Universal Design

for

Digital Media

"Photo Credits:" on page 02

Photo Credits:

Michael Sean Terretta - Everyday Life

National Museum of American History Smithsonian Institute: EveryBody: An Artifact History of Disability in America posters in English (image cropped from original)

Jill Wright - Web Accessibility Word Cloud

Maryellen McFadden - Graphic Design

Introduction

Universal Design is founded on the view that there is a wide spectrum of human abilities and that an individual's abilities change over time and in different situations.[1] [2] A Universal Design approach considers the needs of the diverse range of abilities of users in order to create a design that meets the needs of as many people as possible, including people with disabilities.[3]

This eBook (and, for some of you, accompanying course) will examine the standards and methods for designing digital material which is not only accessible for persons with disabilities but also effective and usable across the broad spectrum of user abilities and technical platforms. In the chapters that follow we will review standards and strategies for achieving Universal Design for websites, focusing on the concepts of usability, accessibility, and Web Standards. During the course of our discussion we will also address related topics such as responsive design, semantic markup and mobile accessibility. Don't worry if some or all of

1 "What is Universal Design?" accessed July 12 at http://www.universaldesign.com/about-universal-design.html.
2 Scott Rains, Universal Design: The Seven Principles [PowerPoiint slides]. Retrieved July 12, 2014 from http://www.slideshare.net/srains/universal-design-the-seven-principles.
3 "What is Universal Design?", paragraph 5.

these terms are unfamiliar – we'll explain them as we proceed through the book (and the course). For now, we will begin by discussing the concept of Universal Design in more detail.

Chapter 1

What are the Features of Universal Design?

Universal Design vs. Accessible Design

Many associate Universal Design with accessibility – that is, making products, buildings and communications, including web pages, accessible to persons with disabilities. Although accessibility is a crucial part of Universal Design, Universal Design is actually more ambitious and comprehensive in its goals.

Designing for accessibility tends to focus on meeting a set of regulations or codes, such as the Section 508 standards[1] for electronic and information technology (EIT)[2] or the ADA Accessibility Guidelines (ADAAG) for buildings, facilities and transportation vehicles. Universal Design aims to go beyond code, to create a design for the widest number of users by anticipating how different types of users will interact with the design.

For a large majority of websites, products and buildings, accessibility is often addressed late in the design or development process, if not after

1 508 is the section of the Rehabilitation Act that applies to federal agencies.

2 The W3C's Web Content Accessibility Guidelines (WCAG) 2.0 are more often used and internationally recognized

implementation. Universal Design advocates for considering accessibility and users' diverse abilities at the start of the design process and for maintaining this focus throughout the development process until implementation.

Accessibility is often accomplished with a separate design or with a separate Assistive Technology product (e.g. screenreaders for audio access to the computer and the Internet for blind or low vision users), a service (a sign-language interpreter translating an uncaptioned video for a deaf individual, for example) or an adaptation to an existing structure or product (a wheelchair ramp added to a building). Universal Design seeks to create one design or product for all users.

The advantages to achieving accessibility through Universal Design in contrast to a separate design or Assistive Technology is summarized in the table below.

Assistive Technology or Special Adaptation	**Universal Design**
After the fact	Before the fact
Individual adjusts to unaltered environment	Alters environment and information
Burden on the user	Burden on the designer
Consumable, with limited use	Used by many in various circumstances
High cost	Lower cost

Table 1.1 Assistive Technology vs. Universal Design[3]

3 Juli Albiero-Walton, Incorporating UDL into Online Curriculum [PowerPoint slides]. Retrieved July 28, 2014 from http://s3.goeshow.com/atia/orlando/2014/profile.cfm?profile_name=download&Handout_key=53163B6C-B7CB-4346-B171-D6A1FE5709A6&xtemplate=1

Real-World Examples

Let's begin with the digital environment. Designing a separate text-only website for individuals with vision impairments, an approach that has been justifiably maligned and decreasing in use in recent years (we'll explain why below), exemplifies a narrow and reactive approach to accessibility – a separate site for a specific population. Conversely, a website developed to address many different audiences and platforms – providing text equivalents for graphic and audio elements, ensuring graceful enlargement of text, separating styling from content to allow user-control of the visual interface and responsive layout for different size viewports (i.e. mobile, desktop and tablet) – is one that takes a Universal Design (UD) approach.

The addition of a ramp to a building is another common example of providing accessibility after the fact. Under the paradigm of UD, access for the wheelchair user (and the delivery person with the handcart and

Figure 1. Skating Ring - Ottawa, Canada (Photo: Mike Gifford)

Ramps added to structures are often unseemly (fig. 1), expensive to install and often do not meet code.

the parent with the stroller) would be considered during the planning and design process. An accessible ramp would be incorporated into the building in accordance with the design of the building and not appear as something added on as an afterthought.

The ramp at right (fig. 2) is clearly steeper than 1:12 (one inch rise to 12 inches in slope) degree angle stipulated by regulations such as the Americans with Disabilities Act Accessibility Guidelines (ADAAG).

Figure 2. Wheelchair Ramp - Shanghai, China

(Photo: Alan [merrionsq])

In a similar vein, accessibility provided through Universal Design tends to be less expensive because accessibility and usability are considered at the beginning of the design process. Retrofitting a building (or product) for accessibility after the concrete has been laid, so to speak, is a lot more expensive than planning it from the beginning. Furthermore, aesthetics and social inclusion are intrinsic to the Universal Design approach. In adopting the perspective that accessibility features should not just functionally but aesthetically integrate with building or product design, UD attempts to avoid stigmatizing the user.

Figure 3. Graham Park

(Photo: Smokey Combs)

Ramps built into the design of a project tend to be aesthetic and allow a low grade incline as shown here (fig. 3 and 4).

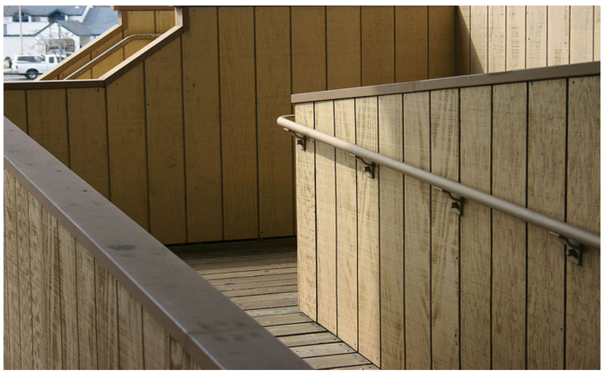


Fig. 4. Wheelchair access ramp to a police trailer under the BART tracks

(Photo: Kai Schreiber)

Text-only Sites: Not True Accessibility

Providing a text-only alternative to achieve accessibility for a website is the digital equivalent of adding a ramp to a building after-the fact – it's an ungraceful and limited solution. The screen shot below shows an example of the text-only version of The Kennedy Center website (http://kennedy-center.org/text). Some of the drawbacks of this approach are apparent when comparing the text-only version to the standard website at http://kennedy-center.org (second screen shot below).

