# Accessibility Best Practices

# MOECC June 2015

By Anne Jackson and Shell Ravas

## Best Practices for accessibility tagging

Contents

[Accessibility Best Practices 0](#_Toc425516670)

[MOECC June 2015 0](#_Toc425516671)

[Best Practices for accessibility tagging 1](#_Toc425516672)

[Alt Tags 3](#_Toc425516673)

[Links 4](#_Toc425516674)

[Citations 5](#_Toc425516675)

[Glossary 5](#_Toc425516676)

[Language changes 5](#_Toc425516677)

[Subscripts and superscripts 5](#_Toc425516678)

[Checkboxes 5](#_Toc425516679)

[Lists 5](#_Toc425516680)

[Math functions 6](#_Toc425516681)

[Title tags 6](#_Toc425516682)

[Iframe 6](#_Toc425516683)

[Fonts 6](#_Toc425516684)

[Heading structure 6](#_Toc425516685)

[Colour contrasts 7](#_Toc425516686)

[Navigation 7](#_Toc425516687)

[Page navigation 7](#_Toc425516688)

[Menu navigation 7](#_Toc425516689)

[Tabbing 7](#_Toc425516690)

[Graphics, labels, lines and buttons 7](#_Toc425516691)

[ARIA (Accessible Rich Internet Applications) 7](#_Toc425516692)

[Common elements: 7](#_Toc425516693)

[Video 8](#_Toc425516694)

[Keyboard behavior 8](#_Toc425516695)

[Forms and form sets 9](#_Toc425516696)

[Forms 9](#_Toc425516697)

[Grouping Controls: 9](#_Toc425516698)

[Labels 9](#_Toc425516699)

[Text Areas 9](#_Toc425516700)

[Check boxes (or radio buttons) 9](#_Toc425516701)

[Buttons and image buttons 10](#_Toc425516702)

[Validation and error recovery 10](#_Toc425516703)

[Form Instructions 10](#_Toc425516704)

[Tables 11](#_Toc425516705)

[Layout tables 11](#_Toc425516706)

[Data tables 12](#_Toc425516707)

[Example Data Table 12](#_Toc425516708)

[Best degradation - table or chart with download 13](#_Toc425516709)

[Advanced Example Data Table 13](#_Toc425516710)

[Footnotes 15](#_Toc425516711)

[Page summary 16](#_Toc425516712)

[Sample infographic, full-width and alt text. 16](#_Toc425516713)

[Images 16](#_Toc425516714)

[Callout boxes format 17](#_Toc425516715)

[Narrow Aligned Right 17](#_Toc425516716)

[Narrow Aligned Left 17](#_Toc425516717)

[Full width 17](#_Toc425516718)

[Line Breaks 17](#_Toc425516719)

[Heading or paragraph indents 17](#_Toc425516720)

[French codes 18](#_Toc425516721)

[Pdf Techniques and ARIA 18](#_Toc425516722)

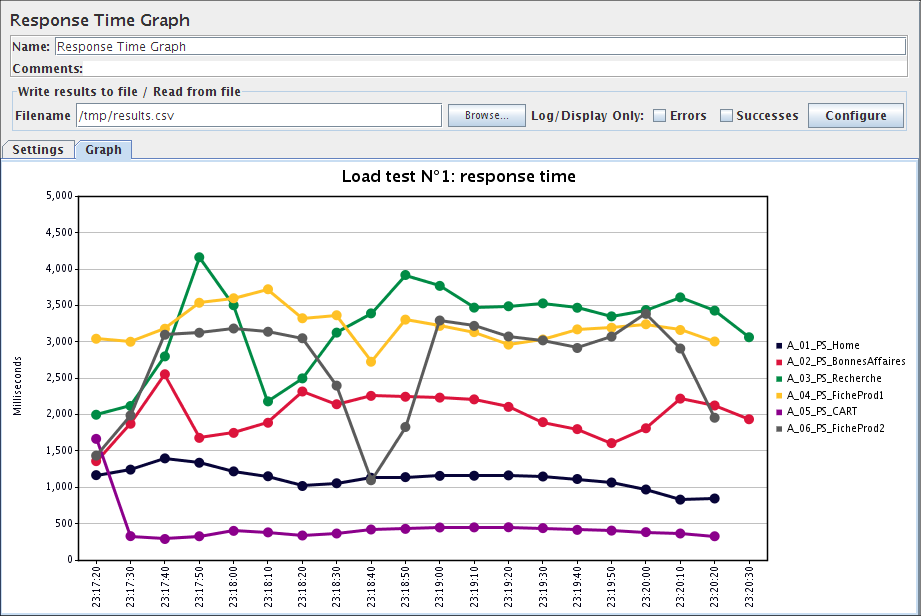
[Disability considerations: 18](#_Toc425516723)

[Assistive Technology software solutions: 19](#_Toc425516724)

[Testing Approach 19](#_Toc425516725)

