A Web Accessibility Primer

Usability for Everyone

Office of Web Communications
Cornell University
2007

Table of Contents

I. Introduction to Web Accessibility 5

People with Disabilities Using the Web 6
Resources 6

Cornell's Approach to Web Accessibility 8
Accessible Design is Good Design 8
Accessibility Standards 9
Checking in at Cornell 9
Policy and Section 508 11

Basics of Accessible Design 12
Think Access 12
Accessible Design Principles 12
Evaluating for Access 14

II. Cross-cutting Accessibility Strategies 16
Labeling and Structure 16
Color Use 16
Acronyms 17
Hyperlinking 17
Avoid Flickering Images 17
Tables 18
Image Use and Alternative text (alt-text) 19
Forms 20
Audio and Video Media 21
Text Presentation and Formatting 22
Writing Style 22
Timed Responses 22

III. Creating Downloadable Files 23

Creating Accessible Word Documents 24
Use Real Headings 24
Make Real Lists 25
Improve Tables 25
Describe Images 26
Resources on Word 27

Making Adobe PDF Content Accessible 28
Improving PDF accessibility 28
Resources on PDFs 30

Creating Accessible Excel Spreadsheet Content 31
Resources on Excel 31

Creating Accessible PowerPoint Content **32**
Guidelines for Increasing PowerPoint File Accessibility 33
Resources on PowerPoint 34

IV. Making "Regular" Web Pages Accessible – HTML 35

HTML Authoring Tools **36**
Using Dreamweaver to Create Accessible HTML 36
Microsoft's Editors – FrontPage and Expression Web Designer 37

HTML Accessibility **38**
Content and Structure 38
Forms 40
Images in HTML Pages 41
Keyboard Accessibility 43
Frames 44
Tables 44
Resources on HTML Authoring 46

V. Beyond HTML: CSS, JavaScript, Plug-ins 47

Cascading Style Sheets (CSS) **47**
Resources on CSS 48

JavaScripts **49**
JavaScript Accessibility Issues 49
Resources on JavaScript 49

Plug-ins **50**
Making plug-ins more accessible 50
Resources on Plug-ins 50

VI. CommonSpot and BlackBoard 51

CommonSpot **51**

BlackBoard **52**

VII. Choosing and Converting File Types 53

Appendix A – 508 Guidelines from A to P 54
The Section 508 Guidelines for the Web 54
Resources on Section 508 55

Appendix B – Related Accessibility Resources 56

Cornell Resources **56**

General Resources **56**

Checklists 56
Link Collections 57
Online Courses and Tutorials 57

Resources by Topic 58

Introductions to Accessibility 58
Specific Disabilities and Web Access 58
Accessibility Standards 58
Evaluation Resources 59
Images and Alt-text 59
Transcripts and Captions for Audio/Visual Media 60
Writing 60
Microsoft Word 60
Adobe PDFs 60
Microsoft Excel 61
Microsoft PowerPoint 61
HTML 61
Cascading Style Sheets (CSS) 62
JavaScript 63
Plug-ins 63
Converting Files 63

Part I - Introduction

When did you last access the internet? What did you use it for? Imagine, now, going to those same web sites and trying to use them if you were blind, deaf, had cognitive trouble dealing with large amounts of content, or couldn't use your arms. At Cornell, we have about 800 students, plus many staff and faculty, with disabilities. Also, of course, many of our website visitors from outside the University, whether prospective faculty or students, or alumni, or the general public, have disabilities that affect their web use.

In many ways, the internet is one of the best things that ever happened to people with disabilities. For example, blind people can now access newspapers with screen readers that read the text aloud. They don't have to wait for expensive audio tapes or costly, and bulky, Braille printouts.

On the other hand, when not designed accessibly, the web creates all kinds of barriers to people with disabilities. What if a newspaper webpage is not screen-reader accessible? What if a video clip has no captions for deaf people? What if a page is only useable with a mouse, and you don't have use of your arms or hands?

With support from this workbook, this workshop should help you:

- Better understand the barriers and frustrations people with disabilities face with inaccessible websites.
- Know what web accessibility means.
- Make the websites and site content you are responsible for more accessible to both people with disabilities, and users generally.

This workshop does not provide full technical training in designing accessible web content. Instead, it introduces you to key concepts and strategies. In some cases, this workbook will tell you all you need to know to make your sites accessible. In other cases, when you'll need particular technical skills or procedures, it refers you to outside resources that you can use on your own as you need them.

While this workbook is designed to be used in a face-to-face workshop, you could also use it on your own as a primer on creating and maintaining accessible websites.

I. Introduction to Web Accessibility

This section introduces the kinds of disabilities that affect web use, Cornell's approach to making our websites accessible, and what web accessibility means.

It should help you:

- Describe the kinds of barriers people with varied disabilities face when they encounter inaccessible websites.
- Feel empathy for people who encounter inaccessible websites.
- Motivate yourself to try to make sites that you are responsible for accessible.
- Understand Cornell's basic approach to web accessibility.
- List some key components of making a site accessible.
- Understand web accessibility as a process, rather than an end.
- Identify some ways your own web sites might be inaccessible.

People with Disabilities Using the Web

Up to 20% of Americans have some kind of disability. The major kinds that affect web use include:

-  **Visual** – blindness, low vision, color-blindness
-  **Hearing** – deafness
-  **Motor** – inability to use a mouse, slow response time, limited fine motor control
-  **Cognitive** – includes learning disabilities, inability to focus on lots of data

Adaptations, often quite simple ones, can make our Cornell websites accessible to most people with most kinds of disabilities. Also, usually, these adaptations benefit nearly everyone, not just people with disabilities. Almost everyone benefits from helpful illustrations, properly-organized content and clear navigation. Similarly, while deaf users need captions, they are also helpful for people who need to view a video without audio, for example in the office or at home with children playing loudly.

These two exercises may help those of you without disabilites glimpse what using the web can be like for people who do have visual or motor impairments.

Visual: Screen readers, such as JAWS, read web page content aloud for people who have low or no vision. Try this simulation, and the associated exercise, for an insight into what using a screen reader is like: www.webaim.org/simulations/screenreader-sim.htm (Shockwave required).

Motor: People who don't have use of their arms or hands sometimes navigate the web via the keyboard, hitting keys with a stick in their mouths. But this requires site design that allows for exclusive keyboard navigation. Go to one of your favorite websites. Try getting around the pages using just the keyboard, without using a mouse. Can you do it?

Resources

The succinct web-based presentation, "A Quick and Dirty Introduction to Accessibility," at www.maxdesign.com.au/presentation/accessibility provides an excellent introduction to web accessibility, along with links to explore assistive technologies available to people with different kinds of disabilities.

WebAIM's introduction to web accessibility at www.webaim.org/intro is also a good starting point, with further links to experiences and needs specific to particular kinds of disabilities. Their series of articles on the user's perspective at www.webaim.org/articles provide great insight into the way people with various kinds of disabilities access the web, or can be prevented from accessing by bad web design. The articles often include short videos and exercises as well.

Disability-specific Resources

WebAIM has extensive resources, including simulations, about making the web accessible to people with visual disabilities. For example, visit www.webaim.org/techniques/screenreader/ for more details on screen readers in particular, with links to further resources available at the end.

WebAIM's section on deafness, at www.webaim.org/articles/auditory, provides some insight into hearing loss and how this impacts web use.

The short article "Making the Web Accessible for the Deaf, Hearing and Mobility Impaired," at www.samizdat.com/pac2.htm, provides some quick tips on accessible design for people with auditory or motor disabilities. The short article "Web Design for the Mobility-Impaired," at www.ptvguy.com/accessibility-web-design-for-the-mobility-impaired, provides some additional tips.

The authors of "An Accessibility Frontier: Cognitive disabilities and learning difficulties" provide guidance in these areas at www.usability.com.au/resources/cognitive.cfm.

Cornell's Approach to Web Accessibility

"Where any person can find instruction in any study" is Cornell's founding motto. Designing accessible web pages is a part of this. So Cornell is committing to making our websites accessible to everyone, including people with the more common sensory, motor and cognitive disabilities. The reasons for this policy are:

- **Moral** – making our sites more readily available to more people is the right thing to do.
- **Practical** – in most ways, a site that is more accessible to people with disabilities is more usable and accessible to everyone. For example, cleaner designs and navigation and alternative media options benefit all users. And, of course, many in the Cornell community, and many we might hope to entice to join Cornell, have disabilities, and we need to make our websites accessible to all of us.
- **Legal** – the Federal government already requires that its own sites are accessible to people with disabilities, and it probably will not be long before state-funded organizations will be required to meet some accessibility standards.

Accessible Design is Good Design

Accessible design is good design, in every sense of the word "good." It's ethically and morally good to make the Cornell web space available to all people, including those with disabilities. Also, most strategies for making sites accessible to people with disabilities lead to sites that are more usable for everyone. It's like the curb cuts in sidewalks. While these ramps are designed to facilitate crossings for people in wheelchairs, nearly everyone has taken advantage of them for ease of access with other kinds of wheels, whether bikes, strollers, rolling suitcases, or trolleys.

Most accessibility strategies facilitate organized and structured content, clean layouts and easy-to-read text. They have other advantages as well. For example, one designer notes that an accessible web design also optimizes the chances that search engines will find your site (see www.alistapart.com/articles/accessibilityseo). Also, accessible designs tend to work better with small devices, such as hand-held computers and mobile phones that people are increasingly using to access the web.

Accessibility Standards

The two main sets of accessibility standards for the web are the Federal government's **Section 508** and the World Wide Web Consortium's (**W3C**) **Web Accessibility Initiative** (WAI).

Section 508 of the Rehabilitation Act, last amended in 1998, requires that the federal government make all of its information technology accessible to people with disabilities. Only part of it deals with the web. Cornell is adopting these Section 508 web standards for making our web sites accessible. The Cornell web accessibility policy, discussed below, describes these standards. Also, for more information on 508 see Appendix A, on page 54.

WAI provides sets of international standards developed by the international, non-profit consortium that develops web protocols and standards, the W3C. These are similar to those that Section 508 lays out. See www.w3.org/WAI for more information.

Checking in at Cornell

Below, a few people at Cornell share their stories, experiences, and views on making the web content they are responsible for accessible for everyone.

Angela Mennitto

Student and Academic Services Web Communications Coordinator

Back when we were making our web sites logo compliant, I chose to use the Cornell templates [see page 40]. I did that both because I'm not a designer and because the folks who designed the templates indicated that they already met most accessibility standards.

Web accessibility and usability are big concerns of mine. I knew the accessibility policy was coming, so I wanted to get in front of that train. With or without a policy, I think it's the right thing to be doing.

I didn't change the templates radically, because of course you can make them inaccessible too. Recently IWS ran one of our sites against accessibility evaluation software; there were very few errors that I have to correct on approximately 50 pages (or on the 4000 page archive that was included in the evaluation). My experience so far has been pretty positive and I think it's because I used the templates as my starting point.

With the push to be sure sites are 508 compliant I would like to add my voice to those who are asking that folks make sure their sites are usable as well as accessible. No piece of software can ensure that a site is clearly organized, easy to navigate and has the text parsed out in comprehensible chunks.

Debra Howell

Associate Director Facilities Services Information Technology

The main goals of the collection of web sites in Facilities Services are to provide access to information and transparency in our processes. We strive to continually enhance our web sites and the user experience. For us, achieving Section 508 Compliance is a part of meeting these objectives.

We treat the pursuit of accessibility as a journey, not a destination. We have found that the process of day-to-day maintenance of our web sites gets us closer to full compliance. Since we are continually adding to our sites, we have to repeat the accessibility evaluation process often and

make necessary adjustments to our code.

Our biggest challenge was finding menu software that was compliant. Most of our other issues were small tweaks like forgotten “alt” tags and the “no script” tag. Overall, the process of working towards compliance has not been too onerous for Facilities. Most importantly, we didn’t have to compromise the feature richness of our sites in order to become compliant.

Mitch Wiedemann
Webmaster, School of Continuing Education & Summer Sessions

Equal access to information in this era known as the "Information Age" is something I feel strongly about. In terms of providing access to people who have disabilities, as well as people who are otherwise disenfranchised, be it by financial, cultural, or other means. Information is power, and we live in a time when near-universal access to all human knowledge at little or no cost is within the realm of possibility for the first time in history.

For this universal access to information to become a reality, information must be published in open, well-documented, and accessible standardized formats. Formats accessible to anyone using any reasonable platform of technology that can receive the information--whether that's a standard desktop computer, an assistive device, a small mobile device, or another computer program. Our Web sites which enable public interaction via online forms should likewise be completely accessible.

The real costs of inaccessibility were seen in news stories following the disaster of hurricane Katrina. Many people who needed to use government Web sites to request aid, were unable to do so because they didn't have access to a computer with the specific Web browser software that the Web site required.

For me, the specific issue of Web accessibility is also a personal one. I have family members and friends who have visual and hearing disabilities. So my decision to design and build my Web code in an accessible manner was an easy one. Cornell's roots as being the "any person … any study" institution, as well as the Federal Section 508 law and the Cornell draft policy on accessibility, provide added incentive to build my Web space in a standard, accessible manner.

