Western

Guidebook:

AODA

Making Western’s Accessibile Web

Communications and Public Affairs

Table of Contents

Contents

- What is Accessibility?
  - What is AODA?
  - What does it mean for Western?
  - Does my website/web program need to comply with AODA?
- Quick Reference
  - Provide alternative text
  - Content is well structured and clearly written
  - Help users navigate to relevant content
  - Provide headers for data tables
  - Do not rely on color alone to convey meaning
  - Ensure users can complete and submit all forms
  - Ensure links make sense out of context
  - Caption and/or provide transcripts for media
  - Ensure accessibility of non-HTML content
  - Miscellaneous
- Comprehensive Reference / Checklist (WCAG)
  - Guideline 1. Perceivable
    - 1.1 Text alternatives
    - 1.2 Time-based media
    - 1.3 Adaptable
    - 1.4 Distinguishable
  - Guideline 2. Operable
    - 2.1 Keyboard accessible
    - 2.2 Enough time
    - 2.3 Seizures
    - 2.4 Navigable
  - Guideline 3. Understandable
    - 3.1 Readable
    - 3.2 Predictable
    - 3.3 Recovery from errors
  - Guideline 4. Robust
    - 4.1. Compatible
- Checkers and Tools
  - Checkers and Tools
  - Resources

About this Guidebook

This book is considered a living document, which means it is subject to change, edits, and updates. Please refer back to it often to see if any updates have been done to it.

If you have any comments, questions, or concerns about the contents of this document, you can email web@uwo.ca.

What is Accessibility?

Web accessibility is the inclusive practice of making websites usable by people of all abilities and disabilities. When sites are correctly designed, developed and edited, all users can have equal access to information and functionality.

What is AODA?

The Accessibility for Ontarians with Disabilities Act (AODA) establishes accessibility standards for providing service to people with disabilities in five key areas of daily living: customer service, information and communications, employment, transportation, and built environment. The goal is to make Ontario accessible by 2025. The Act requires that designated public sector and large organizations meet the established accessibility standards for their websites.

What does it mean for Western?

Western is fully committed to conform to the AODA standards according to the following schedule:

- **By January 1, 2014** new Internet websites and web content on those sites must conform with WCAG 2.0 Level A. A website is considered new if:
    - a new domain name was acquired;
    - a new website was built after January 2012; or
    - an existing website underwent a significant refresh (changes to more than 50 percent of design, content, or technology)
- **By January 1, 2021** - all Internet websites and web content must conform with WCAG 2.0 Level AA (with two exceptions relating to captions and audio descriptions for video).

Content published on a website before January 1, 2012 is not required to be compliant with WCAG 2.0. However, people with disabilities can still request information to be provided in an accessible format under the Section 12, "Accessible Formats and Communication Supports."

Web content, including documents such as Word and PDFs, posted after January 1, 2012 will need to be accessible as per the regulation.

- Create accessible .pdf
- Create accessible .doc

Does my website / web program need to comply with AODA?

AODA requirements apply to all websites and web content, including web-based applications that an organization directly controls or manages through a contractual relationship that allows for modification of the product.

Quick Reference

Provide appropriate alternative text

- Every non-text element needs a text alternative (alt text) that provides an equivalent to the image content.
- Alt text should present the content and function, not necessarily a description, of an image.
- If an image has no relevant content or function, is decorative, or the alternative text is provided in nearby text, then the image should have an empty alternative text value (`alt=""`).
- If an image is a link (or hotspot), the alt text must describe the link's function.
- Avoid words like "picture of," "image of," or "link to."
- Use the fewest number of words necessary.
- Implementation Techniques: webaim.org/techniques/alttext/

Content is well structured and clearly written

- Use the simplest language appropriate for your content.
- Organize your content using true headings (e.g., `<h1>`) and lists.
- Use empty (white) space to improve readability.
- Use illustrations, icons, etc. to supplement text.
- Check spelling, grammar, and readability.

Help users navigate to relevant content

- Provide a link that allows the user to skip over navigation to the main content in the page.
- Use true headings to organize content.

Provide headers for data tables

- Identify all data table headers using the `<th>` element.
- Provide an appropriate scope attribute: `<th scope="col">` for column headers or `<th scope="row">` for row headers.
- If appropriate, add a table `<caption>` for the data table.

