Accessibility for web application developers

Beginning tips for people developing applications with web interfaces.

Progressive enhancement

Progressive enhancement is a design and development approach that ensures the widest functionality across disparate and uncontrollable environments. The web, with its millions of users, each with their own computer and browsers environments, is a perfect example. If you use progressive enhancement, you ensure that everyone has at least the bare minimum functionality, even if they are using dial-up with an old browser and an even more ancient computer.

1. First, you develop your web application so that it works reliably in plain HTML. When demonstrating at this stage, be sure to mention that you are still in "ugly mode", because the next step is to...
2. Add the design elements. Whenever possible, the design should be implemented through the CSS so you don't have to touch the already-tested HTML. This is where branding and graphical usability enhancements (white space, color contrast, fonts and font sizes, etc.) are added.
3. The final step is to add client-side scripting to enhance the basic functionality. This can include extras such as client-side input validation. If you add significant new functionality at this step, you must also provide a reasonably equivalent alternative. Be sure that you do not disable any native functionality. For instance, make sure your scripting handles :focus properly.

Resources for progressive enhancement.

Use valid X/HTML and CSS

Up to 90% of accessibility requirements can be met by using validated code and style sheets. Avoid the use of proprietary tags and attributes. Unlike application code, markup code is designed to be forgiving, so just because a page "works in my browser" it doesn't mean it has no errors, and those errors may cause problems for other people.

Validators

- The W3C QA Toolbox - Validators, checkers and other tools for Webmasters and Web Developers [http://www.w3.org/QA/Tools/]

Specifications

- W3C HTML 4.01 Specification: [http://www.w3.org/TR/html401/]
- W3C XHTML™ Basic 1.1 Specification (mobile): [http://www.w3.org/TR/2008/REC-xhtml-basic-20080729/]
- W3C XHTML 1.0: The Extensible HyperText Markup Language (Second Edition) [http://www.w3.org/TR/2002/REC-xhtml1-20020801/]

Text alternatives

Unless an image is decorative (used to set a mood or to enhance the visual experience) it must have text in the ALT attribute that performs the functional equivalent of the picture.

- If an image is used for words (such as "Administration and Finance" at mass.gov/anf) the ALT value must include those words.
- When in doubt, read the page out loud. Does your ALT text sound redundant? Overly verbose? Kind of silly? It may be that it is really is "decorative".
- Decorative images should either be inserted via the CSS (style sheet) as background images or, if inline, should use a null ALT value: alt="".

Videos must be captioned. Audio files must have text transcripts.

- YouTube Help has directions for adding captions to videos hosted there: [http://www.google.com/support/youtube/bin/answer.py?hl=en&answer=100077] A new beta captioning service for YouTube, CaptionTube at [http://captiontube.appspot.com/], has been getting rave reviews.
- In order to create captions, you first need a text transcript. Many people like to have access to this instead of or in addition to the captioning.

Structure and Navigation

Most assistive technologies have features that let users see the hierarchical structure of pages based on the heading tags that are used. A
common feature is a list of level 1 and 2 headings that they can use to navigate on the page.

- Use heading tags to indicate important sections of the page. Make sure the heading levels you choose paint an accurate picture of the structure of the page.
- Do not use heading tags just to achieve visual effects.
- Do not skip heading levels. If you don't like the appearance, you can change it in the CSS.
- You may also want to include "skip to" links that will allow keyboard-only, mobile device, or screen reader users to skip directly to the data. For instance, they might want to bypass an embedded map and go directly to the data table that is displaying the detailed information.

Forms

Forms are more complicated than most people realize, and using form elements correctly helps preserve the native accessibility. The following page has details on "best practices" for web forms that will ensure they are both accessible and universally easy to use.


Tables

At this point in time, you should not be using tables for layout. CSS techniques are stable and well-supported by browsers, give you more design flexibility, and cause fewer problems with assistive technologies.

There are some easy but important tips for creating accessible data tables.

- The SUMMARY attribute of the TABLE tag should give a brief but useful overview of what the table contains and hows it is organized. (Layout tables should never have a summary.) For instance, "Budget Recommendations with one level of row and column headers", or "Cost of different fruits by month."
- The CAPTION tag should be used instead of ordinary text. (If you use a caption, the summary can describe just the table set up.)
- Use THEAD and TBODY to separate the heading area from content area. It makes CSS styling much easier.
- Use the TH tag for both row and column headings. (Yes, you can have both TH and TD in the same row!)
- The TH tag should include the SCOPE attribute, with a value of either ROW or COL. If it is a table heading for a column, it would be <TH scope="col".... A heading for a row would be <TH scope="row"....

Accessibility testing tools

There are lots and lots of testing tools and other advice on the web, but it can be overwhelming trying to figure out which ones are reliable and up-to-date, and what they are good for testing for. The wiki page Accessibility Testing Tools lists and describes various tools. It links to some areas where there a lot of choices.

Accessibility standards

Massachusetts


World Wide Web Consortium (W3C)

- Web Content Accessibility Guidelines 2.0 http://www.w3.org/TR/2008/REC-WCAG20-20081211/
- How to Meet WCAG 2.0 http://www.w3.org/WAI/WCAG20/quickref/20081211/
- Understanding WCAG 2.0 http://www.w3.org/TR/2008/NOTE-UNDERSTANDING-WCAG20-20081211/
- Techniques for WCAG 2.0 http://www.w3.org/TR/2008/NOTE-WCAG20-TECHS-20081211/