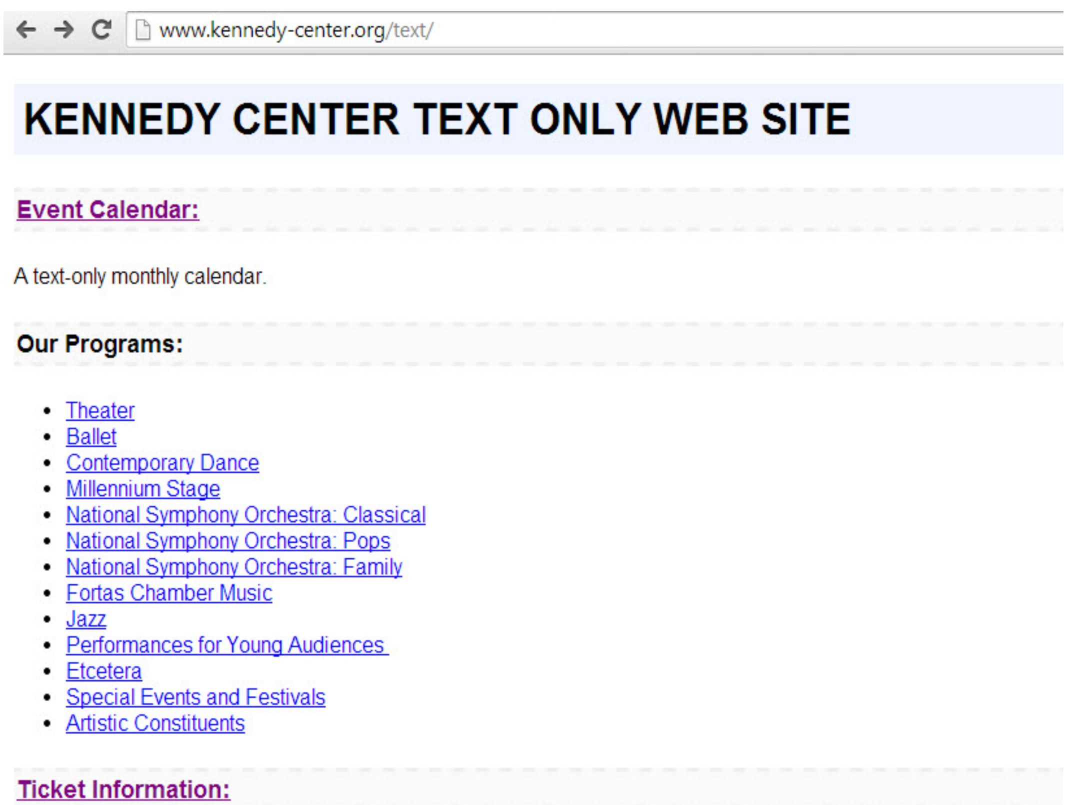


Figure 5. Screen shot of the text-only Kennedy Center site

(Visit http://KENNEDY-CENTER.ORG/TEXT for a "screenreadable" version of the page).

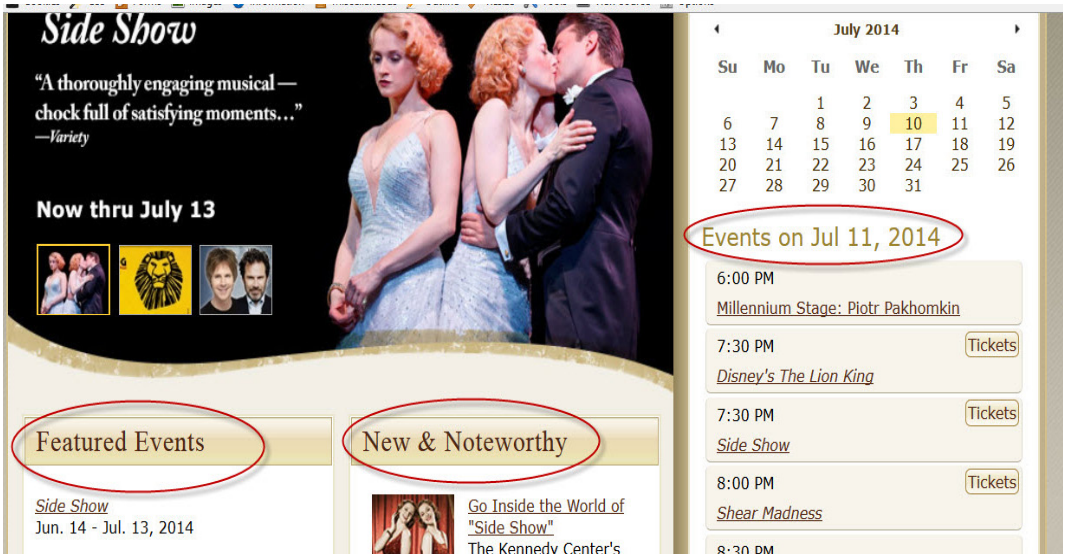


Figure 6. Segment of the standard Kennedy Center home page (Visit http://kennedy-center.org for a "screenreadable" version-of the page).

One significant difference between the two sites is that the more dynamic content found on the standard page – "featured events," "new & noteworthy," or the "events on July 11," (the day I grabbed this screen shot) are not found on the text-only page. In fact, almost all of the links on the text-only page, including "theater," "Ballet," Contemporary Dance," etc., bring you to a list of events in each category that occurred back in 2005 and 2006 – not very useful information in 2014.

The "Events Calendar" and the "Millennium Stage" on the text-only site bring you to current information. Aside from these two areas, however, the information on the text-only site is static (often about past events) – and therefore does not need to be updated. This is a common drawback of text-only sites – they quickly fall out of date because no one wants to maintain them or the organization simply forgets to update the secondary version.

The drawbacks do not end there. A text-only site primarily benefits one specific user type – site visitors who are blind who use screenreader technology (Assistive Technology that reads the text content of a page with synthesized speech). What about those individuals who have vision impairments (poor acuity, for example) or individuals with dyslexia or other reading disabilities (sometimes also screenreader users) who may benefit from graphics and pictures?

Yes, a text-only page (if the content is updated) would be accessible for a certain group of users. But the segregation of a particular audience to a separate site is clearly not in keeping with the tenets or the spirit of Universal Design. It would be a much more effective choice to create one page that is accessible for all users, including the “average” user. Learning how to implement this approach is the goal for this book (and the course).

Chapter 2

Norman's Design Concepts

In chapter 1 we discussed the 7 principles and associated guidelines for Universal Design. So how do we take those principles and use them to build websites that are accessible and usable? As a number of experts have articulated, Universal Design is a design approach, not a checklist or heuristic list of predefined solutions, measurements or products.[1] So how do we build a step by step approach to achieve Universal Design for the web? Partly, that's what we'll attempt to begin to do in this chapter by reviewing basic principles of design as articulated by a number of cognitive scientists and design experts. In particular, we will look at the work of Donald Norman who has had a significant influence on design practices and whose *The Design of Everyday Things* (DOET) (1988) popularized many of the concepts we will discuss in this chapter. Norman's design concepts, as I will refer them throughout this chapter,

---

1 Scott Rains, Universal Design: The Seven Principles [PowerPoint slides]. Retrieved July 12, 2014 from http://www.slideshare.net/srains/universal-design-the-seven-principles.

will provide us with a common and versatile language for talking about design for both physical and virtual environments.

Affordance, Constraints, Feedback, Visibility

In this chapter we will discuss concepts such as affordances, constraints, feedback, visibility and mapping. These concepts are based on research that began in the latter half of the 20th century on how human beings interact with and perceive the operation of devices, objects and interfaces. Key to this complex of concepts and ideas is the notion of affordance – "a property in which the physical characteristics of an object or environment influences its function."[2]

For example, a door knob is designed for hand gripping and pulling, thus it affords turning and pulling; the flat horizontal seat surface of a chair affords sitting, the flat vertical backrest affords leaning our back against the chair. In effective design, the affordance of an object or interface matches its intended function. In poor design, there is a discrepancy between the affordance of an object or device and its intended use. A mismatch between affordance and function creates confusion and difficulty for the user.

Perceived Affordance

Although Norman did not coin the word nor originate the concept of "affordance," he is credited with introducing the concept of "perceived affordance"[3]

2 Universal Principles of Design – loc 410

3 cf xii – the Design of Everyday Things. The seminal work on affordance is credited to James

– i.e. how users interpret and perceive the function of a device. The distinction is subtle. In effective design, the affordance not only matches the intended function, but the affordance is clearly visible to the user.