I've always tried to write code that fit with the W3C HTML standards, which is the first step in making a Web site accessible. Evaluating and refactoring my code specifically for Section 508 accessibility began in earnest when I found a free online evaluation tool. The most common errors found on my site were missing "alt" tags for images, missing form field labels, and misuse of JavaScript for form submission, rather than using the standard "submit" button. The practice of using server "includes" of common code snippets rather than having the code embedded separately in every Web page made the accessibility enhancements a great deal faster and easier, as did using powerful "find and replace" tools. I urge others who are not using server-side includes of one kind or another to do this, it saves time and energy in every aspect of Web development. The process of making sure my sites are accessible is an ongoing project, so I've made the use of the online accessibility tool a part of my regular work flow.

Finally, for me, building accessible Web sites comes down to a matter of craftsmanship. If the HTML markup on my Web site isn't perfect, 99% of the site visitors may never know the difference, but *I* will. My Web site is the public face of my School. It's my job to be sure that it's a site that supports Cornell's mission of "Any person … any study."

Policy and Section 508

Cornell's draft web accessibility policy is at www.cit.cornell.edu/policy/drafts/WebAccess.html. This policy is founded on the moral, practical and legal grounds for making our sites more accessible discussed above, and is based on the Federal Section 508 standards for accessible web design.

Although this policy is still in draft form and the timeline for implementation is generous, start thinking about these approaches now. Designing with accessibility in mind going forward is easier than retrofitting existing sites and content. And accessible design is good design, in every sense of the word "good." So start now.

Message from the Director of Information Technology Policy

Tracy Mitrano

In 1973 Congress passed the Rehabilitation Act to prohibit discrimination against people with disabilities and to provide funding assistance for independent living, vocational and educational support for people with disabilities. In 1998 Congress amended that act to include a new section, "508" which required federal agencies and departments to incorporate some elemental design standards into their web pages and materials with the purpose of making them more accessible to people with the most common visual, hearing and motor impairments. These standards went into effect in 2001 and since have been adopted in whole and in part by many higher educational institutions, particularly state institutions, as well as the entire state government and educational system of California.

As a private institution[1] Cornell University is under no legal requirement to adopt these standards at this time. The Executive Policy Review Group, responding to an initiative from the Offices of the Vice Presidents of University Communications and Information Technology, have approved an impact statement that would incorporate these standards into the university official web space (with exceptions) through university policy. This policy serves the strategic purpose of attracting and retaining talented students and employees, the idealistic purpose of living Cornell's motto of "any person … any study," the realistic goal of working toward universal design of its web space and the legal goal of compliance in advance of a compliance requirement for colleges and universities that receive federal funds.

To date this policy has been four years in the drafting and vetting, and currently is not expected to go into effect until September of 2008. It requires compliance with "508" design standards for all web pages that conduct university business, with notable exceptions, and including an undue burden provision. Moreover, it contains a timeline for implementation that depending on the nature of the site or application stretches out another five years. Thus, this policy is mindful of the administrative burden it places on individuals and units throughout the university. Cornell Information Technologies and Office of Web Communications offer this primer to mitigate that burden, assist individuals and units in the instruction of how to achieve compliance with the policy, and finally to realize its laudable benefits of institutional integrity, reputation and on-going strategic positioning of Cornell University.

[1] By law Cornell is a private institution, including its "contract" state colleges. As a private institution it is not subject to the reach of section 508 as it applies to federal agencies. Receipt of federal funds alone, in the form of either grants or student aid, does not change the corporate and legal status of the university nor make it subject to the law.

Basics of Accessible Design

While words like "checklists", "policy" and "compliance" are often used around meeting web accessibility standards, in reality accessible design is an art, not a science. It is also an ongoing process, not things to simply check off a list.

Think Access

Making your web sites accessible is about trying to look at your design, content and navigation from the perspectives of disabled users. It is not about mindlessly following a set of rules.

How would you describe Munch's *The Scream* to a person who could not see it?

For example, a common "rule" of accessible design is to provide alt-text descriptions ("alternative text" that appears instead of an image for people who choose to view pages without images and also for screen readers, which read the text of the page aloud) of every image. However, if the main content of the page is about the image, then this would be redundant. Also, how or even if you describe an image also requires some thought. You might not bother describing a decorative image, like a flower, that is unrelated and unimportant to page content. On the other hand, some images – such as artwork, charts, and graphs – require you to put more thought into how to convey their content. For example, how would you describe the Edvard Munch's famous painting, *The Scream*, shown here, in less than a thousand words?

Accessible Design Principles[2]

Here are some key principles of accessible design. You can implement most of these very easily, and without impacting the overall "look and feel" of your web pages. Most will improve your website's usability for *all* users.

Each is covered in more detail in the next parts of this workshop. To explore how these and other parts of this workbook match up with the Section 508 policies that Cornell intends to adopt, please see the Section 508 Appendix on page 54.

a) Provide appropriate alternative text

Alternative text provides a textual alternative to non-text content in web pages; for example, it describes the content of a photo. It is especially helpful for people who are blind and rely on a screen reader (which reads pages aloud) to have the content of the website read to them.

[2] Much of this workbook draws from the WebAIM (Web Accessibility in Mind) project, a Utah State University initiative, online at www.webaim.org. This section in particular borrows nearly entirely from www.webaim.org/intro.

b) Provide headings for data tables

Tables should have appropriate table headers (in HTML, this is via the <th> element). Data cells should be associated with their appropriate headers, making it easier for screen reader users to navigate and understand the data table.

c) Ensure users can complete and submit all forms

Ensure that every form element (text field, checkbox, dropdown list, etc.) has a label and make sure that label is associated with the correct form element using the <label> tag. Also make sure the user can submit the form and recover from any errors, such as the failure to fill in all required fields.

d) Ensure links make sense out of context

Every link should make sense if the link text is read by itself. Screen reader users may choose to read only the links on a web page. Certain phrases like “click here” and “more” must be avoided.

e) Caption video, provide transcripts for audio

Videos and live audio need synchronized captions. All audio – whether as part of video, live or archived – should also have a text transcript.

f) Make file downloads (e.g., PDFs) accessible

Ensure accessibility of non-HTML content, including PDF files, Microsoft Word documents, Excel spreadsheets, and PowerPoint presentations. Usually you can do this in the files themselves, though sometimes it might be easier to use HTML instead of these file types.

g) Allow users to skip repetitive elements on the page

You should provide a method that allows users to skip navigation or other elements that repeat on every page. This is usually accomplished by providing a “Skip Navigation” or “Skip to Main Content” link at the top of the page which jumps to the main content of the page.

h) Do not rely on color alone to convey meaning

The use of color can enhance comprehension, but do not use color alone to convey information. That information may not be available to a person who is colorblind and will be unavailable to screen reader users.

i) Make sure content is structured, clearly written and easy to read

There are many ways to make your content easier to understand. Write clearly, use clear fonts, and use headings and lists appropriately.

j) Make JavaScript accessible

Ensure that JavaScript event handlers are device independent (e.g., they do not require the use of a mouse) and make sure that your page does not rely on JavaScript to function.

k) Design to standards

HTML compliant and accessible pages are more robust and provide better search engine optimization. Cascading Style Sheets (CSS) allow you to separate content from presentation. This provides more flexibility and accessibility of your content.

This list does not present all accessibility issues, but by addressing these basic principles, you will ensure greater accessibility of your web content to everyone.

Evaluating for Access

A variety of free and commercial tools lets you check one or more aspects of how accessible your web sites are to people with various kinds of disabilities. Some of the free ones for checking particular aspects of your site are discussed here in the relevant sections.

As Cornell rolls out the accessibility policy, the University will license a full service application for checking all of our web sites. Support for using that tool will then be available.

However, like spelling or grammar checking tools, these tools cannot replace your own judgment and experience. Also, of the 16 standards in Section 508 (see page 54), only seven can be evaluated automatically, and these only partially. So you cannot rely on a tool to determine whether a page is accessible or not. The key to creating websites accessible to everyone is to understand the web accessibility standards and how people with disabilities access the web.

Evaluation Resources

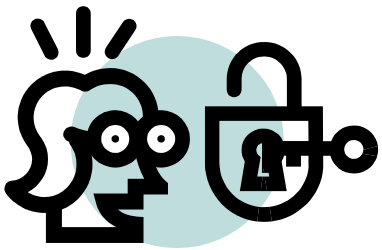
WebAIM, a non-profit organization within the Center for Persons with Disabilities at Utah State University, provides a series of articles on evaluating accessibility (see the Evaluation section of www.webaim.org/articles), which includes a useful review of free online tools (www.webaim.org/articles/freetools). They also host their own free evaluation tool, WAVE, at www.wave.webaim.org.

Part II – How to Make Your Sites Accessible

This section introduces strategies and skills to make the websites that you work on more accessible, whether you are simply creating text documents for uploading or designing entire websites.

It should help you:

- Describe and understand some cross-cutting strategies for making electronic media accessible, no matter what kind of program or application you are using to create it.
- Create or update the following kinds of files to make them accessible:
  - Microsoft Word
  - Adobe PDF
  - Microsoft Excel
  - Microsoft PowerPoint
- Write or edit HTML pages to make them accessible.
- Make CommonSpot and Blackboard sites as accessible as possible.
- Begin to understand accessibility strategies needed when you are using CSS, JavaScript, or plug-ins.
- Find the additional training and reference resources you need to help you make your websites and all of their content accessible to everyone, including those with disabilities.



II. Cross-cutting Accessibility Strategies

These strategies apply to just about anything you put online, from Adobe PDF files to HTML to Word documents. This workbook refers to these approaches in sections about strategies appropriate for particular applications, but does not usually repeat them. So you should return to this section often to keep these basic, but crucial, approaches fresh in your mind.

Labeling and Structure

Breaking down your media into clearly structured and labeled sub-sections makes it more accessible to *all* users, and especially to those with cognitive difficulties.

For people who use screen readers to read the textual content aloud to them, providing machine-readable structure is essential. This means different things in different kinds of media, but mainly it means using "official" heading, bullet and other styles to indicate your file structure. E.g., don't make a "heading" by bolding and enlarging the font. Don't make lists by typing in asterisks. Instead, use the "official" styling and formatting options of the file type you are creating. For example, as discussed in later sections in more detail, use Styles in Word and header and bullet list codes in HTML.

Color Use

Don't use color alone to provide information. For example, you can make hyperlinks a different color, but need to also differentiate them with underlining. This makes them accessible to people with low vision or colorblindness. Similarly, don't use color alone to mark sections of a pie chart, but label them as well.

Blue, yellow, white and black are your best bets for providing color contrast, and not only for people with vision impairments. People who are color blind often cannot differentiate between red and green, so steer clear of these for conveying information. And, again, provide other visual or textual cues when using color to convey information.

> Choose a web page that uses color that you are at least partly responsible for and check for colorblindness accessibility at Vischeck, www.vischeck.com.
>
> While you are there, also explore some their examples of how images look to people with various kinds of colorblindness.

Acronyms

As is a good principle in all circumstances, explain what any acronym stands for the first time you use it. Keep a few things in mind here:

- Screen readers will read out an acronym as a word if it has enough vowels in it. For example, the distress call "SOS" will be read out as something like "sauce". So be sure surrounding text gives enough information to make this still understandable.
- People with reading disabilities or for whom English is not a first language may need even the most common acronyms, such as CEO, explained.

A good article on how to handle acronyms, particularly in HTML contexts, is at www.alistapart.com/articles/hattrick.

Hyperlinking

Three keys to accessible hyperlinking are:

- Don't use color alone to indicate links. Provide other indications, such as underlining.
- Make links and hot spots large enough for people with limited motor control to click.
- In web pages, links should make sense out of context. Blind users can ask their screen readers to skip page content and only read out the links. This renders links such as "click here" totally useless. The link would need to include more detail, such as "click here for more information about our department."[3]

Find a partner, and visit a Cornell-affiliated website that you have worked on (or will work on). The partner should close her/his eyes. Now, read aloud each of the hyperlinks on your webpage (this is what a screen reader set to read just links would do). How much sense do they make to your partner out of the visual context of the page?

Avoid Flickering Images

Rapid strobing, flickering, or flashing effects can induce seizures in some people. Don't use them. They are annoying and distracting to nearly anyone, and are downright dangerous for some. The larger the flashing image, the more likely it is to cause a problem.

Resources on flickering images

For more information and links to examples (which you should not click on if you are susceptible to seizures) visit www.webaim.org/articles/seizure.

[3] Note that if you are concerned about the aesthetics of such long link names, you don't need to visually indicate the link for the entire text. You can simply underline the "click here" for sighted users, but insert the hyperlink for the fuller text so a screen reader will read the entire text.

Tables

Your best bet for checking if your table is accessible to the screen readers that people with no or low vision use to read text aloud is to read it like a screen reader would, as shown in Table 1 below.

Screen	readers	read
Information	across	tables
in a	linear	way
Thereby	making	It
Difficult	to	understand
Information	contained	in tables.

