

# Visual28

Personal Blog of Mark Aplet - Design, web, and other topics

---

## **Common Accessibility Mistakes**

Posted By: Mark Aplet [7 Comments](#) January 28, 2010

I gave a presentation at the Sacramento Web Standards Group about improving your sites accessibility. When I began the presentation I full expected to show lots of code. But what I ended up with was a list of common mistakes made when trying to implement accessibility into a site.

Little would I know that this list sparked some excellent conversation for the evening and maybe—just maybe—changed the way a few people create websites, and that is good for everyone.

So here is my general list of accessibility mistakes and a few ways to improve the overall accessibility of your site for everyone and not just disabled users.

### **Verbose ALT Text**

Adding alt text can sometimes decrease accessibility of your site especially when the text is overly verbose, or not relevant. WebAIM has a great article on the [Appropriate Use of Alternative Text](#)

### **Adding default text to a form field**

When ever possible a default value in a form field should not be used unless the form is being populated with known or expected information that is considered an acceptable default answer. An example of an improper use of of a default value would be a search field with the default value of "search" forcing you to clear the default value before adding your own value. Some developers try to compensate for this by adding :focus event handlers in an attempt to clear the value for you. Improper implementation of this has lead to a lot of headache for able bodied users, let alone the severe accessibility issues it presents for some assistive technologies.

### **No Fieldsets**

Fieldsets group related form content together. This can improve accessibility for regular users to your site as well as impaired users. Also, if your website is accessed with a browser that does not support graphics and css, the form will still make sense. The added benefit is that by using a fieldset you can actually add more hooks to your HTML allowing you more flexibility in styling your form. See also the article titled "[Prettier Accessible Forms](#)" by A List Apart

### **Access Keys & Tab Index**

These two items are really awful. There is no way to properly implement them and so they should not be used. According to Web AIM, "The benefits of accesskey and tabindex rarely outweigh the problems they introduce." By using access keys or tab index, you override keyboard shortcuts used by screen readers and tab navigation users.

## Creating your site as a list

In my testing with screen readers, sites that are coded completely as a series of nested list items will cause some screen reading software to choke up and slow down as it parses all the content. I believe this may be due to the way lists are read in that the reader must keep a memory of all parent items.

## "Click Here" links

It's been proven, that links make more sense when they are in context. As such we should avoid uninformative link text like "Click Here" and "Read More"

## CAPTCHAS

CAPTCHAS, are waste of time and effort. They are not secure easy to hack, and impose such heavy accessibility and usability issues that it's not really worth it to secure a form or page with them. The best method I have found for getting around this is to use CFFormProtect and Comment Moderation. CFFormProtect is a free tool that is invisible to users but has a very sophisticated system for detection and filtering form responses.

Rather than making users jump through hoops to leave a comment, or fill out a form, let the software do the heavy lifting and then manually filter the rest.

## Dependencies on scripts or plugins for navigation

This applies mostly to flash based navigation's, or navigation's built in DHTML or entirely in javascript. Although this may work fine for able bodied surfers or surfers with modern browsers, it imposes accessibility challenges for users with disabilities, or users of alternate browsers like phones, or PDA's. If you must use scripts or plugins for navigation care must be taken to provide alternate means of navigation.

## Sniffers for scripts and redirects

Some site developers are attempting to account for the massive number of browsers, and platforms, that they are "sniffing" or attempting to determine what compatibility a users browser has. Although this can be beneficial in some cases to deliver the right content to the right device, it sometimes fails. If your redirecting users to a different version of your site, then please provide them with a way to get back to the original version of the site.

## Text only versions for accessibility

I think there was this idea that a text only version of your site was the most accessible format that it could be in. However testing has proven that this is simply not true. Often a site is less accessible. This article by Web Credible titled "Separate text-only version? No thanks!" is a great read as well as "Design Considerations: Text-only Versions" by Web AIM is also very good.

## Acronym & Abbr Misuse

Acronym & Abbr only needs to be defined the first time they are used and not for commonly understood items like HTML and CSS.

