoried as being unable to scale then you need to either make the original image smaller or move to your minimum width/height (min-width/min-height CSS rules). If you inventoried the image as one you could scale (due to its use of text), then you need to apply an ID tag to the image like below:

```
#b1 img {
    width: 6.13em;
    height: 9.06em;
}
```

By applying an ID tag you have just forced HTML to recognize the image as a single individual item. HTML will think there is only one “b1”. You then use this to your advantage by creating a unique style rule for “img b1” that might look like:

```
#b1 img{
    width: 6.13 em;
    height: 9.06em;
}
```

A set width and height through CSS allows you to replace the necessity to hardcode the size of the image in HTML (where the image is only sized in pixels) with an EM value that allows the
image to shift with the resizing of text. If the image resizes with the text, and the DIV resizes with the text, then the DIV will resize to fit the image. Congratulations! You’ve just learned how to resize images using CSS. But now let’s move on to the critical part of a liquid layout.

Once the content in your site is set up to scale based on the font size, you’re last concern is making sure that each DIV moves appropriately to the others when scaling. Usually, when you place a DIV item on a page through CSS, the value “float:left” is most commonly used along with “margin” to place the DIV in the right location. If you have placed all the content in one containing DIV, you can apply “margin-left:auto” and “margin-right:auto” to place the site in the center of the page automatically. This allows the rescaling of the site to fit the browser regardless of the resolution of the screen.

The container can be scaled by adding EM values to the width of the container, but I suggest that no values are applied to the height so that content can bleed down the page. Inventory what items need to scale and in what direction, and note where there are DIVs next to them. Where DIV items are located next to them, is where you are going to change the pixel values you used to initially place them to EM value. By doing so, as the DIV next to the item scales in that direction, the DIV will move equally in that direction. As you can see, with this simple rule applied in the CSS (as shown above) the items will be able to scale as necessary.

The last step in a liquid layout is to inventory what items cannot move, for whatever reason. For example, you have three major DIVs on your website: a navigation bar, a content bar, and a news bar. Your employer has made it clear that the news bar needs to be able to be seen no matter what, because it is a big selling point. But when resizing the page, the navigation and the content will push the news off the page into a horizontal scroll and thus not be readily noticeable.

In this situation you are forced to allow the navigation to resize because it is necessary to navigate the website, but you are not required to horizontally scale the content because the content can afford to vertically scroll. Thus you are forced to either set maximum width to the content, or set a specific width (highly suggested) that will stay static through the use of pixels rather than EMs. This brings us to another big question: Pixels or EMs?

**Pixels v EM**

When designing a website to have a liquid layout and still be functional the big question becomes Pixels or EM? The truth is, a more functional design will use a combination of both. When initially designing your website, build a prototype in the size and dimensions at the web resolution (72dpi) you intend to use. Programs such as Adobe Photoshop have measurement tools (the ruler tool in Photoshop) which will allow you pull specific values. These values will allow you to place each item as it’s been build.

A good strategy for building for text scalability is to initially build the site using Pixel values. Hard code the placement of everything and pay attention to which placements rely on each other. For example, the images in our graphic A.1 should be treated as one item inside the DIV, which
should move based on the movement of the font or the DIV above the images. In B.1 each navigation item should be treated as an individual box which scales equally with the text. To do this, size the items with static height and width CSS values.

Once the item is sized and placed as you want it, you can use an EM converter to convert pixel values to EM. I suggest the Em Calculator (http://riddle.pl/emcalc/), but there are others available. To those of you who know that EMs are based on font size, notice that you can change the default font size and the decimal places shown. The EM conversions are based on the closest font conversion so the values, while not precise, are close enough that the changing of font will scale the values as necessary.

**Conclusion**

When confronted with the design challenge of ADA web standards, it is important to know what ADA is and what laws are related to ADA. Sections 504/508 work hand in hand with ADA to encompass almost every possible audience member with a disability. The legal definition of disabled is so broad that a designer must do everything possible to design his or her website to accommodate these disabilities.

Thankfully, the laws are not specific enough to say how they must be accommodated but simply that they must be accommodated. This gives the web designer a chance to break disabilities into the broad categories listed in “Designing for Accessibility”. Designing a scalable simple website that includes transcriptions when and where needed will fulfill ADA web standards. Planning your website layout is critical and referring to the basics can lead you to a better design.

Choosing when to use hard coded pixels or the flexible EMs becomes an art form, but, in the end, it all comes down to planning, hours of work, trial and error, and testing. If you’re still questioning why you bother, remember that there have been “19,453 charges of disability discrimination. EEOC resolved 15,708 disability discrimination charges in FY 2008 and recovered $57.2 million in monetary benefits for charging parties and other aggrieved individuals (not including monetary benefits obtained through litigation)” (eeoc.gov). To be a better web designer, a better boss, or just a better client, you are obligated to educate yourself on these laws.

**Works Cited**


This entry was posted in Uncategorized by Editor. Bookmark the permalink [http://bcis.pacificu.edu/interface/?p=3626].

6 THOUGHTS ON “ADA WEB STANDARDS AND YOU: A “HOW-TO””

Poster Name: Leslie Lyons
Message: I work for a Web design company located out of Louisville, KY and, as an aspiring Web designer; I really want the opportunity to further my knowledge in the field on my own time. I was wondering if you have any recommendations on some resources that will go into further detail on the points covered here. Or maybe you plan to write more on the individual points? I really want to learn more about liquid layouts and pixels vs EM in relation to ADA Web standards. Thanks!

Poster Name: Zach Bingaman
Message: Hi Leslie, Sorry for the delayed response. Most of the methods listed above are more “learn by doing” so I would suggest that you reference Alistapart.com and/or lynda.com. AlistApart is an amazing web developer resource as the articles are written by the people who first envision these uses for css/html. Lynda.com is an invaluable tutorial tool which may help give you ways to practice. These will help with items like EM vs Pixels and liquid layout, but for further understanding of ADA Web Standards I would encourage you read W3C\’s WAI initiative listed in the resources of the paper. It\’s a good read of how things should be, but it does tend to get a little unrealistic with technology constraints. As a rule of thumb, “not every guideline can be perfectly followed and even the best designed sites cannot accommodate everyone perfectly".
Editor  
on May 1, 2009 at 3:44 PM said:

Poster Name: MichaellaS  
Message: tks for the effort you put in here I appreciate it!

naija  
on February 4, 2014 at 10:19 AM said:

I’m quite enjoying the theme/design of your site. Do you ever run into any browser compatibility problems? A little number of my blog audience have complained about my web site not operating correctly in Explorer but seems great in Firefox. Do you have any tips to support fix this issue? I am curious to study what blog method you are utilizing? I’m having some tiny security problems with my latest site and I need to find a thing far more safeguarded. Do you’ve any recommendations?

nigeria entertainment news  
on February 4, 2014 at 10:30 AM said:

moncler piumini

nigeria entertainment news  
on February 4, 2014 at 10:39 AM said:

Pretty section of content. I merely stumbled upon your web site and in accession capital to say that I get actually enjoyed account your blog posts. Any way I will be subscribing inside your augment and even I accomplishment you entry persistently rapidly.