Table 1: How a screen reader reads data in a table[4]

Overarching guidelines for creating more accessible tables include:

- Define column and row size using percentages, not fixed measurements. This allows enlargement without distortion.
- Use the simplest layout possible. Do not use more than one heading per row or column (i.e., no subheadings). Do not span rows or columns. See Table 1 below for an example of spanned rows, and the more accessible solutions to presenting the same information in Table 3 and Table 4 below.

Dept. Code	Class #	Section	Max Enrollment	Current Enrollment	Rm. #	Days	Start Time	End Time	Inst.
BIO	100	1	15	13	5	Mon,Wed,Fri	10:00	11:00	Magde
	100	2	15	7	5	Tue,Thu	11:00	12:30	Indge
	205	1	15	9	6	Tue,Thu	09:00	10:30	Magde
	315	1	12	3	6	Mon,Wed,Fri	13:00	14:00	Indge
BUS	150	1	15	15	13	Mon,Wed,Fri	09:00	10:00	Roberts
	210	1	10	9	13	Mon,Wed,Fri	08:00	09:00	Rasid

Table 2: Example of table with spanned row headings, BIO and BUS[5]

[4] Thanks to Cornell's Sharon Trerise for this table and for Table 5.
[5] These table examples are from www.webaim.org/techniques/tables/data.php

Dept. Code	Class #	Section
BIO	100	1
BIO	100	2
BIO	205	1
BIO	315	1
BUS	150	1
BUS	210	1

Table 3: Possible solution to Table 1

Dept. Code/ Class #	Section
BIO 100	1
BIO 100	2
BIO 205	1
BIO 315	1
BUS 150	1
BUS 210	1

Table 4: Another possible solution to Table 1

If your table layout must be too complex for a screen reader to access, then provide a text explanation that summarizes the information in the table. In some cases, such as providing a class schedule, you'll also need to provide contact information so that blind users can get the information they need.

If you are creating tables in HTML, please refer to page 44 for an additional introduction to creating accessible online tables.

Resources on tables

You'll find a good introduction, including to HTML tables, at www.webaim.org/techniques/tables.

Image Use and Alternative text (alt-text)

Images can be a great way to increase usability and understanding by most web users, with or without disabilities. They are particularly helpful to many with cognitive difficulties, such as reading disorders. A picture can be worth a thousand words. Unless, of course, you are blind. In which case you will need some words to help you understand the content.

The main way of doing this is to prove alternative text, or alt text, that screen readers will read aloud for users with low or no vision. Most common office applications (e.g., Word), as well as HTML, let you add this descriptive text. More technical details for how to do this in several applications is discussed later. But whatever media format you are creating, some general guidelines on providing images and good alt text include:

- Eliminate or minimize graphical (vs. real) text. Graphical text distorts when enlarged, and is inaccessible to screen readers (see Figure 1, below).
- Communicate the purpose of the graphic accurately and succinctly in the alt text.
- In HTML, provide empty or null alt text for graphics which do not convey content (i.e., use alt=""; see the HTML section on page 41). In other media formats, don't bother describing images that are merely decorative.
- In image maps, provide alt text for both the main image and the hot spots.

- Do not repeat the alt text of an image in the adjacent text. You need one *or* the other.
- Do not put important images in the background.

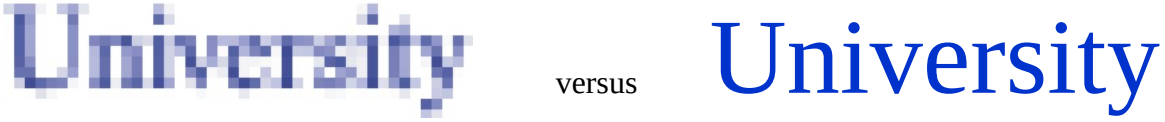


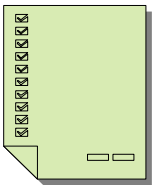
Figure 1: Enlarged graphical text (distorted) vs. enlarged real text (not distorted). (Image from WebAIM)

Most applications let you add alternative text to describe images, whether they are pictures, graphs or other objects. In Word and PowerPoint, right click on the image or object, choose "format picture" or "format object" from the menu, and then enter your alt text in the "Web" tab (see Figure 3 in the section on adding alt text in Word on page 27). For images on HTML pages, see the section about alt text and other options in the HTML chapter (see page 41).

Resources on images and alt text

WebAIM provides a great introduction to image accessibility at www.webaim.org/techniques/images/, which includes useful examples of good alt text for the same images in different contexts. For additional succinct yet thorough guidelines on writing good alt text, turn to the article "Writing good ALT text" at www.gawds.org/show.php?contentid=28.

Forms



Everyone benefits from a well-organized, highly usable form, whether for filling out a survey or registering for a class. Most of the work to create forms accessible to people with disabilities will benefit everyone. Many accessibility strategies are specific to the media type being used, e.g., HTML or PDF, but overall good techniques include:

- Users must be able to complete the form using only the keyboard. (Within web-based forms, this means no JavaScripts that change browser location.)
- Organize forms logically. This includes providing clear instructions, labeling what is required, and lining up each label (e.g., Gender) with the element being asked (e.g., radio buttons for Male and Female). In HTML, it also means being cautious about using table layouts, ensuring screen readers will read things in the right order (covered later).

Experience what a poorly organized form sounds like to someone using a screen reader, and explore the more accessible alternatives, at: www.webaim.org/techniques/forms/screen_reader.php

Audio and Video Media

The two guidelines for audio media are:

- Provide a text transcript for all audio media.
- If the audio is associated with images or video, also provide text captions that are synchronized with the visuals.

Transcripts

Transcripts are text versions of any audio you use on your web site, whether in a web page or embedded in a downloadable file (e.g., a Word Document). They allow anyone that cannot access content from web audio or video to read a text transcript instead, which is useful not just to people with hearing disabilities. For example, transcripts help those who would like to print out the content for later study, or who are working in a setting where audio would be inappropriate. It makes the content searchable. Also, people can often read transcripts faster than they can listen to the audio. In particular, screen reader users often set the readers to read at a rate much faster than most humans speak (and faster than most people without screen reader experience could understand).

Transcripts do not have to be verbatim accounts of the spoken word, and can contain additional descriptions, explanations, or comments that may be beneficial.

For most web video, both captions and a text transcript should be provided. For content that is audio only, a transcript will usually suffice.

Captions

When your audio is associated with any visuals, such as images, slides, or video, you should provide captions as well as a text transcript. Captions should be:

- Synchronized – the text content should appear at approximately the same time that audio would be available.
- Equivalent – content provided in captions should be equivalent to that spoken.
- Accessible – caption content should be readily accessible and available.

Resources on providing transcripts and captions

Most of the above is drawn from WebAIM's article on captions at www.webaim.org/techniques/captions. It provides details on captioning options and technologies that can help you create synchronized, accessible captions.

See also the checklists on video use at www.catea.org/grade/guides/videomust.php.

Text Presentation and Formatting

Some basic guidelines for making your text content more readable, for everyone, include:

- Use footnotes rather than endnotes, to help keep them in context.
- Simple, familiar fonts are best (e.g., choose times new roman over “baskerville old face” or any cursive fonts).
- High contrast between text and background is crucial, and helps all users. For example, black on white and dark blue on yellow provide high contrast. Orange on yellow or green on red are low contrast, and hard for even people with excellent vision to read.
- Create large, empty margins around the text.
- Use blank lines between paragraphs.

Writing Style

No matter who your users are, or what media you’re using, writing clearly and simply will make your content more understandable, accessible and enjoyable for everyone. Guidelines include:

- Use active verbs, avoid passive voice (e.g. “Professor Rajan’s lab team invented the device”, not, “The device was invented…”).
- Avoid the verb “to be” as your main verb (e.g., “She probably will win,” not, “She is probably going to be the winner”).
- Keep sentences short and simple. Avoid double negatives.
- Organize your ideas logically.

For more tips on clear writing, and in particular writing for people with cognitive disabilities and reading disorders, visit www.webaim.org/techniques/writing.

Timed Responses

Occasionally you might require a timed response, such as on an online test or quiz, or when a secure page login might expire. If so, alert the user so that he or she has plenty of time to respond before losing his or her place.

III. Creating Downloadable Files

This section explains core techniques for making the files you create for users to download accessible to more people. It covers the following file types: Word and rich text, PDFs, Excel, and PowerPoint. It is designed to help you:

- Describe key elements required to make each of these file types accessible.
- Easily find references to support you when making these kinds of files accessible.
- Create accessible files of each type, in some cases by following instructions provided here and in others via links to step-by-step guides and tutorials to follow when you are back at your desk.

Creating Accessible Word Documents

You probably use Word, rich text files or another word processing program every working day. Since Word is the most common, that is the focus here, though much of the advice transfers to other programs.

The advice here will not only help you make your documents more accessible to people with disabilities, but make more effective and efficient use of some Word features more generally. Much of the key to improved accessibility is to provide *structure*.

Use Real Headings

When you want to make a heading in one of your documents, like the "Use Real Headings" one directly above, how do you do it? If you simply make the text bold and increase its font size, you are missing out on one of Word's most useful features – heading styles.

For screen readers to be able to differentiate headings, you *must* use these styles. They have many other advantages too. For example, if you use real headings, you can create a table of contents in an instant[6] or change the format of all the headings in the document at once[7].

How to Use Existing Heading Styles in Word

1. Highlight the text you want to make into a heading (see Figure 2).
2. Find the Styles list in the toolbar.
3. Choose one of the Heading styles in that list (Heading 1, Heading 2, Heading 3, etc.)

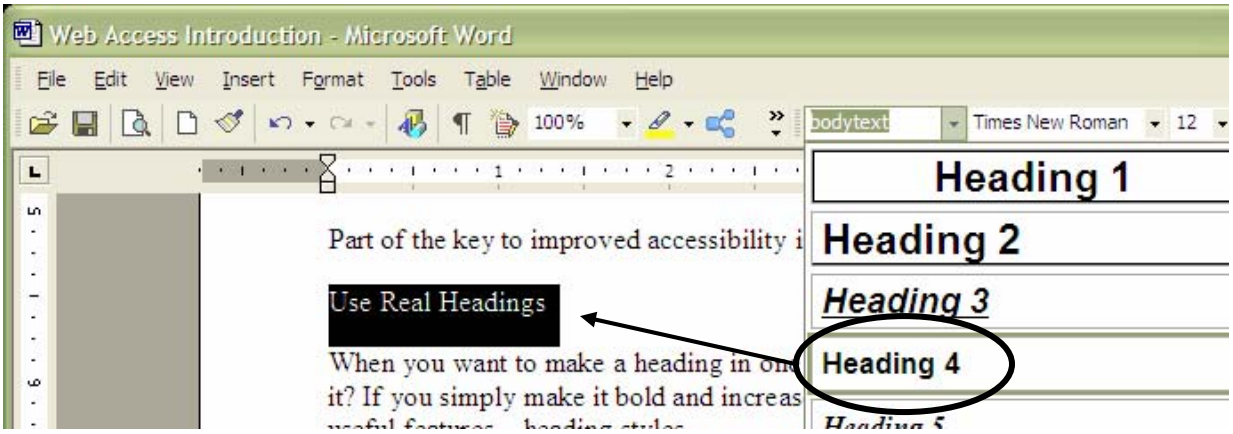


Figure 2: Applying an existing heading style, 'Heading 4," in Word

Use the headings in logical order, i.e., use Heading 1 for your document title (there should only be one Heading 1 in your document), Heading 2 to start each main section, and Headings 3 and

[6] In the Word menu, follow this path: Insert > Reference > Index and Tables > Table of Contents

[7] Format > Styles and Formatting > right click on heading style and choose "Modify" > make format changes > select "add to template." This is also how to modify the heading styles in Word from the existing ones.

lower for subsections under each Heading 2. Don't use heading styles for anything other than indicating your document sections. (If you want to change Word's built-in heading styles, refer to Word help documentation and to the footnote in the introduction to this Word section.)

Breaking down your document into short sections under headings can also make the content more readable for all users, and especially those with cognitive disabilities.

> Try creating your own real headings. Open Word and create a new document. Either type or copy and paste 6 lines of text into the document, with hard returns after each (i.e., hit enter). Gibberish or random words are OK.
>
> Now, make the first line into the built-in Heading 1 style. Make the third line Heading 2 style. And make the fifth line Heading 3 style.

Make Real Lists

As with headers, when you make bulleted or numbered lists, use Word's built-in style features to create them. This makes it easier for you, and possible for screen readers to identify the lists.

Ways of Making Real Lists

Start by highlighting the text you want to make into a bulleted or numbered list. Then either:

- Click the bullet or number icon in your toolbar.  Bullet toolbar icon in Word
- In the Word menu, follow this path: Format > Bullets and Numbering > select the Bulleted, Numbered, or Outlined Numbered tab as appropriate.
- Choose a bulleted or numbered style from Word's built-in styles, in the same way as you select a heading style described in the previous section.