## Missing or inappropriate page titles

Aside from the obvious slam on a lazy or non detail orientated developer, missing page titles or inappropriate page titles should be avoided at all costs. This goes beyond just an accessibility issue, this is a usability issue for everyone. Bookmarking a page only to discover you can not recover it, because the developer never put a page title in, or the page title has nothing to do with what your looking for. This can work against you too if you try and put too much information into the title that may be irrelevant. You will often see this in sites that try to name every product they sell in the title or the title appears to be more of a sales pitch or marketing slogan. Stick to relevant meaningful page titles and everyone will be happier.

## Opening a new window

Opening links in a new window can cause frustration for many users, not just those with disabilities. Opening links in a new window was something marketing people wanted to implement to supposedly keep your website open and available to visitors. The problem is that it breaks users expectations of what happens when they click a link. This article by SitePoint titled "[Beware of Opening Links in a New Window](#)" lists 5 good reasons not to open links in a new window.

## Confusing links

Humans are creatures of habit. We look for links to be of a consistent color, we look for underlines, or other visual clues that tell us that there are links on the page. Making links the same color as the body text, removing underlines, or making the color not stand out as a link are all ways to decrease usability of site links. That's not to say people cannot figure it out, because they can and often will. But at a slower rate if at all in some cases. If your going to change the link colors, being consistent with the colors will dramatically help improve usability on many levels.

## Poor contrast

Not everyone is born with perfect vision, and some people need a higher contrast level in order to distinguish say text from the background. Making sure your text is somewhere around 5:1 contrast ratio will ensure that it's readable by the greatest number of people. Contrast alone is not the only factor in readability. Text size and font selection also make a big difference. If you stick to a high contrast ratio you should be good to go under most circumstances. Web Aim has a [contrast ratio checker](#) but I'm more partial to the [Snook.ca Colour Contrast Tool](#)

## Main navigation links go directly to PDF's

When users click main navigation links they expect to go to a new page. its for this reason that linking to a PDF breaks the users expectations, and ultimately lead to an unhealthy dislike for your website. Also, depending on user preferences, PDF's may not open in a users browser window, but instead downloaded to their hard drive. This presents two challenges. First users don't always know that a file downloaded and may try clicking several more times thereby downloading the document several more times and finally give up and assume there is something broken on the site. The second challenge users face, is understanding their file system. Some users may not know where a file was saved to or where to find the file once it is finished downloading. Ultimately, a user will only try so hard to read what you have to say before they give up, and become dissatisfied with your website.

There are many more reasons why we shouldn't link to PDF's but I think the topic is deep enough that I will save it for a full article. For now, please trust me when I say it's a bad idea.

### Related Entries:

- [Applications to help you create more accessible designs](#)
- [Accessibility Abuse](#)
- [Are acronyms hurting your SEO rank?](#)

Tags: [Accessibility](#) · [usability](#)

### 7 comments so far ↓

- [1 Jared Smith](#) // Jan 28, 2010 at 1:57 PM  
Good stuff! I don't think my list would be much different. I'm glad the WebAIM resources have been useful to you - and thanks for pointing folks our direction.
- [2 Mark Aplet](#) // Jan 28, 2010 at 2:09 PM  
Thanks for commenting Jared. I hope this is all accurate information. Thanks for the great articles and hard work you do over there at WebAIM. The more people we have fighting the good fight the better the web will become.
- [3 Ryan Benson](#) // Jan 28, 2010 at 3:50 PM  
Very nice Mark, on the side of helping people, it would be good to say what they can do to avoid them. While you provide links/feedback to some, they all don't. One that some people would want help with is not using captchas.
- [4 Mark Aplet](#) // Jan 28, 2010 at 4:14 PM  
@Ryan, thanks for the feedback. I elaborated a bit more on the Solution I use for CAPTCHA's as well as linked to another article with more info on accessibility issues with CAPTCHA's.
- [5 Ryan Benson](#) // Jan 28, 2010 at 4:20 PM  
@Mark, welcome. While I love w3c documentation, sometimes it is overwhelming.
- [6 Glenn Weatherson](#) // Jan 28, 2010 at 5:21 PM  
Great round up Mark, I'm guilty of some... A lot of useful information. Hopefully I can practice & remember it, heh.
- [7 Web Axe](#) // Jan 29, 2010 at 12:28 PM  
Great article! I agree with everything.

About "Sniffers for scripts and redirects", the better way to implement is by using progressive enhancement and (unobtrusively) detecting the needed DOM support. Look up Jeremy Keith for more!