

Accessibility in e-Learning
Easy-to-Use Resources

JULY 2014

Contents

Accessibility Evaluation	3
Accessibility Checkers	3
Automated Accessibility Testing.....	3
WAVE	3
Open Ajax Accessibility (OAA) Extension.....	3
Accessibility Developer Tools.....	3
Colour Contrast Testing	3
Readability Testing	3
HTML Markup Validation	4
W3C Markup Validator	4
W3C CSS Validator	4
Captioning Tools.....	4
Amara Caption/Subtitle Editor	4
Capscribe Caption Editor	4
Screen Reader Testing	4
NVDA Screen Reader	5
ChromeVox Add-on for the Chrome Web Browser.....	5
Web Accessibility Courses (Free)	5
Understanding Web Accessibility (OCAD University)	5
Accessibility of e-Learning (Open University)	5
Introduction to Web Accessibility (Google).....	5
Accessibility in e-Learning YouTube Channel	6
Quick Manual Accessibility Tests	6
Creating Accessible e-Learning Content	6

Accessibility Evaluation

Accessibility Checkers

[Automated Accessibility Testing](#)

AChecker is an automated accessibility testing tool based on the W3C Open Accessibility Checks (OAC). Enter a website's URL, paste HTML, or upload an HTML file to have its accessibility checked against the W3C Web Content Accessibility Guidelines (WCAG). AChecker will identify **known problems** that it can identify with certainty, **likely problems** that appear to be a problem but need a decision from a person, and **potential problems** that the checker cannot identify but directs the reviewer to check the issue manually.

[WAVE](#)

Developed at WebAIM (a leading organization that promotes web accessibility), the WAVE Web-Based Accessibility Review Tool provides an icon-based visual presentation in its accessibility reviews.

[Open Ajax Accessibility \(OAA\) Extension](#)

This is an add-on for the **Firefox** web browser; it creates a sidebar in Firefox that inspects the accessibility of the web page being viewed for compliance with WCAG 2.0 success criteria.

[Accessibility Developer Tools](#)

This extension installs on the **Chrome** web browser. To access these tools, you would open the menu in the top right corner of the browser, and select Tools>Developer Tools; then in the developer tools area that opens in the lower half of the browser window, choose the Audit tab and then choose accessibility from the available options.

[Colour Contrast Testing](#)

The Juicy Studio Luminosity Colour Contrast Ratio Analyser is a web-based tool that can help determine if text that appears over a background will be readable by users with poor vision. WCAG Guideline 1.4.3 requires that text colour and background colour have a contrast ratio of 4.5:1 for standard paragraph text, and a contrast ratio of 3:1 for larger heading text. Also, when links are embedded in surrounding text and the standard underline is not present, the contrast ratio between text colour and the link colour must be 4.5:1.

[Readability Testing](#)

This web-based tool evaluates the level of reading skill required to effectively understand the context of a website. Readability is a level AAA requirement of the WCAG 2.0, and therefore not a requirement for the *Accessibility of Ontarians with Disabilities Act (AODA)*. However, content developers should strive to use the simplest language possible when creating content for the general public; ideally, content should be written at a grade 9 or 10 reading level. If your website's audience is more defined and advanced, use language appropriate for your audience, but still be mindful of using simple words wherever possible.

HTML Markup Validation

[W3C Markup Validator](#)

This tool reviews a web page's underlying HTML for conformance with the HTML standard; it identifies errors in the HTML and potential accessibility problems, which can include missing Alt text and duplicate "id" attributes. Most HTML errors will not affect accessibility; however, if significant in number, errors tend to interact and cause assistive technologies to behave inconsistently. Ensuring that your valid HTML is clean (error-free) reduces the possibility of barriers being introduced.

[W3C CSS Validator](#)

Jigsaw is the W3C Cascading Style Sheet (CSS) Validator. Use this tool to check a website's CSS for errors.

Captioning Tools

[Amara Caption/Subtitle Editor](#)

Amara is a crowd-sourcing type web application that allows individuals to collaborate and create – or improve upon – a video's captions or subtitles for a website. You can copy the web address (URL) of the video you want to caption and paste it into Amara to open the video in the caption editor. Once you have created the captions for your video, you can export them as a text file with the ".srt" caption file extension. This file can be imported into your video-editing program to add as a caption track, or it can be imported directly into video services such as [YouTube](#) or [Vimeo](#).

[Capscribe Caption Editor](#)

If you use a Mac, Capscribe is a good option for captioning video. One handy feature is Capscribe's ability to add audio description (described video) directly into the captions. It will output a description track using text-to-speech technology that you can add to your videos to describe the meaningful action in the video for learners who are blind.