Most of the time, at least for simple objects (e.g. a pencil, a hammer, scissors), we are not conscious of the cues a device relays to us – this usually indicates sound design. If we have to think about how to operate a device, it usually means that its "perceived affordance" is not clearly visible to the user or that the device's affordance does not match its function. The concept of affordance also applies to electronic devices and computer interfaces. Throughout this chapter we will discuss how both physical objects and web pages can either reveal correct cues for operation or misinform the user.

In later editions of DOET Norman shares how readers have sent him numerous examples of "Norman Doors" or "Norman devices." These are doors or devices whose affordances communicate the wrong cues for operation: a handle on a door that communicates pulling when pushing is required, for example, or a door handle designed for pulling with a wide flat design that signals pushing because of its shape. The pictures below show examples of door handles with different affordances. Do you see any examples of "Norman doors," doors with handles that convey incorrect information to the user?

Gibson (Universal Principles of Design loc 444).

Figure 1. Door handle with cartoon guide

(Photo: Alcook101)

The door handle shown at left (fig. 1) affords pulling. The drawing above the handle provides a secondary cue to pull.

The lever style handle at below (fig. 2) affords downward force to unlock. However, similar to door knobs, it's not always clear from the handle alone whether to push or pull to open the door.

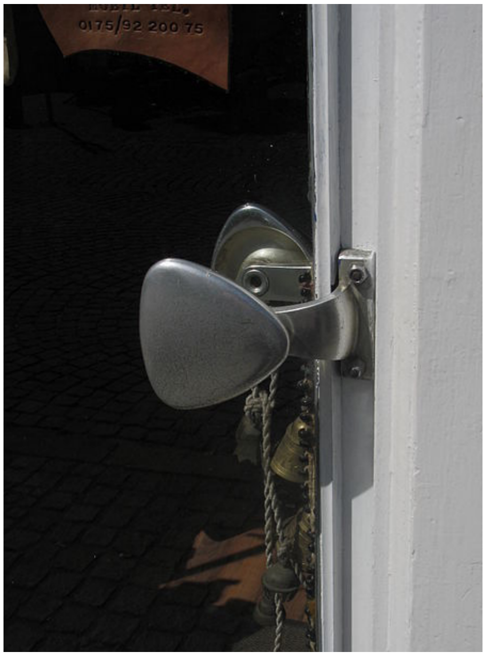
Figure 2. Lever style door handle

(Photo: Kathryn Apland)

Figure 3. Double door with vertical handles

(Photo: Demidow)

The vertical handles of this double door (fig. 3) clearly afford pulling.



` **Figure 4. Flat style door handle from Germany**

(Photo: Demidow)

Although not a common design in the U.S., the flat surface of this design (fig. 4) from a door in Germany affords pushing.

Constraints

Constraints limit the actions of users in order to prevent error and misuse of a device. In contrast to affordance which cues users towards certain operations, constraints limit a system's options in order to prevent error, damage to the device or harm to the user or others. Through limitations constraints guide the user towards correct operation or use of a system. Using some automotive examples, constraints include the inability to remove a key from a car ignition until the engine is shut off; the inability to switch the transmission into reverse while the car is moving forward; and the auditory chime when we try to exit the car with the key still in the ignition.

Different design theorists categorize areas of constraints varyingly. Lidwell, Holden and Butler in *Universal Principles of Design* divide constraints into two categories: physical and psychological.[4] Norman also divides constraints between physical and psychological categories but breaks down psychological constraints further, dividing them into 3 categories: semantic, cultural, and logical.We will examine Norman's classifications of constraints below. psychological constraints further, dividing them into 3 categories:

A lack of proper constraints can be fatal. The following is a headline from the January 25, 2010 issue of the New York Times reporting on a medical radiation device used for cancer treatment that allowed settings that emitted fatal doses of radiation.[1]

A Lifesaving Tool Turned Deadly

Radiation Offers Powerful New Cures, and Ways to Do Harm

By WALT BOGDANICH

As Scott Jerome-Parks lay dying, he clung to this wish: that his fatal radiation overdose — which left him deaf, struggling to see, unable to swallow, burned, with his teeth

Soon after the accident, at St. Vincent's Hospital in Manhattan, state health officials cautioned hospitals to be extra careful with linear accelerators, machines that generate beams of high-energy radia-

and treat their disease. It also inflicted unspeakable pain.

Yet while Mr. Jerome-Parks had hoped that others might learn from his misfortune, the details of his case — and Ms. Jn-Charles's — have until now been shielded from public view

1 Bogdanich, W. (2010, Jan 24). A lifesaving tool turned deadly. New York Times (1923-Current File) Retrieved from http://0-search.proquest.com.bianca.penlib.du.edu/docview/1467504414?accountid=14608

4 William Lidwell, et al. Universal Principles of Design: 125 Ways to Enhance Usability, Influence Perception, Increase Appeal, Make Better Design Decisions, and Teach through Design (Beverly, Massachusetts: Rockport Publishers, 2010), 60.

semantic, cultural, and logical.[5] We will examine Norman's classifications of constraints below.

Physical Constraints

The first two automotive examples provided above are representative of physical constraints: cars are designed with the physical properties to prevent shifting in reverse while the car is moving forward and to prevent the key from being pulled from the ignition while the car is running or not in park (for an automatic transmission).

The 3rd example – the chime warning – is a semantic constraint. The chime is an auditory signal that something errant has occurred. This capturing of the users attention is usually sufficient to remind us that the headlights are still on or the key in the ignition. The warning chime is at least partially culturally based. We have become accustomed to loud repetitive beeps or tones warning us of error or danger – the beeping of a large vehicle backing up, a cooking timer or the loud blaring beep of a smoke alarm.

Auditory cueing is also tied to human psychology – discordant tones are associated with warnings and more pleasing sounds with positive feedback for a correct operation; the two-tone note when MS Windows loads or shuts down, an example of the latter. My flat-screen TV plays a pleasant single tone when I shut it off or on. The feedback is particularly useful when turning it on since there is a delay before the picture appears. The tone

5 Donald A. Norman, The Design of Everyday Things, Reprint Edition (New York: Basic Books, 2002), 84-86.

prevents me from pressing the button again (thus shutting it off), thinking the device did not receive the first signal.

Semantic or symbolic constraints convey proper operation and warn of improper use through language, symbols or auditory cues. (The cartoon of the person pulling on the door in figure 1 above is an example of a symbolic constraint). Cultural or conventional constraints direct appropriate action and discourage misbehavior through learned experience and cultural norms, hence the confusion or anxiety (or alternatively, excitement and stimulation) when we encounter a foreign culture or environment.

Cultural Constraints

We know from convention that we pay for purchases at a cashier, usually at the front of a store (or scattered at different locations in department stores).[6] If we have traveled by plane we know to check-in on one level and pick up our luggage at the baggage carousel at a different level, usually on the lowest floor

"Visibility" for the Blind or Deaf user

Our review of Norman's concepts has taken place within the context of a seeing population. How do we apply the concept of visibility for a blind user? Or what about the "visibility" of sound for the deaf individual.

To extend Norman's concepts beyond usability to include accessibility we need to consider users who lack sensory capability in one or more areas. Feedback and visibility mean something different for the person who cannot see the computer screen. In these cases, "visibility" may be provided through audio output (e.g. ATMs with audio jacks) or through transcripts of audio output for an individual with a hearing impairment.

In module three and throughout the rest of this course we will address how to provide visibility and feedback in a context that is broader than the normalized seeing and hearing population.

6 This convention may change with portable wireless cashiers on smart phones.

of the terminal. These are all conventions we have learned from experience. Similarly, web users have learned through convention that the home link or menu item takes us to the home or front page of the site; that the navigation bar will usually be located near the top in a horizontal layout or on the left side in a vertical display. Computer users have learned that control or command-Z undoes an operation in almost all software.