Improve Tables

Even the best structured table in Word can be challenging for a screen reader to make sense of. However, particularly in academics, tables still often provide the clearest way of presenting information. If you do need to use tables, it is better to create them in HTML than in Word. (Refer to the HTML table formatting section on page 44).

When you do create tables in Word, simplify them, define column widths with percentages (not, e.g., inches) and create clear row and column labels.

Simplify

Simplify your tables as much as possible. For example, because screen readers have trouble making sense of them, avoid using more than one heading for rows or columns (i.e., don't use subheadings), and don't use headings that span more than one row or column. Also, complex

layouts can be confusing for all users, and particularly for people with some kinds of cognitive difficulties.

Define Column Widths with Percentages

If you are creating a table to represent data, define column widths with percentages, rather than fixed numbers. This lets a user with low vision increase the table size without losing the proportions.

Put your cursor in the table, and then in the Word menu, follow this path: Table > Table Properties > select the Columns tab > select "Percent" from the "Measure in:" menu.

Create Clear Row and Column Labels

While you cannot make row and column labels accessible for screen readers in Word, by setting them off with bold and larger font formats you can make them more readable for everyone who has sight, including people with low vision. (Note: do not use real Word heading styles for tables, this will confuse the structure of your document).

Describe Images

No matter what medium you are using – e.g., Word, PowerPoint, or HTML – you should provide text descriptions of any image you include if it is providing more than background decoration. This is descriptive text is called **alt text**, short for "alternative text".

In short, imagine what would be most useful to you if you couldn't see the page. When the screen reader comes across this alt-text, it will say "Image" and then read aloud whatever alt text you provide. Shorter is usually better, but if the image is central to the content, e.g., a painting shown on an arts course site, then you should probably lean towards being more descriptive. Refer back to the earlier section on drafting good alt text generally on page 19.

How to Insert Alt-Text

1. Highlight the image.
2. Right click and select "Format Picture."
3. Select the "Web" tab and enter your alternative text (see Figure 3).

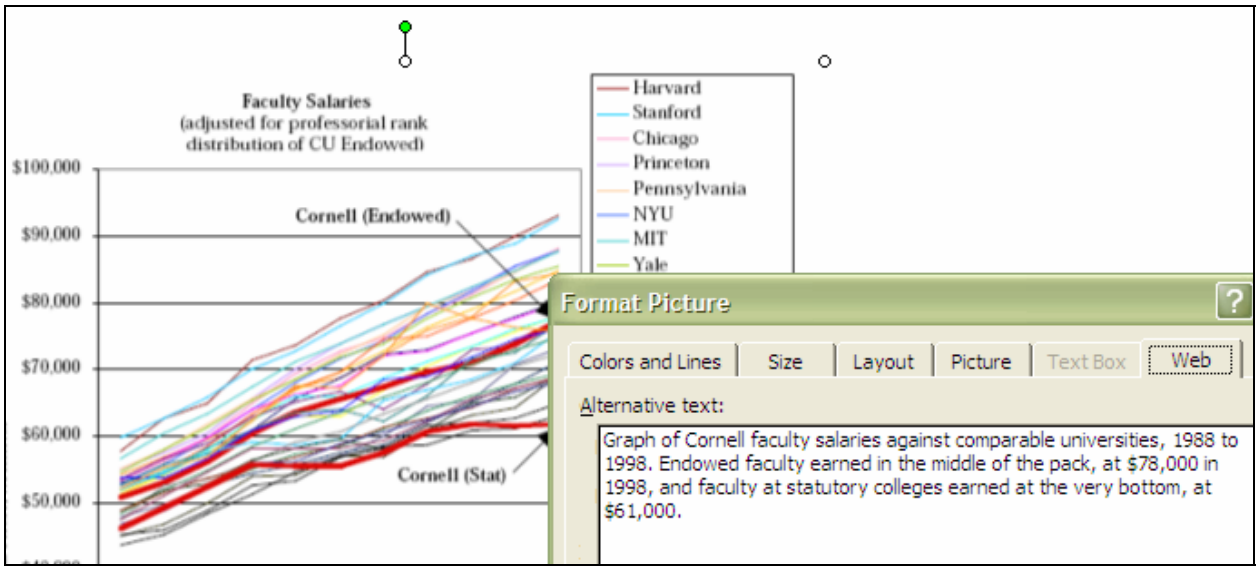


Figure 3: Adding Alt Text to an Image in Word

Resources on Word

As always, WebAIM offers great tips on making Word documents accessible, including some information on converting to HTML, at www.webaim.org/techniques/word.

If you like checklists, you'll like the "musts", "shoulds" and "mays" of accessible Word use at www.catea.org/grade/guides/wordmust.php.

The free online Access E-Learning course, at www.accesselearning.net, has an entire tutorial with detailed instructions and practice labs on this topic.

Making Adobe PDF Content Accessible

Adobe PDF files are simply conversions of other file formats, such as Word or PowerPoint. The free Adobe Reader program opens PDF files. However, to create accessible PDF files, you will need to purchase Adobe Acrobat Professional.

PDFs can be challenging to make accessible to people with low or no vision for several reasons. These include that only two, expensive brands of screen readers can read PDFs[8]. Also, authoring accessible PDF files takes some technical skill, and converting existing inaccessible files requires a lot of work.

You have two options for making PDF content accessible:

1. Make an HTML version available.
2. Make the PDF file itself as accessible as possible.

The table below summarizes factors that might influence which you choose.

Factors making PDF option easier	Factors making PDF option harder **(consider HTML)**
You have few or one author(s)	**You have multiple authors**
You have advanced technical skills	**You have limited technical skills**
You are creating a new file	**You are working with an existing PDF**
You have the original source file	**You only have the PDF file**
The layout is simple	**The layout is complex**

Table 5: Summary of factors influencing the ease of making an accessible PDF file vs. providing an HTML version

Improving PDF accessibility

Accessible PDF files contain real (not scanned) text, are tagged (see below), and have alt-text for all image and graphics. In general, creating an accessible PDF file means:

- Following all of the guidelines for making the source file (e.g., Word document, PowerPoint show) accessible, as described in other sections. For example, provide alt-text for all images and use real heading styles. Starting with a correctly formatted Word document simplifies making its PDF version accessible.
- Following all of the general strategies for accessibility (see page 16) , for example, about use of color and contrast, hyperlink names that makes sense out of context, how

[8] Adobe does offer a "Read-out-loud" feature in Adobe Reader, which, if you follow guidelines for making an accessible PDF file, can make the content available to users with visual impairments.

you structure tables, and making any “hot spots” big enough for those with limited motor control to use.

- Keeping the source document layout simple. Avoid using multiple column layouts, sidebars, and breaking articles between pages as is common in newspaper and newsletter layouts.
- In Word, when converting your document to PDF, under “Settings”, select “Enable accessibility and reflow with Tagged PDF” (see Figure 4; you’ll only have this option if Acrobat is installed on your computer).
- Tagging the content of your PDF so that a screen reader can make sense of it. This is the most complex, though hardly insurmountable part. See below.
- Evaluating the content of your PDF in Adobe Acrobat (in the main menu, follow Advanced > Accessibility > Full Check). Fix any problems.

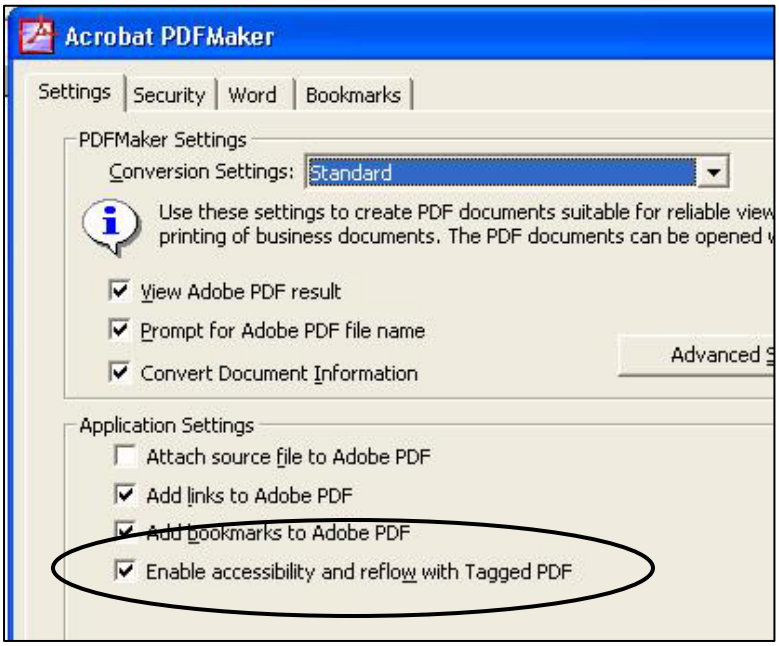


Figure 4: In Word, enabling accessibility when converting to a PDF

Also, to be accessible, the PDF file must be real text (you can copy and paste it), not a scanned image.

Tagging PDFs

PDF tags are structural markers that identify the role of each piece of text, e.g., heading, list, or paragraphs. They perform a function similar to styles in Word or headings and other structural tags in HTML. Tagging your PDF text is essential to make it accessible to screen reader users.

If you are converting a well-structured Word document, using styles appropriately as discussed on page 24, then your resulting PDF should be well tagged. If you are working with an untagged or poorly tagged PDF, then you have more work in front of you. Start with the WebAIM reference on tagging PDFs and then turn to the full instructions that Adobe provides for detailed instructions and guidance (see resources, below). Your tagging work will be easier if you have Acrobat Professional 7.0, rather than earlier versions. Tagging requires version 5.0 or above and, if converting from Word, Word 2000 or above.

Adobe also has a free Make Accessible Plug-in to help create tagged Adobe PDF files from untagged files (see resources below). This also requires Acrobat 5.0 or higher. Once you have installed the plug in, open the PDF file you want to tag and got to Document > Make Accessible. However, the effectiveness of this tagging has received mixed reviews from web developers.

Resources on PDFs

If you decide to post PDF files online, a good introductory how-to guide for Acrobat 6.0 is California Community Colleges High Tech Center Training Unit's "Creating Accessible PDFs," at www.htctu.fhda.edu/trainings/manuals/web/Creating_Accessible_PDFs.pdf. Checklists of musts, shoulds and mays start at www.catea.org/grade/guides/acrobatmust.php, with a full how-to tutorial on PDF as a module within www.accesselearning.net.

The University of Wisconsin offers advice specifically for faculty and for librarians, in addition to step-wise instructions for evaluating the accessibility of a PDF and for producing PDFs from Word documents or scanned documents (www.cew.wisc.edu/accessibility/tutorials/PDF-L2.htm).

At www.webaim.org/techniques/acrobat/understandingtags.php, WebAIM explains how to add tags to new and existing PDF documents. For complete instructions (100+ pages), visit http://www.adobe.com/accessibility/ and download Adobe's own guide to "Creating Accessible PDF Documents with Adobe Acrobat" for Adobe 7.0.

Adobe's free Make Accessible Plug-in for tagging PDFs is online at www.adobe.com/support/downloads/detail.jsp?hexID=88de. However, the effectiveness of this tagging has received mixed reviews from web developers interested in accessibility.

Adobe also offers a free tool for converting PDFs to HTML at www.adobe.com/products/acrobat/access_onlinetools.html.

Creating Accessible Excel Spreadsheet Content

Your best bet for making Excel content accessible is to convert it to more accessible formats, such as HTML, Rich Text, or even PDF. (See the Conversion section on page 53).

Within Excel, in addition to general good accessibility practices, best practices include:

- **Make tables simple**. A screen reader can read a table of text or numbers with one header row and just a few columns. If it gets more complicated that that, you should also provide a text explanation of the table.
- **Explain formulas**. You can either save the formula as an image and provide alt-text (see page 19), or provide a text description of the formula. Neither is ideal, but these are the only ways to provide access to formulas to people with low or no vision.
- **Describe charts.** Charts made using Excel's Wizard don't offer an alt-text option, so you'll need to provide a text description within the spreadsheet.

Resources on Excel

You can take a full online course module on Excel at www.accesselearning.net or review checklists for accessible Excel spreadsheets at www.catea.org/grade/guides/excelmust.php.

Creating Accessible PowerPoint Content

While people with most kinds of disabilities can access most PowerPoints (.ppt files) quite well if you follow the cross-cutting guidelines for accessibility, PowerPoints are less available to those who need screen readers (i.e., with low or no vision).