Implementation Techniques: webaim.org/techniques/tables/

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. Be especially cautious of red/green color combinations.
- Make sure that color contrast is strong, especially between text and background.
- Implementation Techniques: webaim.org/articles/visual/colorblind/

Ensure users can complete and submit all forms

- Put form labels adjacent to or near their controls, so the labels are associated visually.
- Use the `<label>` element to associate labels and controls.
- Group similar elements (such as checkboxes or radio buttons) together using `<fieldset>`.
- Clearly identify required form elements. Don’t make a field required if it is not necessary. Ensure all directions and cues are readily accessible.
- If there are errors in a form that has been submitted, alert the user in an accessible way (especially to a screen reader user) and make it easy to fix the incorrect information and resubmit the form.
- Implementation Techniques: webaim.org/techniques/forms/

Ensure links make sense out of context

- Avoid phrases like “Click here”, “Here”, “More”, “More information”, “Read more”, and “Continue.”
- URLs as link text should usually be avoided, unless the URL is relevant content.

Caption and/or provide transcripts for media

- Videos and live audio must have captions and a transcript. A transcript is sufficient for archived audio.
- Captions should be synchronized, equivalent, and accessible.
- Implementation Techniques: webaim.org/techniques/captions/

Ensure accessibility of non-HTML content

- HTML content will almost always be more accessible than content in any other format.
- PDF, Microsoft Word and PowerPoint files, OpenOffice.org, and Adobe Flash provide basic accessibility features.
- Provide accessible alternatives when non-HTML content cannot be made fully accessible.
- Test the accessibility of non-HTML content in assistive technologies.

Miscellaneous

- Ensure that the page is readable and usable when fonts are enlarged 150-200%.
- Provide a descriptive page `<title>`.
- When using scripting, ensure events are available with both mouse and keyboard.

Make all scripted content and page updates/changes available to screen readers.
- Limit pop-up windows and notify users when pop-ups are used.
- Provide a descriptive title for all frames (e.g., `<frame title=”navigation”>`).
- Follow HTML and CSS coding standards.

Comprehensive Reference / Checklist (WCAG)

To bring your website to Level A compliance, consider making changes based on the following guidelines:

Guideline 1. Perceivable - web content is made available to the senses – sight, hearing, and/or touch.

1.1 Text alternatives: provide text alternatives for any non-text content.

1.1.1 Non-text content

Provide appropriate, equivalent alternative text for non-text elements. All images (including form image buttons and image maps) must have appropriate alt text.

What you can do:
- If your links are buttons, do not use tags that begin: “A link to...”
- When determining the right alt tags to use, think, “What would I put here if I could not use this image?” In general, images that get alt tags are those that provide context to content, inform about an action, or are the subject of the content itself. Some examples of good alt tag implementation are:
  - A photo of Dr. Chakma: `alt=”Dr. Chakma”`
  - An image with Dr. Chakma’s name captioned below it can use a null attribute to avoid redundancy: `alt=””`
  - An image of Dr. Chakma that links to his homepage: `alt=”Dr. Chakma”`
  - Images that do not contain content and are decorative can have null alt tags: alt=””
  - If the image is functional (like a banner or navigation button): `alt=”United Way fundraiser”`
  - If the image is decorative and serves no function rather than as a design element (ie: a graphical line break): `alt=””`
  - If the image supplements a block of text but adds nothing to the content (ie: a tulip on campus in an article about mental health): `alt=””`
  - If you’ve replaced standard form input buttons with custom graphics, the alt tag should inform what the button will do when clicked, for example: `alt=”Submit search”`
  - If using image maps, the main image itself must have an alt tag. The tag doesn’t have to convey all of the regions of the image, just the idea that it’s a container of content, like so: `alt=”An aerial map of Western”`
  - When using a logo on your page, providing text that signifies your name is sufficient: `alt=”Western University”`
- Write alt tags and long descriptions for complex images on a separate linked page
- Add descriptive labels to form elements: `<label for=”name”>Name</label>`

- Add text labels to form inputs: `<input id=”name” type=”text” name=”textfield”>`
- Identify embedded multimedia with accessible text
- Title any frames and iframes you use appropriately: `<frames src=”x.html” title=”Navigation menu”>`

1.2 Time-based media: provide alternatives for time-based media. Note: per the AODA legislation, this is not a requirement now but eventually will be. It is a good idea to get into the habit of doing this now.

1.2.1 Pre-recorded audio-only and video only
Descriptive text transcripts are provided for live, web-based audio (podcasts, MP3 files, etc.) and for non-live, web-based video.

What you can do:

- Provide a detailed description of the content the user is about watch or listen to. Include speaker info, discussion topics, and start and end times
- How to set this up for YouTube: https://support.google.com/youtube/answer/2734796?hl=en

1.2.2 Captions (prerecorded)
Synchronized captions are provided for non-live, web-based video (YouTube videos, etc.).

What you can do:

- Once recorded, spend some time captioning your video.
- Provide a detailed description of the video and post it along with the file in an easy-to-find and read place

1.2.3 Audio description or media alternative (pre-recorded)
A descriptive text transcript OR audio description audio track is provided for non-live, web-based video.