When we design websites or devices we should ensure we follow conventions. Conventions become "perceived affordances." When we break convention – renaming the "home" button to "front-page" or another variant of the word "home" for example – we cause confusion and require extra cognitive effort on the part of the user, slowing down and degrading the user experience. Breaking convention thus creates a gap between the perceived affordance of a system and its actual operation.

The third and last psychological constraint Norman mentions is "logical." The distinction between logical and semantic or cultural constraint is subtle. A crossword puzzle, perhaps, provides the best example for our purposes. Answering the clues of a crossword puzzle involves cultural knowledge and semantics – the meaning of words, historic events, etc. However, for most of us, there will be entries we cannot answer. If we have answered enough of the puzzle so that only a few letters are missing, we may be able to complete the entry because only a few letter choices would make semantic sense (i.e. form a valid word). Even though we do not have the required cultural knowledge, we are able to deduce the answer through logic. Norman terms this a logical constraint.

The nuances of distinction between these areas of constraint are not important. The overall concept is what is crucial for thoughtful design and to follow the discussion on the concept throughout this book (and course).

Examples of constraints implemented by software or websites include the screen boundaries placed on mouse movement, the warning we receive to save a document before closing a word processor or document window, the warning of lost data when a user selects to navigate away from an online form or the "snap-to-grid" feature of many design programs.

Visibility & Feedback

Norman advises that physical constraints are most effective when they are made visible thereby discouraging even an attempt at a wrong operation. A diagram on a gas-station credit card device can prevent us from inserting or sliding the card in the wrong orientation before physical resistance informs us to reverse the direction of the card. Visibility also has a contrary application – invisibility. Sometimes, we want to hide options that are not appropriate for the current state of the system. We witness this feature in computer software when menu options not available during particular states are not shown or when the non-focused window in an application is grayed out.

Making applicable items visible and non-relevant operations invisible, particularly in electronic interfaces, is key to effective design. With computer software and the web we accomplish this through hierarchical menu systems and organization of information. Presenting users with all the options of a website or computer application on the opening screen would obviously be overwhelming and counterproductive. This would produce what we might term *too much* visibility – too many options to scan and choose from. Instead we break up menus and information into major categories at the top of the hierarchy and reveal more specific options as the user drills down to more specific areas of the site.

The authors of *Universal Principles* term this "context sensitivity." Context sensitivity reveals and hides particular sets of controls based upon context and the particular location of the user in the system.[7] (Context sensitivity is very similar to what's termed "progressive disclosure" in web design circles, a concept we will talk about later).

Another strategy for revealing and hiding information is to present the basic operation of an application or widget, providing the option to reveal more complex features upon user request. For example, it is convention to present a simple search interface for services such as Google with an option to access "advanced search" features. This creates a simpler and easier to use interface, still providing the option for more complex operations for the advanced user.

7 William Lidwell, et al., 250.

Visibility, Feedback, Constraints & Affordance are Closely Related

According to the principle of visibility, systems and devices are more usable when their status and possible operations are clearly visible and the user is advised of the consequences of certain actions before they are performed.[8] Providing feedback and system status once an action has been selected is also crucial.

We can use the example of a laptop computer to demonstrate these concepts. A blank (i.e. black) screen with no lights on the panel, particularly the power light, indicate the computer is in an off state; a blank screen and a flashing power LED indicates the device is in sleep mode; a smiling Mac icon or a "starting Windows" message indicates the device is initializing. It is particularly important to communicate system status during operations that require a relatively long time to complete.[9]

We have all experienced the confusion that occurs when we submit a form, start to download a file, initiate a financial transaction, or perform some other action that produces no feedback from the system. Has the system crashed, should we press the "submit button" again (perhaps paying for something twice)? How long should we wait before trying again or closing the page, or, with a frozen computer screen, power down our device.

8 Ibid.

9 According to Jakob Nielsen, any delay longer than about .9 seconds will be interpreted as a hung or crashed process unless the user is provided with status updates - http://www.nngroup.com/articles/response-times-3-important-limits.

Mapping

The last design concept we will cover in this chapter is mapping. Mapping is the presentation of controls that mirror the operation to be performed or through positioning or proximity, indicate the device or component associated with the control. A "common-sense" example of this is the placement of light switches near the light associated with the switch. We intuitively determine the light switch to operate for any particular light by looking for the closest switch.

A more ambiguous situation is shown for the stovetop controls in figure 6. There is only one burner in the middle – thus it is clear that the closest knob at front middle is the correct control. The two burners on the left, however, have two control knobs placed parallel to the burners. Thus is not clear through proximity which knob controls the front and which the back. This stove, like many, has a diagram (fig. 7) – a semantic constraint – to cue the association between the knobs and the burners. However, the stove in fig. 8 cues the association through the physical placement of the knobs. The knobs set higher back control the back burners. Thus through positioning, the knobs mirror the arrangement of the burners and cue the proper association.[10]

10 Full disclosure – notes accompanying the photo indicate the controls have been moved with Photoshop but you get the point.

Figure 6. Large stovetop with multiple burners

(Photo: julochka)

Burners shown in fig. 6 have ambiguously placed controls. Can you tell which knob on the left controls the back or front burner?

Figure 7. Burner knobs with graphical cues

(Photo: anna)

In fig. 7 diagram cues relationship of control to burner

Figure 8. Stovetop with position of the control knobs mapped to the corresponding burners

(Photo: Peter Rukavina)

Lest we forget that our focus is web design and not kitchen appliances, let's return our focus back to the computer. How does mapping manifest in the digital environment? As we might expect, similar to that seen in the physical environment. `

The radio buttons in the form field below (fig. 9) show the importance of placement and proximity. Due to convention, we have come to expect radio buttons placed to the left of the label. (As mentioned above, convention becomes a form of conformance). Although the radio buttons below are placed closer to their associated label at right, they are close enough to the previous label to perhaps cause some confusion.

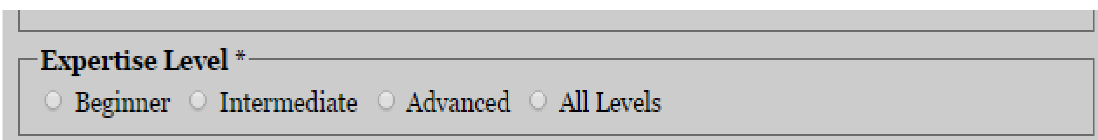


Figure 9. Radio buttons

In fig. 10 we have increased the spacing between radio elements, thus making the association between the button and the label clearer.

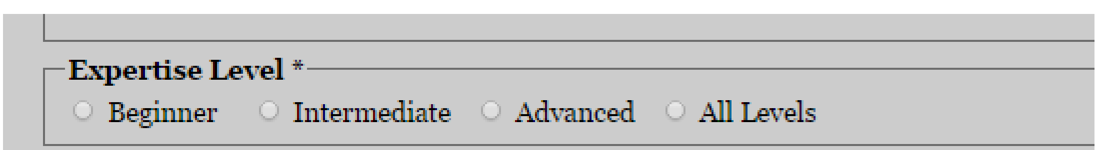


Figure 10. Radio buttons with wider spacing

Software-based controls for volume, brightness and levels in general, map similarly to physical controls. As shown in fig. 11-14, right and up map to higher, left and down to lower.