There are a number of options for making PowerPoint content accessible, each with advantages and disadvantages. Options include:

1. Post the original PowerPoint (.ppt) file. This works if you know your audience has access to the PowerPoint program on their computers or in computer labs. It is best to combine this with an HTML version of at least the text content (see 2 below), if not a full web page conversion (see 3).
2. Post the text content of the file by either:
    a. Saving it as rich text format (File > Save as > select "Outline/RTF" from "save as type" menu) and then posting this as HTML.
    b. Cutting and pasting the outline from the Outline tab to the left of your open presentation and then posting this as HTML. [If this pane isn't showing, go to View > Normal (Restore Panes).] Note that either of these will probably require you to do some reformatting of the outline. Anything you did not put into slide layout boxes (see first guideline bullet, below) you'll need to add back in by hand.
3. Save the file as a web page by either:
    a. Saving your .ppt file as a web page (in the main menu, File > Save as Web Page). However, then you will have to work to make this HTML accessible, as discussed starting on page 35.
    b. Converting your file into accessible HTML using third-party tools, such as the wizard at www.accessiblewizards.uiuc.edu (see page 53). This is the better option.
4. Make web-based (HTML) presentations. This is the most accessible approach, both in terms of people with disabilities and for downloading times. However, it can also be more complex. Options include:
    a. Saving your slides as images and posting them with additional information. Good step-by-step instructions for this are available as part of the PowerPoint module of the free online Access E-Learning course, at www.accesselearning.net.
    b. Creating slide shows directly as web pages, without using PowerPoint. See "Alternatives to PowerPoint" at www.webaim.org/techniques/powerpoint/.

For all of these methods, you should start by following the general PowerPoint guidelines below, as well as all of the cross-cutting accessibility guidelines covered earlier.

Guidelines for Increasing PowerPoint File Accessibility

Using these tips will make your original PowerPoint more accessible to screen readers (for those who have access to the PowerPoint application), and improve the accessibility of any HTML conversion you later make.

- **Use the slide layout boxes** that PowerPoint automatically provides for Slide Titles and contents, rather than making your own text boxes. This creates better structure, including for creating an outline you can copy and paste into other formats.
- **Group related graphic objects**, such as text boxes and images. Such objects won't appear in the outline, and will need alt-text descriptions in the HTML version. By grouping the objects, when you convert to HTML you will only have to provide one alt-text description for the group. Figure 5 shows an example of how to group two images of mouth sticks.
- **Use the Slide Notes feature to provide more textual information**. You can provide more context and detail about each slide without adding to slides themselves.
- **Double check your structure.** To be sure you are using the title and bullet slide layouts well, look at the Outline view of your PowerPoint as described in 2b, above.

Figure 5: Grouping related graphic objects in PowerPoint

- **Create Charts and Tables using the PowerPoint menu** (Insert > Chart/Table). This makes the data accessible to screen readers within PowerPoint, and in any HTML conversion using a third-party conversion tool. (If you insert charts or tables from other sources, they will appear as images instead when converted. This means you'll have to provide alt-text descriptions for them.)

Resources on PowerPoint

You'll find WebAIM's succinct introduction to accessible PowerPoint use at www.webaim.org/techniques/powerpoint, with a full training module on PowerPoint as part of the course at www.accesselearning.net.

Two fact sheets, "Posting Accessible PowerPoint Presentations" and "PowerPoint Presentation into an Accessible Webpage (HTML File)," are at www.catea.org/grade/factsheets.php, with checklists at www.catea.org/grade/guides/powerpointmust.php.

IV. Making "Regular" Web Pages Accessible – HTML

Most web pages are written in HTML, or Hypertext Markup Language. HTML can be very accessible to people with nearly all kinds of disabilities, if you use it correctly. This section is designed to help you:

- Be aware of the accessibility features in common HTML authoring tools.
- Easily find references to support you when authoring or editing HTML for accessibility.
- Know how to create accessible HTML.

HTML Authoring Tools

You have a choice of many software packages that help you create HTML pages without knowing much, or anything, about HTML. For example, Dreamweaver and FrontPage let you create "what you see is what you get" (WYSIWYG) web pages, where you create content much as in a word processor, and the program converts it to HTML code for you. Such programs tend to have options that make it pretty easy to create *accessible* HTML, if you turn those features on.

Most Microsoft Office applications, like PowerPoint and Word, also let you convert to HTML, but this conversion rarely results in accessible HTML, and so requires HTML editing to meet the standards outlined in the next section. For making accessible conversions of these kinds of files, see the Conversion section (page 53).

Using Dreamweaver to Create Accessible HTML

Dreamweaver is one of the most common web authoring tools used at Cornell, and (for versions beyond 4.0) also has some of the best accessibility features.

The most important step is to *turn the accessibility prompts on*. Unfortunately, these are disabled by default. To turn them on, in the main menu, go to Edit > Preferences > Accessibility > and then select the four checkboxes (Form objects, Frames, Media, Images) as shown in Figure 6 below.

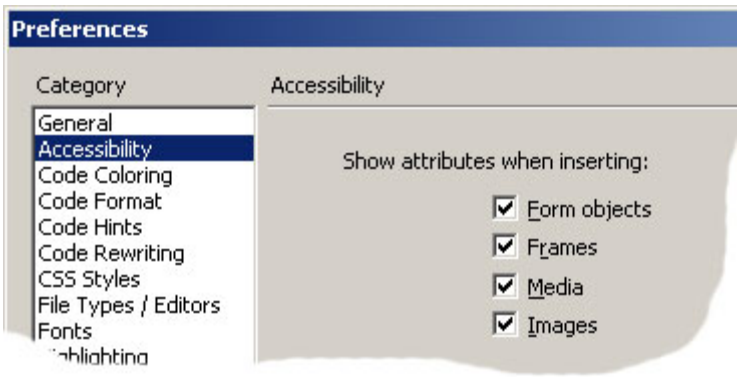


Figure 6: Turning on Accessibility prompts in Dreamweaver MX

With this feature enabled, as long as you are in the Design View (not in the Code View), Dreamweaver will prompt you to make various elements of your HTML accessible as you insert them into your file.

Resources on Dreamweaver

For an introductory, one-page guide to taking advantage of the basic Dreamweaver accessibility prompts, visit www.webaim.org/techniques/dreamweaver.

A more detailed tutorial, "Achieve Accessibility with Dreamweaver," includes helpful

screenshots, and is online at www.wise-women.org/tutorials/accessibility/. The 10 or so pages include the following topics: Set Preferences, Install Extensions, Source Order and Skip Navigation, Title and Longdesc, Relative Measures, Making Tables Accessible, Color and Contrast, Forms, Summary of Key Concepts.

Adobe provides more complete, and complex, details at:
www.adobe.com/resources/accessibility/dw8/

Microsoft's Editors – FrontPage and Expression Web Designer

Other common HTML editors include Microsoft's FrontPage and its successor, Expression Web Designer. The advantages of these packages are that they are cheaper than most other editors, and they integrate well with other Microsoft applications, such as Word. Both have accessibility features, though Expression Web Designer is superior to FrontPage for creating accessible HTML.

Resources on FrontPage and Expression Web

You'll find introductions to the accessibility features of FrontPage and Expression Web at www.webaim.org/techniques/frontpage/ and www.webaim.org/techniques/msew/.

HTML Accessibility

Each section below introduces individual aspects of authoring and editing HTML to make it accessible.

Content and Structure

As for all kinds of files, providing structure to your web pages helps screen readers make sense of the content. It also helps with accessibility on small screens like cell phones and palm pilots. This includes using real headings and lists, as well as following some hyperlinking guidelines.

Use Real Headings

As in Word documents, you can and should use heading styles to structure your page content. This lets screen reader users skim through the sections and helps organize your content for easier use by everyone.

Use the headings to create an outline-like hierarchy, with 1st degree headings (<h1>) being the most important (usually page titles or headings), then 2nd degree headings (<h2> - usually major section headings), down to 3rd degree headings (sub-sections of the <h2>), and so on.

Do not use formatting such as bolding or increased font size to create headings, but only the 'official' HTML heading tags (<h1>-<h6>. Likewise, do not use headings to achieve visual, formatting results, but only as section headers that outline your content.

> Check one of the Cornell pages you have had a hand in creating or editing. Enter the URL of the page in the Address field at http://validator.w3.org/detailed.html, check the "Show Outline" box, and then click the "validate this page" button. Skip to the bottom of the output to the Outline section. Does it have an outline, i.e., is your page outlined properly using heading tags?

Use Real Lists

HTML lists - , <ol>, and <dl> - also convey a hierarchical content structure. Each of these has rules regarding its use as well. Unordered lists (<ul>) should be used when there is no order of sequence or importance. Ordered lists (<ol>) suggest a progression or sequence. Definition lists (<dl>) should be used explicitly for presenting a structure for definitions. As with headings, lists should be used correctly and for the right purposes. Unordered and ordered lists should always contain list items. Definition lists must always have definition descriptions. Empty lists are incorrect HTML. Lists should never be used for merely indenting or other layout purposes. Nested lists should be coded properly.[9]

[9] This list section is directly from www.webaim.org/techniques/semanticstructure/. Much of the other content here on HTML also draws from these pages.

Links

Core guidelines for including hyperlinks in your pages include:

- Provide full keyboard-only access to links - i.e., users must be able to navigate to (usually via the Tab key) and select (usually via the Enter key) each link using the keyboard alone. HTML alone does not interfere with this. JavaScript event handlers (often to open drop down menus), Java, and Flash use in an HTML page can, however, create problems unless you design with accessibility in mind.[10]
- Make links readable for screen readers. A screen reader can be set to read only links and, regardless, will always have to read the entire text of the links. So screen reader accessibility includes:
  - Create link names that make sense out of context. E.g., “click here” and “more information” are useless as link names when read out of context by a screen reader.
  - Keep link names as short as possible, while still being descriptive.
  - If using graphical words (a drawing of a word) rather than real text as a link, simply provide alt text that reads out the graphical word. You don’t need to include the word “link” as screen readers tell users when something is a link.
  - Separate any side-by-side links with non-linked characters (e.g., with |), so screen reader (and other) users won’t confuse them as one link.
  - Use words and phrases linked to URLs, not actual URL addresses, unless the URL is very short.
- Don’t use color alone to identify links, since this is invisible to color-blind users. Underline links as well.
- Links should look like links, and nothing else should. Users may get frustrated if they try to click on textual phrases or graphics that look like links but are not. They will also be frustrated if they have to move their mouse all over the page trying to discover links that do not look like links. Do not underline text in an HTML document unless it is a link.

"Skip Navigation" Links

A screen reader will read everything on a page, starting at the top left. If your page headers start with dozens of navigation links, then screen reader users will have to sit and listen to each of these dozens being read aloud every time they load a page. That is, unless you provide a way to skip the navigation and get to the main content.

Options for allowing users to skip to main content include:

- Providing visible “skip navigation” or “skip to main content” links at the top of the

[10] Start at www.webaim.org/techniques/hypertext/ for more information if you are using these tools to create links.

page. This is the easiest method, and is highly effective.

- Providing visible “skip” links elsewhere on the page. This only works if you assign a ‘tabindex’ attribute to the link, so users reach it first via tabbing, no matter where it is on the page. It also assumes that a screen reader user is going to tab through the links before starting to read the full page.
- Making the link invisible to anyone but screen reader users. Put a small invisible .gif image at the top of the page with alt="Skip navigation" as the alternative text. (This, however, excludes use by people with limited mobility, who cannot use a mouse to scroll.)
- Another method is to use Cascading Style Sheets (CSS) to hide the link. This method is more elegant than the hidden graphic method. WebAIM has instructions at: www.webaim.org/techniques/css/invisiblecontent/#hidingfromsighted.

Find out more about these and other “skip” methods at www.webaim.org/techniques/skipnav.

Templates

Templates are reusable patterns or sections of HTML markup that can appear on multiple pages on a web site. It keeps sections such a header and sidebar navigation consistent, while you change only the main, or center, content, on each new page. Using templates has several advantages for accessibility for all users, and ease of designing and maintaining your pages, including:

- Making it easier for all users to navigate your pages once they have familiarized themselves with the basic design.
- Needing to create accessible navigation only once, and then being able to apply it to all pages.
- Letting you fix any problems with the template globally across your site, instead of having to fix each page.

Cornell provides some HTML templates at http://cornelllogo.cornell.edu/templates. These have accessible options such as including “skip navigation”. No template can ensure your site will be accessible – only you can do that, by using your judgment and following the guidelines here.

For more on using templates, see www.webaim.org/techniques/templates.

Forms

Badly designed forms can confuse all users, but mainly create problems for screen readers. Keys to accessible HTML form design include:

- Make the form available using only a keyboard, normally via the tab and enter keys. Among other things, this means no JavaScripts that change browser location.
- Organize forms logically. This includes providing clear instructions, labeling what is

required, and lining up each label (e.g., Last Name) next to the element being asked (e.g., the text box for last name; see Figure 7 below).

- Make sure the form is linear. I.e., order it so that a screen reader will read it as you intend. If you use a table to lay out the form, this can become an issue if you don't use HTML code to number what order the elements should be in. See the Tables section below.

Figure 7: Line up labels in HTML forms (image from S. Trerise)

- Use HTML form labels (<label>) to identify each element in the form. A screen reader will then read the text within the label aloud (e.g., "first name") to tell the user what to enter. Label everything but the buttons (screen readers will already read aloud any text on the button).
- If you have groups of questions in the form, associate them using a <fieldset>. Then describe each fieldset with a <legend>. See the sample HTML form code at the bottom of www.webaim.org/techniques/forms/screen_reader.php.
- Create accessible form controls (e.g., radio buttons, checkboxes). See www.webaim.org/techniques/forms/controls.php for instructions.