What you can do:

- Provide a detailed description of the video and post it along with the file in an easy-to-find place.

1.3 Adaptable: create content that can be presented in different ways without losing information and structure.

1.3.1 Information and relationships
Semantic markup is properly used to designate headings, special text. Tables should be used for tabular data only, not for design or layout. Lastly, form-input elements should have text labels associated with them.

What you can do:

- Structure page content using proper tags `<h1>` followed by `<h2>`, etc. and paragraphs surrounded by `<p>` tags;
- Use Cascade’s built-in editor tools when marking bolded or italicized text. If text is copied from Word, view source and change any `<b>` tags to `<strong>` and `<i>` to `<em>` to make it more accessible;
- Add captions to your data tables to explain what they are:

```
<table>
```

```
<caption>2016 Department Expenses</caption>
```

- Use table headers to structure tables:

```
<table>
<caption>2016 Expense Reports</caption>
<tr>
<th>Faculty Member</th>
</tr>
```

- Include summaries in all of your tables:

```
<table summary=”A table that details expenses paid for by
the purchasing group over the past 12 months.”>
<caption>2016 Expense Reports</caption>
<tr>
<th>Faculty Member</th>
</tr>
```

- Avoid complex tables with cells that span different rows and columns. Some screen readers still have difficulty parsing them.

1.3.2 Meaningful sequence
The reading and navigation order (determined by code order) is logical and intuitive.

What you can do:

- Spend time to organize the content of your page into a logical flow and design

1.3.3 Sensory characteristics
Instructions cannot rely on colour, shape, size or visual location or sound to illustrate concepts to the user

What you can do:

- Do not use instructions similar to the following:
- Click the red button
- See the link above
- Click the button at the right
- (AUDIO) “Click the next button when you’re finished reading”

1.4 Distinguishable: make it easier for users to see and hear content. Separate foreground from background.

1.4.1 Use of colour
Colour cannot be used to convey content or distinguishing visual elements. Similarly, do not use it to distinguish links from surrounding text unless the contrast between the link colour and the text colour is at differentiated by at least 3:1.

What you can do:

- Use tools like this Contrast Checker to ensure your colours are suitable: http://webaim.org/resources/contrastchecker/
- Avoid using links that are the same colour as your text to illustrate their function
- Use alternative methods of function indication on images rather than colour like strong visual metaphors or consider using text instead

1.4.2 Audio control
Make sure a mechanism is in place to stop, pause, and play music or adjust volume for audio that automatically plays for more than three seconds.

What you can do:

- Do not embed audio or video that auto plays in your page
- If you must have auto playing content, only use players that have easy to identify controls to stop, pause, and play video (like YouTube or Quicktime).

Guideline 2. Operable - Interface forms, controls, and navigation are operable.

2.1 Keyboard accessible: make all functionality available from a keyboard.

2.1.1 Keyboard
All of the page's functionality is available using a keyboard (unless functionality does not exist on a keyboard).

What you can do:

- Do not create keyboard shortcuts that interfere with browser and screen reader commands.

2.1.2 No Keyboard Trap
Keyboard focus is not locked on any one element; the user can skip between all elements on the page using their keyboard.

2.2 Enough time: provide users with enough time to read and use content/ instructions.

2.2.1 Timing Adjustable
If a page or application has a time limit, the user is given options to turn off or alter that time limit.

What you can do:

- If you have a redirect page in place with content on the page, adjust the script to allow users to read the content before the page changes.

2.2.2 Pause, Stop, Hide
Content that automatically blinks, scrolls, or moves for more than five seconds must be pauseable, stopped or hidden by the user, including news tickers, notification alerts and pages that automatically update and refresh (a live blog, for example).

What you can do:

- Before you put automatically-updating tools on your website, first ensure there are Play, Stop and Pause controls attached to it or omit them.

2.3 Seizures: do not design content that could cause seizures.

2.3.1 Three flashes or below
No page content flashes for more than three times per second unless that element contains no red or the flashes are small or of low contrast.

What you can do:

- Do not use blinking content on your site. Instead, rely on well-written and well-designed content to draw attention to things.

2.4 Navigable: provide ways to help users navigate, find content and determine where they are.

2.4.1 Bypass Blocks
A link to skip navigation and other page elements are repeated across web pages.

What you can do:
• Use breadcrumbs to provide links through a page
• Provide anchor links to pages with lengthy content

2.4.2 Page Titles
All pages should have a descriptive and informative page title.

What you can do:
• Consider the content that will be on the page when writing your page title
• Your title should describe the topic or purpose of the page

2.4.3 Focus Order
The navigation order of links, form elements, etc. is logical and intuitive.

What you can do:
• Plan your website, content, and links before committing them to the page;
• Organize links alphabetically or break long lists into shorter, easier to consume lists.