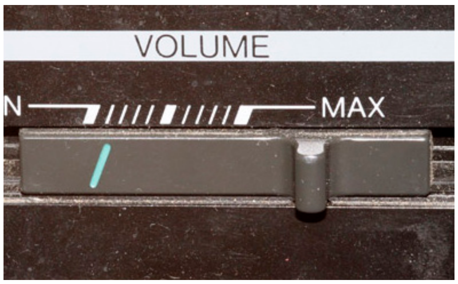


Figure 11. Volume control on radio

(Photo: Jenn Durfey)

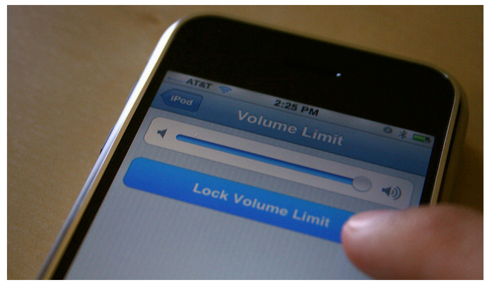


Figure 12. Volume control on iPhone

(Photo: Pieter Ouwerkerk)

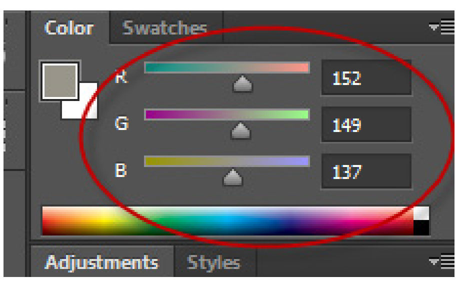


Figure 13. Color levels in Photoshop

Figure 14. Levels on an amplifier

(Photo: Sergiu Bacioiu)

Mapping, particularly the importance of positioning and proximity, will be discussed further when we turn to visual design in Module 4. As we turn to web accessibility and usability best practices, we'll see that Norman's concepts help to explain the reasons why certain practices improve user understanding and some practices inhibit use. Overall, these design concepts provide us with a language to discuss the practices we'll be talking about for the rest of the course.

(Footnotes)

1 Bogdanich, W. (2010, Jan 24). A lifesaving tool turned deadly. N*ew York Times (1923-Current File)* Retrieved from http://0-search.proquest.com.bianca.penlib.du.edu/docview/1467504414?accountid=14608

Chapter 3

Web Standards, Structure, and Semantics

This chapter (and, for some of you, this module of the course) will examine the interrelated topics of Web Standards, structure, and semantics. These three areas comprise the foundation of our approach to designing websites that are effective and usable across the broad spectrum of user abilities and technical platforms. I refer to these as foundational because when we design to standards, implement proper structure, and use semantic markup, we are arguably 60 - 70% of the way to accessibility. Furthermore, as the integrity of a building is dependent upon its foundation and underlying structure, accessible and effective web design is founded upon a reliable, standards-based structure.

What Do We Mean by Web Standards?

Approaching the definition from a technical perspective, Jeff Zeldman, the author of Designing with Web Standards (2009) and co-founder of the Web Standards Project (WAsP), offers the following: Web Standards mean using “structural languages like HTML, XHTML, and XML, presentation languages like CSS, object models like the W3C DOM, and scripting languages like JavaScript."

For those non-programmers scratching their heads, another way to put it is to say that Web Standards means creating web pages with the syntax and code elements that will be recognized by the range of browsers and platforms that are standards-compliant. This includes I.E.9+, Apple's Safari 3.0+, Firefox 3+, Chrome, and Opera 9+.

In it's most basic sense, it means using one of the Doctypes listed in the panel below (fig. 1) at the top of our html or htm file and checking the code with a validator tool to ensure compliance.

Web Standards also encompasses the following: separation of content from presentation – using (x)HTML for structure and content, and CSS for styling; avoiding tables for layout and using semantically correct code (more on this below). It may be easier to explain the benefits of standards- compliant pages by looking at non-standards-compliant pages and sites. Any site that uses proprietary coding is an example of a non-standards-compliant site. For example, Amazon uses proprietary coding for some features of its site, such as checkboxes. (Kathy discussed this in her "Norman Design Concepts" video in module 2).

Valid Doctypes

[XHTML 1.0 Strict]

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

[HTML 4.0 Strict]

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

[HTML 5]

<!DOCTYPE HTML>*

*Note: the current status of HTML 5 is "Candidate Recommendation," a step or two below full recommendation.

Figure 1. (x)HTML Doctypes

Before the adoption of Web Standards by the major browsers in the early 2000s, and gradual acceptance by web developers in the years to follow, many if not most websites used browser-specific proprietary coding techniques. Zeldman discusses at length the "code-forking" techniques (i.e. branching to different sections of proprietary code based on the detected browser) used by developers to allow their pages to run on different browsers. This resulted in bloated code and extra development and maintenance costs (any development and changes to a site required creating code segments for a variety of browsers). Another approach was to code for only one browser. During the late 90s it was not unusual for sites to display statements such as "this site displays best in Internet Explorer."

Although these types of browser-specific techniques and issues are thankfully a thing of the past, non-standards coding appears in other ways. Today, many CMS systems and site-building services are guilty of producing non-standards-compliant code and sites. One glaring example this author has come across is the website builder WIX. WIX previously built visually "stunning" sites using Flash-based technology.

However, with the lack of support for Flash on mobile devices (iPhone in particular), WIX transitioned to creating similarly stunning sites using HTML5/JavaScript. (See below and http://www.wix.com/sample/website for examples). The problem lies in the underlying code and structure. When examined for standards compliance, many of their sites have precious little structure – no headings, lists, paragraphs or other structural or semantic elements so important to accessible design.

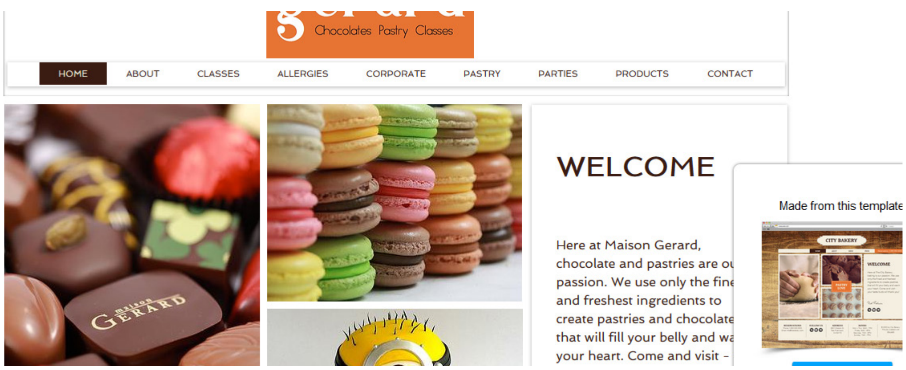


Figure 2. Bakery site example.from WIX

The lack of structure and semantics is not only a problem for screen-reader and keyboard-only users, but also for search engine optimization (SEO) and fluid and responsive design.

Because the sites lack headings, screenreader users, who often use headings to find the major content areas of a page, are unable to effectively navigate through WIX-created sites. In addition, the example page lacks alt text and provides no feedback on cursor focus for the keyboard user.