Resources on Creating HTML Forms

WebAIM's article on forms, at www.webaim.org/techniques/forms, includes sample HTML code as well as instructions for using Dreamweaver and FrontPage to create accessible forms. They also host an article on Usable and Accessible Form Validation and Error Recovery (www.webaim.org/techniques/formvalidation).

Another good introduction, "Accessible Forms," is at www.usability.com.au/resources/forms.cfm.

The HTML module of the online Access E-Learning tutorial lets you practice creating accessible HTML for building forms by updating inaccessible code. Sign up for free access at www.accesselearning.net.

Images in HTML Pages

Don't hesitate to use images in your designs. They make your content more accessible to almost all users. However, people with limited or no sight will need text descriptions of any image you include.

Options for Describing Images

Three options for describing images in your web pages include providing:

- **"Regular", descriptive alt text**. Use up to 150 characters to provide the same information for users who cannot see the image as those who can the image. This might range from a simple transcription of graphical text in logos (e.g., see Figure 8 below) to a few sentences to describe a painting to art students. Avoid using images as bullets. See the earlier section on alt-text (page 19) for more guidance on writing good image descriptions.
- **Null alt text**. If you do not need to describe an image, e.g., if it is simply decorative or if the description is the page content, then always insert empty or null alt text. While some screen readers will skip images without alt text, others will read out the name of the image file itself, which can be very confusing. For example, in the case of the logo in Figure 8, it would read out "cornell-underscore-01.jpg"). So for every image not needing a description, enter alt="". Note there is no space between the quotation marks.[11]
- **Long descriptions**. If your image requires more than 150 characters to describe (e.g., this is likely the case for explaining the data in a graph or chart), then provide a long description of it on a separate web page. Ideally, simply link the caption of the image to this descriptive page. However, another option is using the "longdesc" attribute of the HTML Image tag to link to the description, taking users to a webpage that describes the image in full[12].

Figure 8: The descriptive alt text for this logo might be "Department of Communication"

Also see the guidelines and resources on images in the "cross-cutting" section on page 19.

Image Maps

An image map is a picture that contains several links that lead to other Web pages, depending on where a user clicks on the image. The image area for each link is called a "hot spot".

You need to create alt text both for the image map overall and then for each hot spot. For example, in the map shown in Figure 9, you would create null alt text for the overall image, and then create alt text that repeats the text content for each of the 5 hot spots. The HTML coding would be as follows (the alt text pieces are bolded to highlight them:

Figure 9: Image Map with 5 Hot Spots

[11] Dreamweaver lets you set the alt text to "empty" in the Properties dialogue box, though with many other editors you will need to create null alt text manually for decorative and other images not requiring descriptions.
[12] E.g., sample HTML including a longdesc link is: <img src="graph.jpg" width="526" height="353" alt="Graph of percentage of total U.S. noninsitutionalized population age 16-64 declaring one or more disabilities" longdesc="media/description.htm">. This example is from www.webaim.org/techniques/images/longdesc.php.

repeats the text content for each of the 5 hot spots. The HTML coding would be as follows (the alt text pieces are bolded to highlight them:

```
<img src="imagemap.jpg" alt="" width="199" height="303" border="0" usemap="#Map">
<map name="Map">
<area shape="rect" coords="7,9,191,54" href="#maps" alt="HOME">
<area shape="rect" coords="7,68,191,114" href="#maps" alt="Products">
<area shape="rect" coords="7,127,190,172" href="#maps" alt="Services">
<area shape="rect" coords="6,186,190,229" href="#maps" alt="Contact us">
<area shape="rect" coords="7,245,189,289" href="#maps" alt="Index">
</map>
```

Unless you go out of your way to design it otherwise, all your image maps will be "client side," which means the user's browser processes the map. The alternative – a server-side map – is *not* accessible, so do not create server-side image maps.

Other Notes on Image Use

Additional guidelines on using images in HTML pages include:

- Don't use graphical text (images that include text) if you can avoid it because it distorts when enlarged (turn back to Figure 1 to see an example). If you must use it, make the font size large, use good color contrast and simple block fonts, and provide alt text.
- Don't include rapidly flashing images, as these can induce seizures in some people.
- Provide as much color contrast as possible, but don't use color alone to convey meaning.
- Don't put any images that convey content in the background, because you cannot add alt text to these.

For more on this topic, refer back to the Image section in the Cross-Cutting Strategies chapter (page 19).

Keyboard Accessibility

Make sure users can access all of your website content using just the keyboard. Blind people can't use a mouse at all. Also, many people with mobility impairments don't use a mouse, but instead use tools such as the head wand or mouth sticks on the keyboard (see photos in Figure 5). A keyboard accessible site may also make it more usable to people using small, portable web technologies.

Normally, keyboard-only users tab through links, and click 'enter' to go to the link destination. Most of the time, the default tab order will be the same as the visual order of your site. You don't have to do anything special to make it work.

Tabindex

In rare cases, such as in some online forms or when you use tables to create complex page layouts, you might need to create the correct tabbing order using <tabindex>. See

www.webaim.org/techniques/keyboard/tabindex.php for more information.

Common barriers

The main barriers to keyboard-accessible sites tend to include:

- JavaScript DHTML menus. Dynamic HTML (DHTML) is rarely keyboard-accessible. Web designers often use DHTML to create menus that appear when the mouse rolls over them. These might look good, but they are inaccessible without a mouse. You can use them, but only if you provide full, equivalent text navigation menus on the destination pages.
- WYSIWYG absolute positioning. Some WYSIWYG (What You See Is What You Get) editors, such as Dreamweaver, let you create absolute-positioned "div" or "span" elements that you can then click and drag around the page. This can put the visual order out of whack with the order in which keyboard-only users can tab through it, putting the page out of order for them. Make sure your visual and tab orders add up, using tabindex (above) if needed.

So avoid both.

Frames

Frames let designers put more than one webpage into the same visual space. For people who can see the page, a page with framesets looks like any other, single page. For people using screen readers, however, who cannot simply scan the page visually, using frames can be disorienting. Consider using CSS to achieve your desired effects instead, as discussed on page 47.

However, if you are committed to using frames, some strategies for making them more accessible include[13]:

- Give each frame an appropriate title, e.g., "navigation links" and "main content".
- Provide noframes content. Content in the noframes tag should always be available if the user cannot or chooses not to view frame content. The noframes content should indicate what the contents of the frames are and provide links to individual frame pages if appropriate.
- Use the correct document type. The proper frameset doctype lets screen readers and other browsers know that the document consists of multiple frames. See sample code at www.webaim.org/techniques/frames.

Tables

In HTML, you can use tables for data presentation or for layout. You should review and follow the general guidelines on tables discussed on page 18, including defining column and row size

[13] The content here on frames comes from www.webaim.org/techniques/frames/.

using percentages, not fixed measurements.

Data Tables

Guidelines for creating data tables include:

- Keep data tables as simple as possible, with no double sets of row or column headings, and no headings that span more than one column or row.
- Explain the contents of complex tables with alternative text, long descriptions, or the summary tag.
- Designate row and/or column headers with the table header (<th>) tag in data tables (but not in layout tables.)
- Associate the headers with the correct rows and columns using the scope attribute (e.g., <th scope="col">Name</th> to identify and associate a column called “name”). In more complex tables, you can use header and ID attributes to associate rows and columns with headers.
- Use the caption tag to describe, or name, the table. (E.g., <caption>Biology 100 Grades</caption>

Layout Tables

While HTML tables were created originally for housing data, they are probably more often used to create page layouts containing other kinds of content. Cascading Style Sheets (CSS), discussed below, let you design your page appearance more flexibly, easily and accessibly than using tables. However, it is possible to create accessible pages using tables for layouts. Guidelines include:

- Keep layout tables as simple as possible. The lines may be invisible to sighted users, but a screen reader will tell their users about every single row and column, which gets disorienting quickly in a complex layout.
- Do not use table header tags in layout tables.
- Make sure a screen reader will read it in the order you intend. No matter what your table looks like, the screen reader just reads the content in the literal order that it appears in the code. Make sure that literal order is the same as the logical order

Resources on tables

You’ll find a great introduction, including sample HTML code, at www.webaim.org/techniques/tables.

For designing data tables, the article “Accessible Data Tables” at www.usability.com.au/resources/tables.cfm is helpful.

The tool at www.accessify.com/tools-and-wizards/accessibility-tools/table-builder/ actually

creates accessible data table HTML for you if you enter your requirements, such as the number of rows and columns.

Resources on HTML Authoring

Find detailed techniques and instructions at www.w3.org/TR/WCAG10-HTML-TECHS. The Access E-Learning course module on HTML is also helpful: www.accesselearning.net.

V. Beyond HTML: CSS, JavaScript, Plug-ins

This section introduces a few other languages and applications that web designers often use in addition to HTML when creating web sites. It highlights core accessibility issues and strategies for each, and points to outside resources for more information and guidance.

Cascading Style Sheets (CSS)

You can use HTML to control the structure, content and presentation of your web pages. Alternatively, you can leave structure and content to HTML, but control appearance instead with another markup language, Cascading Style Sheets (CSS, or .css files).

Using CSS (also sometimes simply called "style sheets") to create your page presentation separates page appearance from the page's content and structure. This has several advantages, including:

- Web designers no longer have to embed formatting with HTML code around each piece of format. Instead, they can create the formatting rules in a separate style sheet. For example, if a designer wants to increase font size in all paragraphs, in HTML she would have to insert this change in the markup for each paragraph. Using a CSS, she would just have to update the sheet in one place and the change would cascade throughout the affected paragraphs.
- Users can choose to apply their own style sheet to your site, instead of your sheet, to meet their own access needs. For example, someone with low vision could easily increase the color contrast and font size.

Once you learn how to use it (and it's not that hard), CSS lets you control your webpage presentation more flexibly and easily, while (potentially) increasing accessibility. Compared with using tables or frames, CSS makes it much easier to control your layout while keeping the content linear and, therefore, accessible to screen readers. You can also use hidden CSS that won't change your visual impact while making your sites more accessible to users with disabilities.

Figure 10 shows a web page with the host site's style applied, and then how it looks with the styles removed. For pages styled with CSS such as this one, a user could then apply their own style to make it more useable for them. For example, someone with low vision might set the page to show a large, white font on a black background.

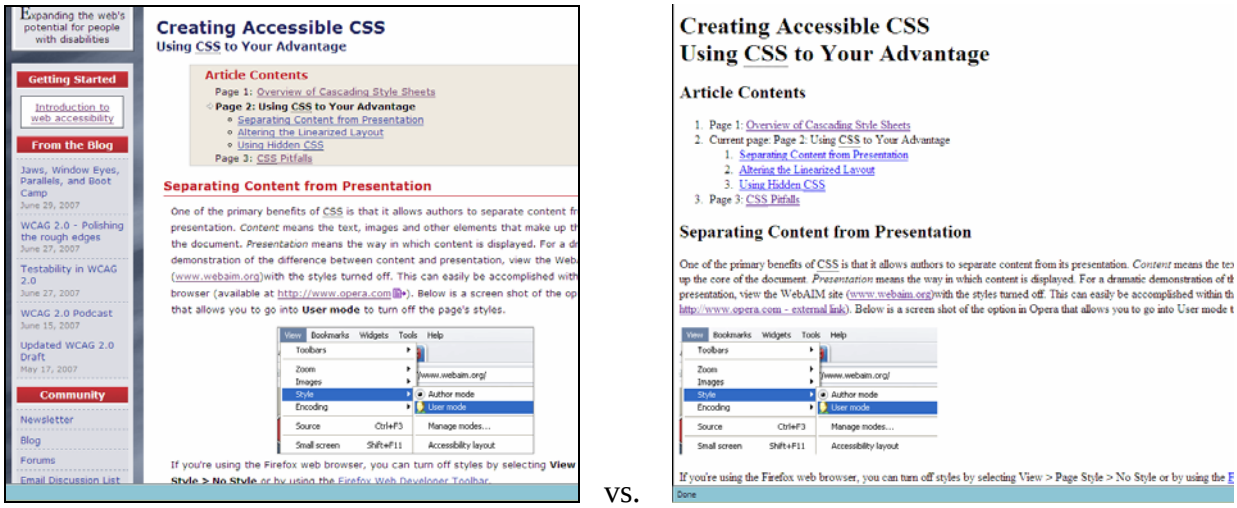


Figure 10: A WebAIM webpage. The first image shows it with the WebAIM CSS style sheet applied. The second shows the page with styles removed.

Note that even when using CSS, the content and structure you provide with HTML needs to follow the same guidelines as when you aren't using CSS.

Also, don't make your pages dependent on CSS – it is *only* for presentation and appearance. If you try to use it to control page structure, your pages will become less accessible.