Screen Reader Testing

[JAWS Screen Reader](#)

One of the most popular screen readers is **JAWS** (Job Access for Windows) developed by Freedom Scientific [21]. A licence for the multi-featured JAWS falls in the \$1,000 range, typically more than most content developers want to spend. However, you can download a free demo version from the developer's website; it will run for 40 minutes, after which you would need to reboot your computer if you want to run it again for 40 minutes, and so on. JAWS is a complex application that takes some time to learn. But if you plan to use a screen reader regularly, or are part of a group of people who want to test content with a screen reader, purchasing a JAWS license may be a good investment. To find out if JAWS is right for your needs, you should review [documentation on JAWS](#).

[NVDA Screen Reader](#)

If you prefer to use a full-featured screen reader for your Windows system but cannot afford to purchase JAWS, NVDA (Non Visual Desktop Access) might be a good option for you. This free screen reader has most of the functionality of JAWS, although on the downside its voice output sounds more “computerish” than the voices available with JAWS. Check out [NVDA documentation](#) and user guides to learn more about this screen reader.

[ChromeVox Add-on for the Chrome Web Browser](#)

If you occasionally test screen readers and prefer an easy-to-use screen reader, consider using ChromeVox. Its use requires the Chrome web browser, and it only works as a screen reader when viewing web content. Unlike JAWS and NVDA, ChromeVox does *not* work as a screen reader for a full operating system, but it does run on any system (unlike JAWS and NVDA that only use Windows), and just has a two-page list of [keyboard shortcuts](#) to operate it, and of these only a handful of commands are frequently used. For details on installing and configuring the screen reader, see [Accessibility in e-Learning: ChromeVox Screen Reader Setup](#).

Web Accessibility Courses (Free)

[Understanding Web Accessibility \(OCAD University\)](#)

If you want to expand your knowledge of web accessibility, and WCAG guidelines and strategies for creating accessible web content, you might consider taking OCAD University’s Understanding Web Accessibility (UWA) course. It is a self-guided online course, which upon request can be moderated by its author for groups seeking a certificate (fees would then apply). The course describes the WCAG guidelines in practical terms for a non-technical audience, and provides examples and strategies used by developers to create accessible web applications and content. It also presents simulations of possible barriers to help readers better understand the barriers that some learners face.

[Accessibility of e-Learning \(Open University\)](#)

The Open University in U.K. offers a course on web accessibility with a focus on e-learning, disabilities and barriers that people face, as well as the variety of assistive technologies. Available for credit for those registered with the Open University, this course is a good compliment to the UWA course, providing on greater emphasis on the human aspects of accessibility and disability.

[Introduction to Web Accessibility \(Google\)](#)

Another short course on web accessibility is offered by Google. It includes information about using Chrome and ChromeVox, as well as the Accessibility Developer Toolbar. It also provides details on WAI-ARIA, and ARIA live regions and how they are implemented, as well as details on various HTML5 accessibility enhancements that can be used to organize web content in meaningful ways that will make sense when accessing web content with assistive technologies.

[Accessibility in e-Learning YouTube Channel](#)

The Accessibility in e-Learning YouTube channel is a compilation of YouTube videos on various topics associated with e-learning accessibility, including videos on how learners with disabilities use the web, developing accessible e-learning content using HTML5, and the accessibility features of various e-learning authoring tools.

Quick Manual Accessibility Tests

1. **Select All:** A quick test to see what content on web pages will be accessible to assistive technologies is to use the keyboard to select all of the page's content. Any elements that are not selected will likely not be accessible to everyone.

Key Combination to Select All:

Windows: Ctrl + A

Mac: Command + A

Linux: Ctrl + A

2. **Tab Navigation:** For another quick test of keyboard accessibility, position the cursor in the your browser's location bar, and then press the Tab key repeatedly until the cursor enters the content of the browser window. You should be able to see the cursor's location, or focus, in the web content as you press the Tab key. All functional elements such as links, form input fields, and buttons should receive focus. If you are unable to follow the cursor, imagine the difficulty in navigation that users with poor vision would experience. If any elements do not receive focus, they will likely be inaccessible to assistive technology users.
3. **Mouse Over Images:** If you are using **Internet Explorer** to test web content, hold a mouse pointer over each of the images on a web page to see if its text alternative appears in a pop up. If no text appears, a text alternative is likely missing. If the text appears, determine whether it accurately describes the meaning in the image; or if the image is a button or link, whether it accurately describe the function or destination.
4. **Turn Images Off:** If you do not have Internet Explorer for the manual test previously noted, you can disable images in your browser settings to see what appears when images are removed. You should be able to understand the content of a web page and operate the links, forms, and buttons equally well with and without images enabled. For more information, see **Accessibility in e-Learning: Turning Images Off in Web Browsers**.

Creating Accessible e-Learning Content

For those seeking greater detail on how web content is made accessible, see **Accessibility in e-Learning: Accessible Content Authoring Practices**. The document provides the accessibility basics of HTML, describing how HTML is authored to be accessible.