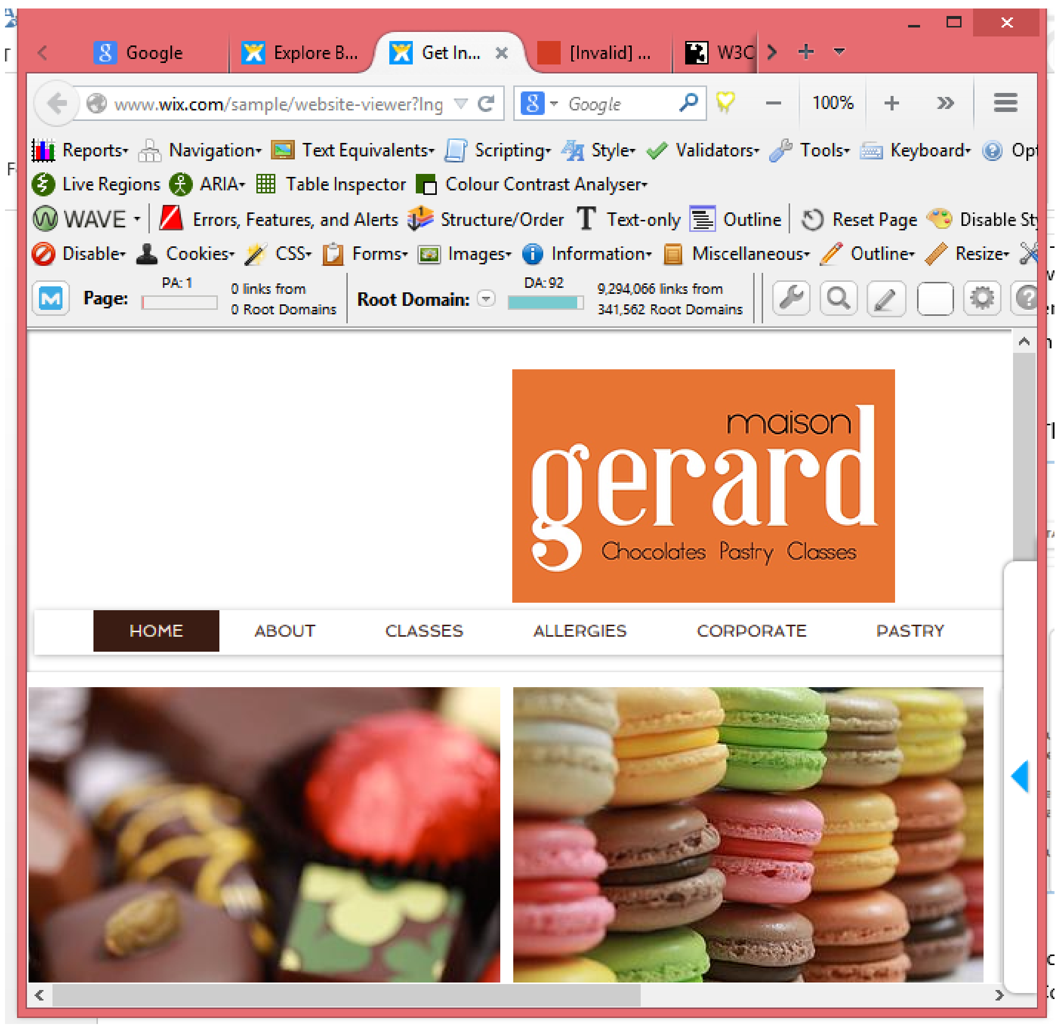


Figure 3. Bakery site example from WIX

Regarding SEO, search engine algorithms weigh the text in headings and titles more heavily than text elsewhere on the page. Thus, a site that neglects to structure its semantic content in readily accessible headings undermines its own opportunities for higher search engine ranking. The lack of structure also precludes the ability to create CSS to allow WIX pages to adjust and format for different viewport sizes and/or different devices. Changing the window size of the browser results in no adjustment to the text and objects within the WIX page – any elements beyond the window are simply cut off (see fig. 3). A smartphone or tablet version of the site would each require a completely separate design.

One of a number of other problems with the site is shown here (fig. 4). The navigation at the top of the page is only mouse accessible – the keyboard

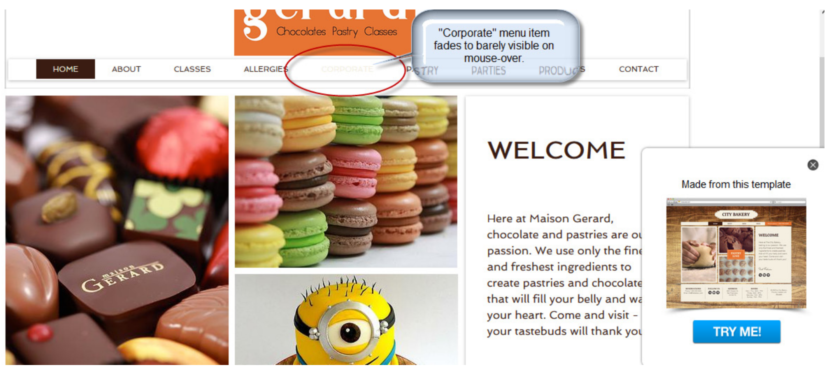


Figure 4. Bakery site example from WIX

[tab] cannot reach these links. Unrelated to the structure or semantics of the page, the menu items almost disappear on mouse-hover, as shown above for the “Corporate” menu item.

What do Web Standards Accomplish?

What are we accomplishing regarding accessibility and usability with Web Standards? Aside from avoiding the problems just mentioned above, we are ensuring that pages will display as intended across the range of browsers that are standards-compliant. It is much more likely that a standards-compliant site will render correctly across browsers, platforms and devices. A non-standards page is much more likely to "crash" or only render correctly on a limited number of code-reading devices.

Here are a list of other benefits of designing with Web Standards:

- Web pages are lighter (i.e. smaller) because the separation of style from content limits the style code to one location instead of repeating on each content page. Smaller pages load faster, use less bandwidth, and save on storage.
- File size is also reduced with the elimination of "code forking" for different browsers – an approach required before standards-based coding took hold. Only one set of compliant code needs to be written instead of three or four for different proprietary syntaxes. Developer time is also reduced since only one set of code is required and the complexity of developing sites to work on different browsers is significantly reduced.
- The pre-standards version of specifying page layout through tables and spacer gifs also increased file size due to the heavier code caused by table markup and graphic elements. Furthermore, presentation accomplished via static graphics (e.g. navigation elements created with images) and layout tables are much more time consuming to edit and update than layout controlled and rendered with CSS.
- The control of presentation through external style sheets allows developers to offer various styles to their users; users can use custom style sheets to render pages in a manner that meets their particular needs – such as enlarged font or high contrast foreground/background colors.
- The same flexibility with presentation, along with responsive design, allows the content of a site to appear effectively on different size viewports and devices, possibly eliminating the need for a separate mobile site. (Visit http://www.csszengarden.com/ to see a demonstration of the flexibility allowed through CSS-based presentation).

- Separate style files improve search engine optimization (SEO) by facilitating cleaner content pages, allowing web crawlers to more easily access and index content.
- Web Standards improves accessibility overall because it is easier for Assistive Technology, such as a screenreader or voice recognition (e.g. Dragon NaturallySpeaking), to interpret the content of a page when the structure and coding follow standards.
- A standards-compliant page encourages accessibility by requiring the inclusion of alt text for image and graphic elements (or a null alt attribute [alt=""] to indicate an image offers no informational content).
- External CSS is a much more powerful for controlling the presentation of a site. Hundreds or thousands or even tens of thousands of pages can be controlled through a central style file or set of CSS files, allowing much easier site-wide changes.

Structure & Semantics

Structure and semantic markup extends and complements the benefits provided by Web Standards. Semantic markup both allows flexibility in the presentation of content and structures the document so that the meaning of content is communicated to both Assistive Technology and to any other code-reading systems that analyze web pages, for example search engine web crawlers.

Although headings structure a page by delineating the start of page sections and (should) indicate the content that follows (e.g. a heading level 2 "staff directory" above a list of employees and contact information), the structural capability of headings are limited because they only designate the start of a section and not the point of closure.