Resources on CSS

See WebAIM's introduction to style sheets at www.webaim.org/techniques/css, including a section on using CSS to improve accessibility with content that is invisible to users who don't need it. The Access E-learning tutorial includes some tips on CSS within their HTML course module at www.accesselearning.net. The article at www.mcu.org.uk/articles/tables.html discusses the virtues of CSS vs. tables for layout.

W3schools offers CSS tutorials, 70 examples, and a CSS reference listing at www.w3schools.com/css. Also, Cornell's CIT offers face-to-face training sessions on Getting Started with CSS. Visit the technical training schedule available at www.cit.cornell.edu/training to see when the next workshop is.

JavaScripts

Of the scripting languages, JavaScript may be the most popular. Web designers often embed JavaScripts into their HTML pages to get web pages to do things HTML alone cannot do. For example, designers use it to create pop-up windows, check that all required fields in a form have been filled in, or display changes when you put your mouse over an image.

JavaScript Accessibility Issues[14]

JavaScript allows developers to add increased interaction, information processing, and control in web-based content. It can even be used to increase accessibility, such as by providing warnings on pages that require a user response or will otherwise time out. However, JavaScript can also introduce accessibility issues. These issues include:

- **Navigation** – barriers to navigating using a keyboard or assistive technology.
- **Hidden content** – content or functionality not accessible to assistive technologies.
- **User control** – lack of user control over automated content changes.
- **Confusion/Disorientation** – altering or disabling the normal functionality of the user agent (browser) or triggering events that the user may not be aware of.

Accessible approaches and solutions are:

- When using event handlers, use only those that are device independent (e.g., do not require the use of the mouse only).
- Content and functionality that is provided through scripting must be made accessible to assistive technologies.
- Web pages that use scripting must be fully navigable using a keyboard.
- JavaScript should not modify or override normal browser functionality in a way that may cause confusion.
- When JavaScript cannot be made natively accessible, an accessible alternative must be provided, such as with the NOSCRIPT element.

Resources on JavaScript

How to implement each of the approaches and solutions listed above is covered at www.webaim.org/techniques/javascript/. WebAIM also includes a piece on accessible AJAX, www.webaim.org/techniques/ajax/. Additional information on accessibility and AJAX is located at www.washington.edu/computing/accessible/accessibleweb/ajax_accessible.html. You will find a thorough 3-hour self-guided tutorial about accessible scripting, including JavaScripts, as the last module in the free online Access E-Learning course at www.accesselearning.net.

[14] This section comes directly and indirectly from www.webaim.org/techniques/javascript

Plug-ins

Plug-ins are computer programs that interact with a host application, e.g., a web browser, to accomplish a task that the host cannot do on its own. Plug-ins might be used to read particular kinds of files (e.g. using QuickTime to show a video), or play and watch presentations in browser (e.g., watching a Flash presentation).

Three things for web developers or designers to keep in mind about plug-ins are:

- The plug-in itself must be accessible, i.e., it should work well with assistive technologies. Check with the company that makes the plug-in, or refer to some of the resources below.
- The content that the plug-in presents needs to be accessible, i.e., it should follow all the other guidelines covered here. For example, most images need alternative text, text should be high contrast, all audio requires transcripts, and all video requires timed captions.
- Don't make your important site content or functionality dependent on plug-ins. For example, do not use Flash-only navigation.

Making plug-ins more accessible

Several things you can do to make plug-ins generally more accessible include:

- Link to the site where the relevant plug-in can be downloaded.
- When embedding plug-in content and nesting OBJECT and EMBED tags, put the OBJECT tag at the outermost level for Internet Explorer and next the EMBED tag for Netscape browsers.[15]

Resources on Plug-ins

WebAIM compares media players (RealMedia Player, RealOne, Quicktime, and Windows Media Player) in this article: www.webaim.org/techniques/captions/mediaplayers. They recommend using players as standalone, rather than embedding them, to make them accessible.

WebAIM also hosts an article about using Flash accessibly, www.webaim.org/techniques/flash/. The "Flash and Accessibility" article at www.usability.com.au/resources/flash.cfm provides a bit more detail, and you'll find checklists at www.catea.org/grade/guides/flashmust.php. Adobe's Flash Accessibility information is at www.adobe.com/accessibility/products/flash.

[15] Much of the content in this plugin section is adapted from the Plugin section of the HTML self-training module at www.accesselearning.net. That section includes sample HTML for this embedding.

VI. CommonSpot and BlackBoard

Many in the Cornell community use CommonSpot to manage their site content and BlackBoard for course pages. This section covers how to make sites that you maintain with these tools more accessible.

CommonSpot

CommonSpot is a content management system (CMS) for developing and maintaining web pages, produced by the company PaperThin. Many Cornell units have adopted CommonSpot as their CMS.

You can make your CommonSpot-generated sites more accessible by:

- Turning on the two accessibility options in the author mode. One prompts you to add alt-text for every image, and the other helps you set up accessible tables.
- Make sure any downloadable files (e.g., PDFs) you upload are accessible (see the section about this that starts on page 5)

While you can make accessible websites using CommonSpot, currently the authoring mode in CommonSpot is not accessible. Cornell is working with the vendor, PaperThin, to improve this.

Requiring Alt-text in CommonSpot

To require that your website content editors must provide alternative text to every image added to your CommonSpot site, do the following:

1. Log in as a CommonSpot administrator at http://author.yoursite.cornell.edu/admin.cfm.
2. Click the "[your website] Site Admin" option in the top menu navigation.
3. Open the "Accessibility" settings and launch the Accessibility Properties popup.
4. Check the "Require the Alt attribute on images and objects" box Figure 11.

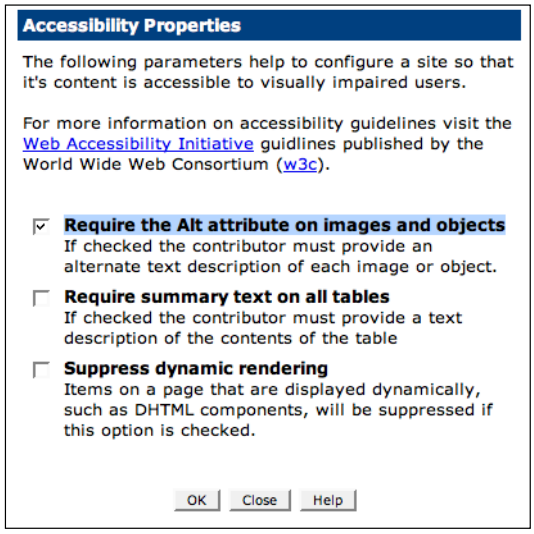


Figure 11: CommonSpot Accessibility Properties pop-up

Tables in CommonSpot

When adding a table in CommonSpot, add a brief "Description (Summary)" (see Figure 12). The summary is simply a short description of the table. For example, for a calendar, the summary can be as simple as "Monthly calendar with links to each day's posts."

If you use tables for layout, you should give each of those tables an empty summary, to indicate that the table is used exclusively for visual layout and not for presenting tabular data.

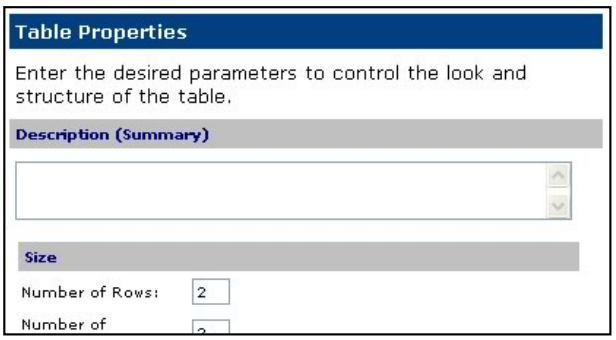


Figure 12: CommonSpot Table Properties Menu

To add Headers (<th></th>) to Tables:

1. Use Internet Explorer 6. Currently only IE6 allows the addition of headers to tables in CommonSpot.
2. Add table with appropiate table columns and rows, with the table summary.
3. Right click on the top row and select "New Row Properties."
4. In the "New Row Properties" dialogue check the "Header Row" box (see Figure 13).
5. Add text into table header cells.

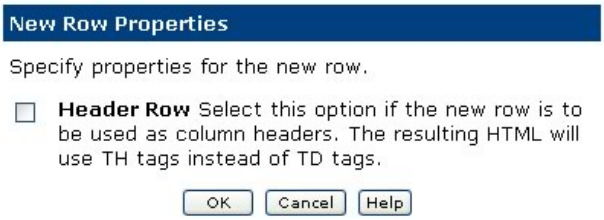


Figure 13: CommonSpot Row Properties Box

BlackBoard

Currently, Cornell has a license for using BlackBoard for managing online course content. However, BlackBoard is not as accessible as it could be, and other options are under review. In the meantime, the most important things you can do are to:

- If you are using BlackBoard, make sure any downloadable files (e.g., PDFs) you upload are accessible (see the section about this that starts on page 5).
- Evaluate the needs of your students. If BlackBoard cannot meet their needs, consider using Cornell's course site service instead, where you can design your own site. See www.cit.cornell.edu/atc/itsupport/webservices.shtml for more information.

VII. Choosing and Converting File Types

No one file type provides the answer to accessibility. Knowing what you do now about the plusses and minus of different kinds of files (e.g., Word, PowerPoint, HTML), you are the best judge about which best fits your content, your needs, the needs of your audience, your skill set and your budget.

Sometimes it will make sense to convert most or all of your content to HTML. Sometimes you'll be able to work with the source file, using what you have started to learn here, to make it accessible directly as a download.

If you do want to convert a Microsoft Office file into accessible HTML, your best bet might be the Illinois Accessible Web Publishing Wizard for Microsoft Office, online at www.accessiblewizards.uiuc.edu. You can test it for free, and a license costs about $40.

For converting PDFs to HTML, Adobe offers a free online tool at: www.adobe.com/products/acrobat/access_onlinetools.html.

Appendix A – 508 Guidelines from A to P

Section 508 of the Rehabilitation Act, last amended in 1998, requires that the federal government make all of its information technology accessible to people with disabilities. Only part of it deals with the Web. Cornell is adopting these Section 508 web standards for making our web sites accessible.

The Section 508 Guidelines for the Web

Each of these 16 "rules" is covered in this workbook, albeit not exactly in this way or order. Refer back to the Table of Contents to find information on each.

These are taken directly from www.section508.gov and are the core of Cornell's draft accessibility policy. Note that this listing reflects neither how common nor how important a guideline might be. For example, making your downloadable files accessible will likely take up much of your time, but that "rule" is relatively buried in guideline "m".

> A compliance mind-set, where you mindlessly check off each "rule" that you are following without thinking through your design with disabled users in mind, is unlikely to lead to a useable site. So as important as these rules and guidelines are, they are not sufficient. Think access.

The "Rules"

a) A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content).

(b) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.

(c) Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.

(d) Documents shall be organized so they are readable without requiring an associated style sheet.

(e) Redundant text links shall be provided for each active region of a server-side image map.

(f) Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.

(g) Row and column headers shall be identified for data tables.

(h) Markup shall be used to associate data cells and header cells for data tables that have two or

more logical levels of row or column headers.

(i) Frames shall be titled with text that facilitates frame identification and navigation.

(j) Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.

(k) A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.

(l) When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.

(m) When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with §1194.21(a) through (l).

(n) When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.

(o) A method shall be provided that permits users to skip repetitive navigation links.

(p) When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.

Resources on Section 508

Full information on Section 508 is online at www.section508.gov. From there, look up the 508 Law Technical Standards on "1194.22 Web-based intranet and internet information and applications."

WebAIM has a nice chart detailing what would "pass" and "fail" against each 508 rule at www.webaim.org/standards/508/checklist.php. The University of Wisconsin provides an extensive online library of 508-centered resources including:

- A detailed explanation of each rule, a-p: http://helpdesk.wisc.edu/accessibility/guideline/508guidelines.html
- Good and bad examples of each rule being fulfilled (or not): www.cew.wisc.edu/accessibility/evaluation/section508presentation.htm
- An online course on web accessibility, organized by 508 guidelines: www.doit.wisc.edu/accessibility/online-course/start.htm

Appendix B – Related Accessibility Resources

This appendix compiles all the topic-specific resources mentioned throughout this workbook and provides an overview of general resources on web accessibility, as well as what is available at Cornell.

Of course, the web is a moving target, so if you find that any of these sites are no longer operational, or have another good resource to suggest, please contact the Office of Web Communications.

Cornell Resources

Join the new email discussion list on Web accessibility, webaccessibility-l. This list disseminates progress on the draft policy, training and education opportunities, and other developments in this area. It is also a forum for you to ask question and discuss accessibility issues. To join the list, send an email to webaccessibility-l-request@cornell.edu with the word "join" (no quotes) in the body.

Cornell provides some HTML templates at http://cornelllogo.cornell.edu/templates. These have accessible options such as including “skip navigation.”

Cornell’s draft web accessibility policy is at www.cit.cornell.edu/policy/drafts/WebAccess.html.