2.4.4 Link Purpose
The purpose of the link can be determined from the link alone and if links are close to one another their target is identifiable from within the text.

What you can do:
• Do not use words like "click here" to indicate action;
• If you have a bunch of links on the page, test the language you use in your links by putting them next to each other and see if you can tell the difference between the links via your indicator text.

Guideline 3. Understandable - content and interface are understandable.

3.1 Readable: make text content readable and understandable.

3.1.1 Page Language
The page's language is identified using the HTML lang attribute.

What you can do:
• Use html declaration `<html lang="en">`

3.2 Predictable: make web pages appear and operate in predictable ways.

3.2.1 On Focus

When a page element receives focus, it does not initiate a change of context
(e.g. rolling over a link should not open a new window or load new page)

What you can do:
- Activate things in ways that are predictable for users. For example, a page pops up a new window only when the user clicks on a button rather than using hover state to pop up a new window
- A "submit" button is used to move on to the next screen rather than having the next screen appear automatically

3.2.1 On Input
User-initiated changes do not automatically cause a change of context unless the user has been advised of the behaviour before using the component
(e.g. entering information in a form does not load a new form field or submits a form)

What you can do:
- Ensure that data or a form control has predictable effects. For example, provide "submit" button for all forms (.html, .pdf, flash, etc)
- Describe what will happen when a change to a form control causes a change of context

3.3 Recovery from errors: help users avoid and correct mistakes.

3.3.1 Error Identification
If an input error is detected, the item that is in error is identified and the error is described to the user in text. Users are allowed to easily fix the error and resubmit the form.

What you can do:
- For forms with mandatory fields
- Indicate required fields (in all types of documents)
- Provide text descriptions to identify required fields that were not completed

3.3.2 Labels or Instructions
Labels or instructions are provided when content requires user input.

What you can do:
- Provide expected data format and example
- Provide text instructions at the beginning of a form or set of fields that describes the necessary input
- Provide text descriptions to identify required fields that were not completed

Guideline 4. Robust- content can be used reliably by a wide variety of user agents, including assistive technologies.

4.1. Compatible: maximize compatibility with current and future user agents, including assistive technologies

4.1.1 Parsing
Significant HTML/XHTML validation/parsing errors are avoided.

What you can do:

- Validate the website at http://validator.w3.org/

4.1.2 Name, Role, Value
Markup is used in a way that facilitates accessibility.

What you can do:
- Following the HTML/XHTML specifications and use forms, form labels, frame titles, etc. appropriately.

This success criterion is primarily for Web authors who develop or script their own user interface components. For example, standard HTML controls already meet this success criterion when used according to specification.

Checkers and Tools

Checkers and Tools

- A-Checker - Web Accessibility Checker from the University of Toronto.
- Wave WebAim - Web Accessibility Checker.
- WAVE Toolbar - Firefox extension checks for accessibility issues.
- PDF-Accessibility-Checker (PAC) - Test the accessibility of PDF files.
- Fangs Screen Reader Emulator - Firefox extension screen reader simulator generates text similar to what would be read by a screen reader, allowing a sighted user to better understand the experience that a screen reader user would have.
- NonVisual Desktop Access (NVDA) - An open-source screen reader for Windows. Widely used in the visually impaired community. Excellent for testing.
- WCAG Contrast checker - Firefox extension checks colour contrast.
- WebAIM contrast Checker
- Juicy Studio Luminosity Colour Contrast Ratio Analyser
- Snook.ca Colour Contrast Check Tool
- CheckMyColours: tests colour for contrast ratios.
- RGB to HEX colour code converter
- Microsoft readability test
- Web Developer Toolbar - Firefox extension with a wide range of tools.
- Holmes - A CSS stylesheet which adds warning markers to pages to indicate HTML problems.
- CleanCSS - CSS minifier that reduces file size. Based on CSSTidy.

Resources

- W3C Guidleines and Checklist (Elsevier)
- Ontario.ca/accessibility
- Council of Ontario Universities Toolkit (Section 14)
- Accessibility for Ontarians with Disabilities Act (AODA)
- W3C Accessibility
- Web Content Accessibility Guidelines 2.0
- Accessible Digital Documents and Websites
- Creating accessible .pdf documents
- Creating accessible .doc files
- Creating accessible .ppt (PowerPoint) files

- Creating accessible .xsl (Excel) files
- Accessibility for Web writers from 4Syllables
- Why Your Links Should Never Say “Click Here”

I’ts here, trust me. Select all!

Western

Guidebook:

AODA

Making Western’s Accessible Web

Communications and Public Affairs