Furthermore, there are often discrepancies between what we may want to convey semantically and visually. For example, the navigation area or

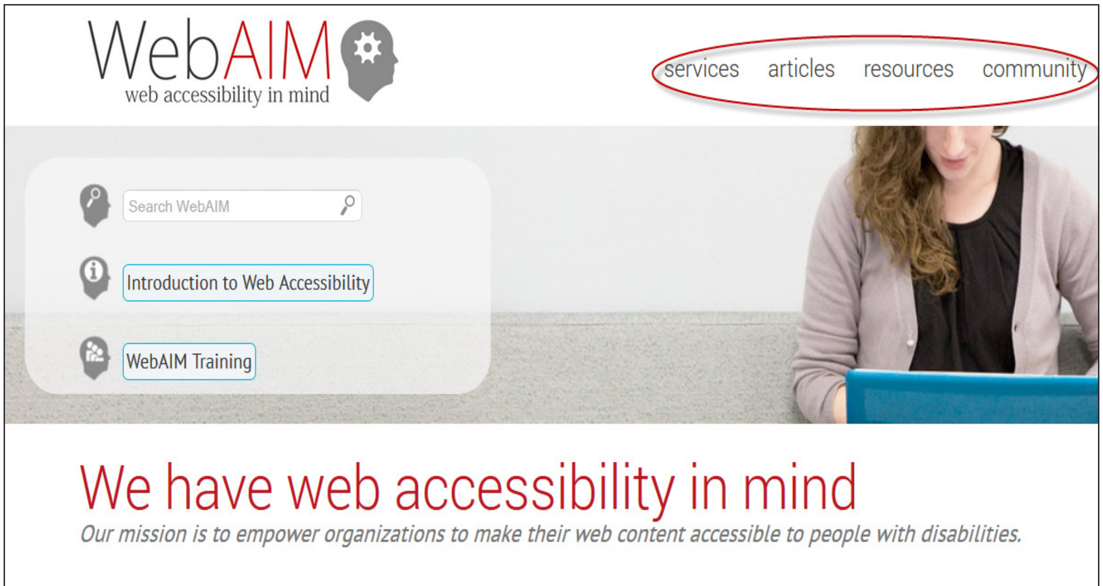


Figure 5. WebAIM home page - note there are no headers (that can be seen) above the navigation menu

search form of a page might semantically call for a heading so that a screen-reader user could easily find (or skip) those sections of the page. However, we usually do not put a visual heading above these areas because convention and other visual cues readily indicate their location and purpose. An inspection of the WebAIM home page offers a demonstration of this issue (fig. 5 below).

Note that the navigation section has no visual heading. However, visually, we have no difficulty locating it because users have come to expect that navigation will appear as a horizontal menu near the top of the page or aligned vertically on either the left or right. However, if we use the WebAIM tool WAVE to display an outline view of the page, a heading level 2 for navigation ("Main Navigation") is revealed (fig. 6 below).

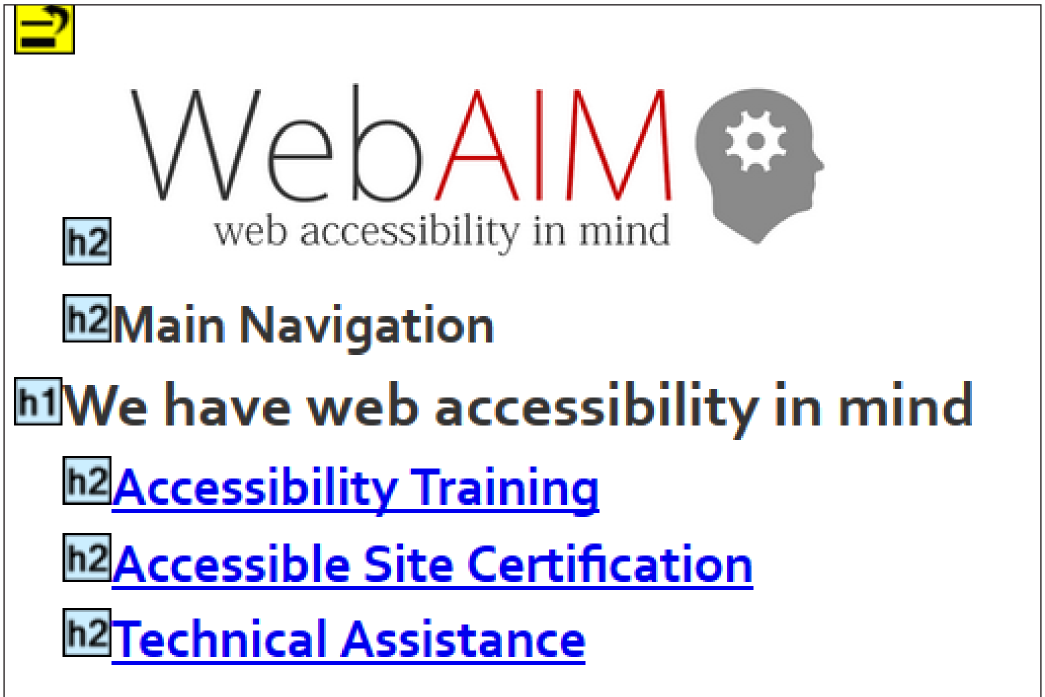


Figure 6. WebAIM home page - outline view shows a level 2 heading for the navigation area

The section of html code below (fig. 7 & fig. 8) and the CSS code in figure 9 reveal that WebAIM accomplishes this by using a class they call "hidden" to hide the heading level 2 "Main Navigation" text.

```
<h2><a href="/"><img src="/media/logo.png" alt="WebAIM - Web
Accessibility In Mind" height="80" width="302"></a></h2>

<nav role="navigation">
    <h2 class="hidden">Main Navigation</h2>
    <ul>
        <li><a href="/services/">Services</a></li>
        <li><a href="/articles/">Articles</a></li>
        <li><a href="/resources/">Resources</a></li>
        <li><a href="/community/">Community</a></li>
    </ul>
</nav>
```

Figure 7. The heading for “Main Navigation” is visually hidden with the ‘class=”hidden”’ attribute

```
<nav role=”navigation”>
        <h2 class=”hidden”>Main Navigation</h2>
        <ul>
                <li><a href=”/services/”>Services</a></li>
                <li><a href=”/articles/”>Articles</a></li>
                <li><a href=”/resources/”>Resources</a></li>
                <li><a href=”/community/”>Community</a></li>
        </ul>
</nav>
```

Figure 8. Text version of code for hidden “Main Navigation” heading

The CSS code for the “hidden” class is shown below.

```
/* Various element styles */

.hidden {
        position:absolute;
        left:0;
        top:-500px;
        width:1px;
        height:1px;
        overflow:hidden;
}
```

Figure 9. CSS code for visually hiding class “hidden”.

```
<div id="headcontainer" class="clearfix" style="background-image: url(/media/banners/home.jpg)">
<header role="banner">
    <div id="skiptocontent"><a href="#maincontent">skip to main content</a></div>

    <h2><a href="/"><img src="/media/logo.png" width="302" height="80" alt="WebAIM - Web Accessibility In Mind"></a></h2>

    <nav role="navigation">
```

Figure 10. Heading 2 tag enclosing banner

The CSS styling for the class “hidden” hides the text by placing it 500 pixels above the screen. (Note that you should not use “display: none” to hide text because it would hide it from the screenreader – and any other code-reading technology accessing the page).

The WebAIM banner (the top element in fig. 6 above) is also included in the heading structure of the page, enclosed within an <h2> tag (see fig. 10).