This workbook and announcements about associated training sessions are online under “Web Accessibility” at www.cit.cornell.edu/policy/framework-chart.html

General Resources

Web Accessibility in Mind (www.webaim.org), or WebAIM, may be the best single introduction to online accessibility. It hosts an extensive collection of articles and how-tos on accessibility generally and on specific applications and file types. WebAIM also provides insight into how people with different kinds of disabilities access the web, including many simulations and videos.

Checklists

If you like checklists, you might find these sites helpful:

- This Quick Accessibility Checklist is meant to help faculty and staff who want to develop or modify Web-based course material, lectures, and assignments in an accessible way: http://tlt.its.psu.edu/suggestions/accessibility/check.html.

- These Guidelines for Accessible Distance Education categorize strategies into musts, shoulds and mays for PDF, Excel, Flash, PowerPoint, Video and Word: www.catea.org/grade/guides/introduction.php.

Link Collections

Some nice resource list collections include:

- A collection of accessibility papers and other articles written by consultant Roger Hudson, along with and links to other sites and tools: www.usability.com.au/resources
- An incredibly long list of websites on nearly all accessibility topics. This is particularly good if you are looking for something particular and don't know where to start: www.d.umn.edu/itss/support/Training/Online/webdesign/accessibility.html
- A nicely annotated list of "user experience design resources" about accessibility is at www.deyalexander.com/resources/uxd/accessibility.html. It includes links to many contextualizing discussion articles.

Online Courses and Tutorials

Online courses and tutorials covering web accessibility generally include:

- A Section 508 Web Accessibility Tutorial includes sections on text alternatives, checking accessibility, navigation, image maps, audio and multimedia, color/flicker use, forms, tables, scripts and applets, style sheets, and 508 provisions: www.jimthatcher.com/webcourse1.htm
- The Access E-Learning tutorial, available via free registration, provides extensive information and practice labs on disabilities, creating an accessibility plan, PowerPoint, Video, Flash, Word, Excel, PDF, HTML and Scripts/Java: www.accesselearning.net.
- "Web Accessibility 101" course is organized by 508 guidelines, and also includes information on evaluating for accessibility: www.doit.wisc.edu/accessibility/online-course/start.htm.
- The New York State Forum's IT Accessibility Committee created an "Accessibility Curriculum" in 2005, also available online. It includes introductory information and covers images, cascading style sheets, tables, forms, scripting, PDF, XML, multimedia tools, usability and validation/evaluation. The curriculum includes useful sample files to demonstrate each concept, and HTML and PowerPoint versions of each curriculum topic: www.nysfirm.org/accessibility/resources/curriculum.

Resources by Topic

These are the resources pointed out throughout this workbook, organized by topic.

Introductions to Accessibility

The succinct web-based presentation "A Quick and Dirty Introduction to Accessibility," at www.maxdesign.com.au/presentation/accessibility, provides an excellent introduction to web accessibility, along with links to explore assistive technologies available to people with different kinds of disabilities.

WebAIM's introduction to web accessibility at www.webaim.org/intro is also a good starting point, with further links to experiences and needs specific to particular kinds of disabilities. Their series of articles on the user's perspective at www.webaim.org/articles provides great insight into the way people with various kinds of disabilities access the web, or can be prevented from accessing by bad web design. The articles often include short videos and exercises as well.

Specific Disabilities and Web Access

WebAIM has extensive resources, including simulations, about making the web accessible to people with visual disabilities. For example, visit www.webaim.org/techniques/screenreader/ for more details on screen readers in particular, with links to further resources available at the end.

WebAIM's section on deafness, at www.webaim.org/articles/auditory, provides some insight into hearing loss and how this impacts web use.

The short article "Making the Web Accessible for the Deaf, Hearing and Mobility Impaired," at www.samizdat.com/pac2.htm, provides some quick tips on accessible design for people with auditory or motor disabilities. The short article "Web Design for the Mobility-Impaired," at www.ptvguy.com/accessibility-web-design-for-the-mobility-impaired, provides some additional tips.

The authors of "An Accessibility Frontier: Cognitive disabilities and learning difficulties" provide guidance in these areas at www.usability.com.au/resources/cognitive.cfm.

Accessibility Standards

The two main sets of accessibility standards for the web are the Federal government's **Section 508** and the World Wide Web Consortium's (**W3C**) **Web Accessibility Initiative** (WAI) (www.w3.org/WAI).

Cornell is following the Section 508 standards. Full information on Section 508 is online at www.section508.gov. From there, look up the 508 Law Technical Standards on "1194.22 Web-based intranet and internet information and applications."

WebAIM has a nice chart detailing what would "pass" and "fail" against each 508 rule at www.webaim.org/standards/508/checklist.php. The University of Wisconsin provides an extensive online library of 508-centered resources including:

- A detailed explanation of each rule, a-p: http://helpdesk.wisc.edu/accessibility/guideline/508guidelines.html
- Good and bad examples of each rule being fulfilled (or not): www.cew.wisc.edu/accessibility/evaluation/section508presentation.htm
- An online course on web accessibility, organized by 508 guidelines: www.doit.wisc.edu/accessibility/online-course/start.htm

Evaluation Resources

WebAIM, a non-profit organization within the Center for Persons with Disabilities at Utah State University, provides a series of articles on evaluating accessibility (see the Evaluation section of www.webaim.org/articles), which includes a useful review of free online tools (www.webaim.org/articles/freetools). They also host their own free evaluation tool, WAVE, at www.wave.webaim.org.

Checking for Colorblindness Accessibility

Vischeck, www.vischeck.com, lets you assess your site for accessibility to people who have limited or no color vision.

Checking for HTML Headings

Enter the URL of the page you want to check at http://validator.w3.org/detailed.html, check the "Show Outline" box, and then click the "validate this page" button. Skip to the bottom of the output to the Outline section.

Images and Alt-text

WebAIM provides a great introduction to image accessibility at www.webaim.org/techniques/images/, which includes useful examples of good alt text for the same images in different contexts. For additional succinct yet thorough guidelines on writing good alt text, turn to the article "Writing good ALT text" at www.gawds.org/show.php?contentid=28.

Resources on flickering images

For more information and links to examples (which you should not click on if you are susceptible to seizures) visit www.webaim.org/articles/seizure.

Transcripts and Captions for Audio/Visual Media

WebAIM's article on captions at www.webaim.org/techniques/captions provides details on captioning options and technologies that can help you create synchronized, accessible captions.

See also the checklists on video use at www.catea.org/grade/guides/videomust.php.

Writing

For more tips on clear writing, and in particular writing for people with cognitive disabilities and reading disorders, visit www.webaim.org/techniques/writing.

Acronyms

A good article on how to handle acronyms, particularly in HTML contexts, is at www.alistapart.com/articles/hattrick.

Microsoft Word

As always, WebAIM offers great tips on making Word documents accessible, including some information on converting to HTML, at www.webaim.org/techniques/word.

If you like checklists, you'll like the "musts", "shoulds" and "mays" of accessible Word use at www.catea.org/grade/guides/wordmust.php.

The free online Access E-Learning course, at www.accesselearning.net, has an entire tutorial with detailed instructions and practice labs on this topic.

Adobe PDFs

If you decide to post PDF files online, a good introductory how-to guide for Acrobat 6.0 is California Community Colleges High Tech Center Training Unit's "Creating Accessible PDFs," at www.htctu.fhda.edu/trainings/manuals/web/Creating_Accessible_PDFs.pdf. Checklists of musts, shoulds and mays start at www.catea.org/grade/guides/acrobatmust.php, with a full how-to tutorial on PDF as a module within www.accesselearning.net.

The University of Wisconsin offers advice specifically for faculty and for librarians, in addition to step-wise instructions for evaluating the accessibility of a PDF and for producing PDFs from Word documents or scanned documents (www.cew.wisc.edu/accessibility/tutorials/PDF-L2.htm).

At www.webaim.org/techniques/acrobat/understandingtags.php, WebAIM explains how to add tags to new and existing PDF documents. For complete instructions (100+ pages), visit http://www.adobe.com/accessibility/ and download Adobe's own guide to "Creating Accessible PDF Documents with Adobe Acrobat" for Adobe 7.0.

Adobe's free Make Accessible Plug-in for tagging PDFs is online at www.adobe.com/support/downloads/detail.jsp?hexID=88de. However, the effectiveness of this tagging has received mixed reviews from web developers interested in accessibility.

Adobe also offers a free tool for converting PDFs to HTML at www.adobe.com/products/acrobat/access_onlinetools.html.

Microsoft Excel

You can take a full online course module on Excel at www.accesselearning.net or review checklists for accessible Excel spreadsheets at www.catea.org/grade/guides/excelmust.php.

Microsoft PowerPoint

You'll find WebAIM's succinct introduction to accessible PowerPoint use at www.webaim.org/techniques/powerpoint, with a full training module on PowerPoint as part of the course at www.accesselearning.net.

Two fact sheets, "Posting Accessible PowerPoint Presentations" and "PowerPoint Presentation into an Accessible Webpage (HTML File)," are at www.catea.org/grade/factsheets.php, with checklists at www.catea.org/grade/guides/powerpointmust.php.

HTML

Find detailed techniques and instructions at www.w3.org/TR/WCAG10-HTML-TECHS. The Access E-Learning course module on HTML is also helpful: www.accesselearning.net.

FrontPage and Expression Web

You'll find introductions to the accessibility features of FrontPage and Expression Web at www.webaim.org/techniques/frontpage and www.webaim.org/techniques/msew.

"Skip Navigation" Links

Find out more about these and other "skip" methods at www.webaim.org/techniques/skipnav.

Templates

Cornell provides some HTML templates at http://cornelllogo.cornell.edu/templates. These have accessible options such as including "skip navigation". Of course no template can ensure your site will be accessible – only you can do that, by following your judgment and the guidelines here.

For more on using templates, see www.webaim.org/techniques/templates.

Forms

WebAIM's article on forms, at www.webaim.org/techniques/forms, includes sample HTML code as well as instructions for using Dreamweaver and FrontPage to create accessible forms. WebAIM also hosts an article on Usable and Accessible Form Validation and Error Recovery (www.webaim.org/techniques/formvalidation).

Another good introduction, "Accessible Forms," is at www.usability.com.au/resources/forms.cfm.

The HTML module of the online Access E-Learning tutorial lets you practice creating accessible HTML for building forms by updating inaccessible code. Sign up for free access at www.accesselearning.net.

Tabindex

In rare cases, such as in some online forms or when you use tables to create complex page layouts, you might need to create the correct tabbing order using <tabindex>. See www.webaim.org/techniques/keyboard/tabindex.php for more information.

Frames

Use the correct document type. The proper frameset doctype lets screen readers and other browsers know that the document consists of multiple frames. See sample code at www.webaim.org/techniques/frames.

Tables

You'll find a great introduction, including sample HTML code, at www.webaim.org/techniques/tables.

For designing data tables, the article "Accessible Data Tables" at www.usability.com.au/resources/tables.cfm is helpful.

The tool at www.accessify.com/tools-and-wizards/accessibility-tools/table-builder/ actually creates accessible data table HTML for you if you enter your requirements, such as the number of rows and columns.

Cascading Style Sheets (CSS)

See WebAIM's introduction to style sheets at www.webaim.org/techniques/css, including a section on using CSS to improve accessibility with content that is invisible to users who don't need it. Access E-learning includes some tip on CSS within their HTML course module at www.accesselearning.net. The article at www.mcu.org.uk/articles/tables.html discusses the virtues of CSS vs. tables for layout.

W3schools offers CSS tutorials, 70 examples, and a CSS reference listing at www.w3schools.com/css. Also, Cornell's CIT offers face-to-face training sessions on Getting

Started with CSS. Visit the technical training schedule available at www.cit.cornell.edu/training to see when the next workshop is.

JavaScript

How to implement each of the approaches and solutions listed in the JavaScript section is covered at www.webaim.org/techniques/javascript/. WebAIM also includes a piece on accessible AJAX, www.webaim.org/techniques/ajax/. Additional information on accessibility and AJAX is located at www.washington.edu/computing/accessible/accessibleweb/ajax_accessible.html. You will find a thorough 3-hour self-guided tutorial about accessible scripting, including JavaScripts, as the last module in the free online Access E-Learning course at www.accesselearning.net.

Plug-ins

WebAIM compares media players (RealMedia Player, RealOne, Quicktime, and Windows Media Player) in this article: www.webaim.org/techniques/captions/mediaplayers. They recommend using players as standalone, rather than embedding them, to make them accessible.

WebAIM also hosts an article about using Flash accessibly, www.webaim.org/techniques/flash/. The "Flash and Accessibility" article at www.usability.com.au/resources/flash.cfm provides a bit more detail, and you'll find checklists at www.catea.org/grade/guides/flashmust.php. Adobe's Flash Accessibility information is at www.adobe.com/accessibility/products/flash.

Converting Files

If you want to convert a Microsoft Office file into accessible HTML, your best bet might be the Illinois Accessible Web Publishing Wizard for Microsoft Office, online at www.accessiblewizards.uiuc.edu. You can test it for free, and a license costs about $40.

For converting PDFs to HTML, Adobe offers a free online tool at: www.adobe.com/products/acrobat/access_onlinetools.html.