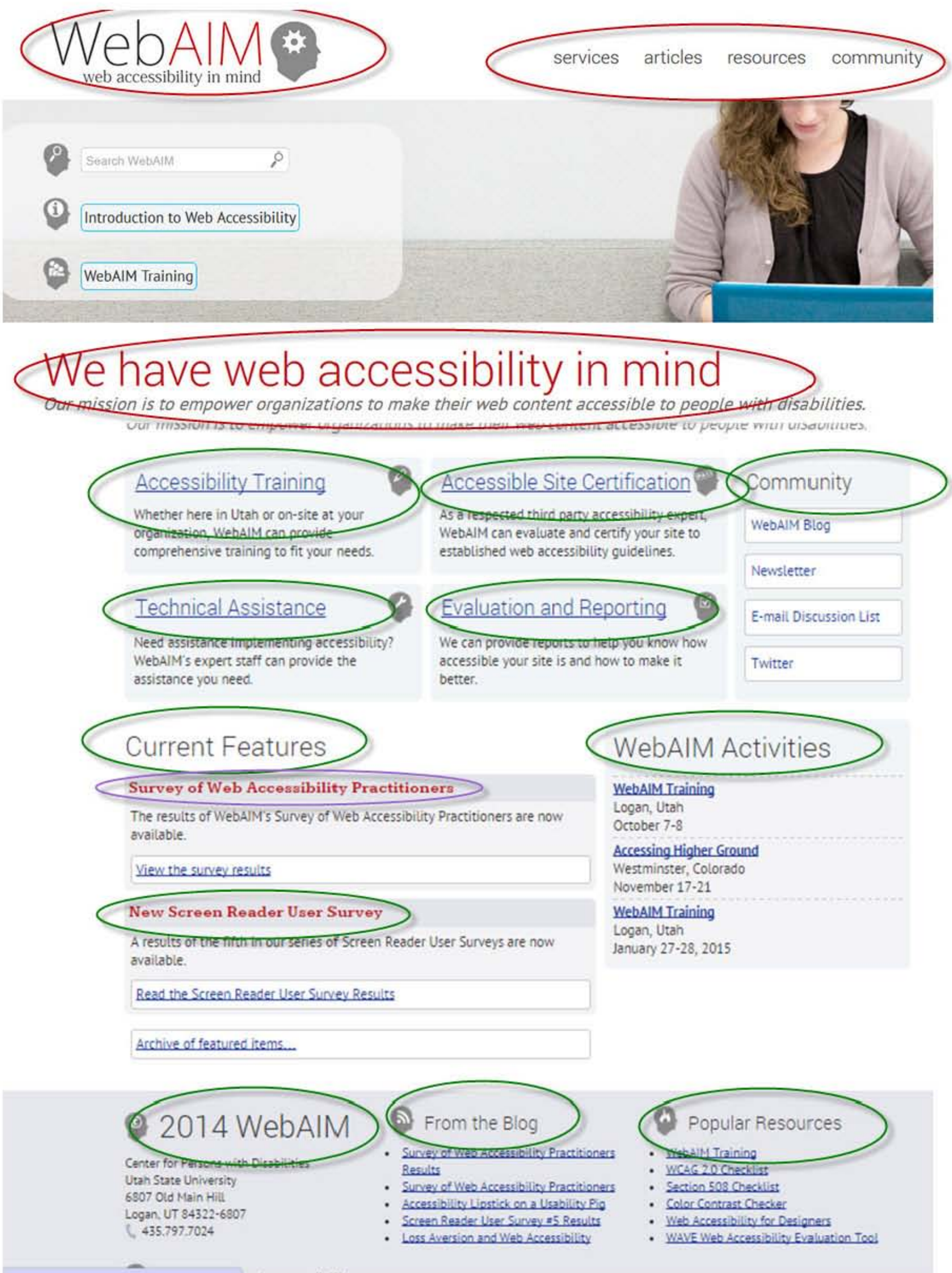


Figure 11. WebAIM home page

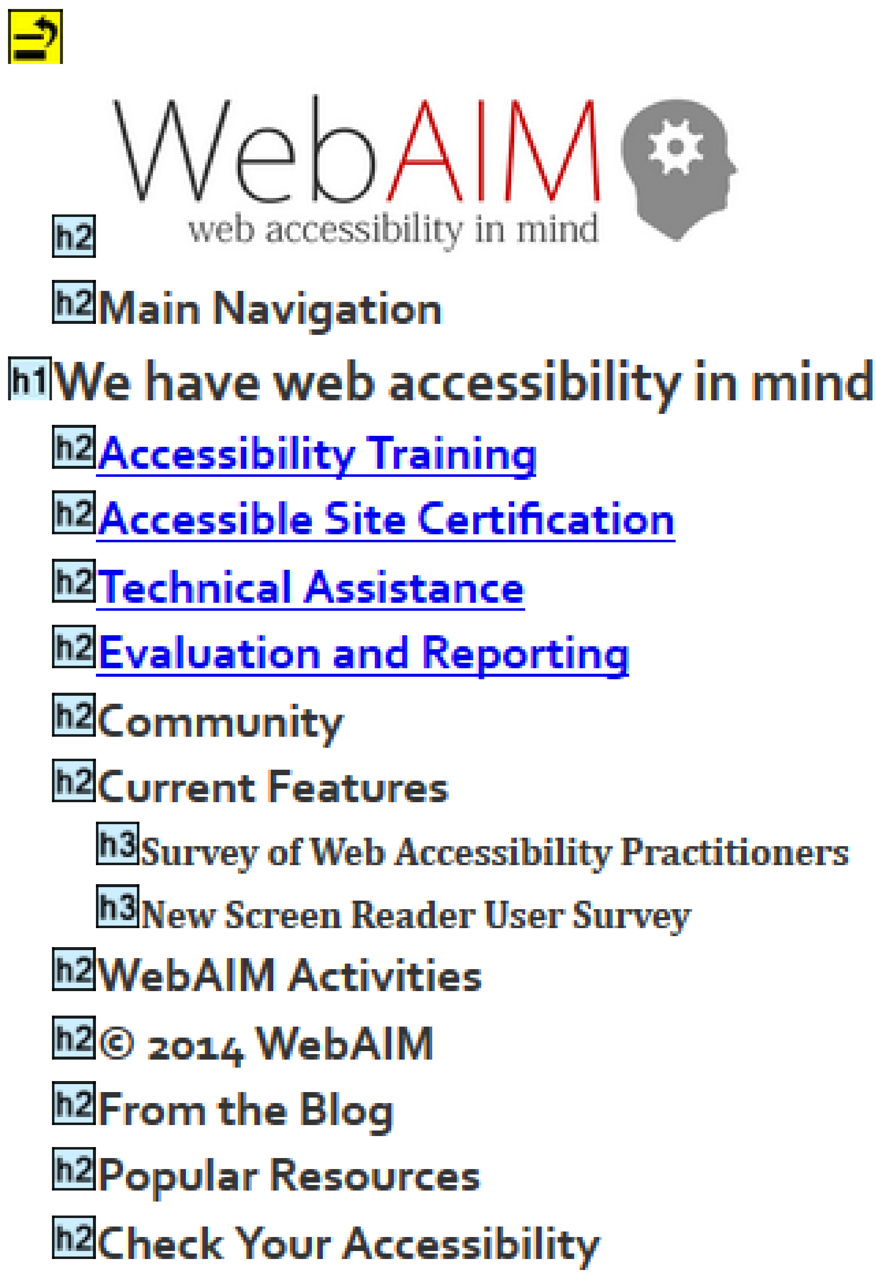


Figure 12. Outline view of WebAIM home page

A review of both the rendered WebAIM home page (fig. 11) and an outline view of the full page (fig. 12) shows how headings are used to segment and provide semantic organization to the entire page.

When we mark-up content on a page semantically, we are not only marking it up for appearance but for its meaning. The semantic tools provided by html 4 (and earlier versions) were somewhat limited. As mentioned above, we could use a heading to indicate the start of a content area, but the non-semantic <div> was required to identify and structure sections of the page with start and end points.

HTML 5 address the limitations of html 4 by providing additional semantic elements such as navigation, banner, footer and search. However, before exploring the additional semantic capability provided by HTML 5 (3rd article in this module), we will take a look at a WebAIM article that explores the use of headings and other semantic elements in greater detail.