### Alt Tags

* Alt text is accessed by screen reader users to provide them with a text equivalent of images. In visual browsers, the Alt text is displayed when an image is broken, or when all images have been disabled. Using ALT tags is also beneficial to users on low-bandwidth connections, where images are slow to download.
* Math formula images should say “Math formula for zero emissions”
* The ALT tag adds a text description to an image on a Web page, and should be used for all images, graphical bullets, and graphical important horizontal rules. The Alt text within the ALT tag should let the user know what an image’s content and purpose are. It is held within the image tag and should be a description of the image as if you are describing the image to the low vision user.
* Length should be no more than 125 characters Although there is no set rule, but currently Jaws will read only the first 125 characters.
* Links **- Images** used as links should have Alternative (or "Alt") text describing the destination of the link, not the image itself, so include “link to…” or “home page of…”
* Null Alt tags - Images used as spacers or in lines in toolbars should have an empty ALT tag (i.e. <Alt="" >). Screen readers will simply skip over images with empty ALT tags.
* Image sizes:
  + **landscape photo** 220px wide x 120-160px tall
  + **portrait photo** 160px wide x 207-220px tall
* Alt text on image bullets:
  + <IMG SRC="blueball.gif" ALT="Item is a "> Pencil <BR /> - this is OK
  + <IMG SRC="blueball.gif" ALT="\*"> Telephone <BR /> this is acceptable
  + <IMG SRC="blueball.gif" ALT=""> Calendar <BR /> this is ok and will just list the items as the Alt is null so wont be read by a screen reader
* Long descriptions can be handled as an overt link under the image like [**Extended Text Description Available**](#null). This can be useful for infographics or complex graphs with a method of getting back to the original page.



**Descriptive, long description link here**

### Links

* Do not use **target=”\_blank”** and if you do then let the user know this is happening for example: **Visit The Wonderful World of fish running (Please note: this link will open the page in a new browser window.)**
* Use the **Alt tag** to let the screen reader know the image link is maybe a link but not using click here etc. An example would be “our products list” and the screen reader would read it as “Link our products list”
* Let the user know what format a download is in for example PDF and any other behaviors. <a href="1040.pdf">Tax form 1041 (PDF)</a>and if it opens in a new window or is a download .
* Should be underlined but controlled by css.
* **Event handlers - onblur, onfocus (expand)**
* Avoid dashes like 1 – 2, 5 – 10
* Use actual sentences rather than the website url so this is good :

<a href="http://www.omafra.gov.on.ca/english/index.html">Agriculture, Food and Rural Affairs</a>

And

Existing Ontario legislation may be found at the <a href=" http://www.e-laws.gov.on.ca"> E-Laws website.</a>

And

<a href=”/docs/879-data-ed.csv>Data download XXX available in csv format.</a>

* The link text is short enough to be quickly read, descriptive enough to know where it will take you.
* Titles and headings are very important. People decide whether or not to click on a search result by looking at a page title. Similarly, people often skim headings on a page and skip to the most useful section. This is a common tactic for screen reader users.
* be concise (4-5 words)
* be descriptive and include the keywords people type into search engines (if you’re not sure what terms people use to find your type of information online, check Google Trends)
* use subject-verb-object or indefinite article where possible (e.g. get a driver’s licence instead of get your driver’s licence)
* if there are sub-clauses, use colons to introduce them (e.g. UFO insurance reform: impact assessment instead of An assessment of the impact of proposed reforms to UFO insurance)
* avoid using “Ontario” in titles or headings – most search engines automatically localize results and the Ontario.ca domain already contains the word
* avoid phrasing headings as questions
* **Level 2 (H2**) headings automatically appear in a Table of Contents sidebar – you can have up to 9 of these on an article page. Headings should make sense when read from the Table of Contents.

### Citations

* <blockquote cite="http://www.w3.org/TR/1999/REC-html401-19991224/ struct/text.html">

### Glossary

<dl>

<dt>Aesthetic</dt>

<dd>aspects of drinking water quality (namely taste, odour, colour and clarity) that are perceivable by the senses.

</dd>

</dl>

### Language changes

Include <span lang="en">English text</span> in French copy where English text exists.

Example:

<span lang="en">Clinical Trials Ontario</span> crée une approche simplifiée …

### Subscripts and superscripts

* H<sub>2</sub>O
* E = mc<sup>2</sup>

### Checkboxes

Use the word “yes” instead of a check mark. If not, an image with the alt text = yes.

### Lists

Nested lists are not advised as they can be used for layout and become confusing when complex. Suggest using <ul> unordered list followed by ordered list <ol> so the information become more readable.

(nb: This is depreciated code and css styling is preferred)

**<ol type="a">**

<li>a</li>

<li>b</li>

</ol>

**<ol type="A">**

<li>A </li>

<li>B</li>

</ol>

**<ol type="i">**

<li>i</li>

<li>ii</li>

</ol>

**<ol type="I">**

<li>I</li>

<li>II</li>

</ol>

### Math functions

Use math ML http://www.washington.edu/doit/are-there-screen-readers-can-read-math-equations

### Title tags

These should generally not be used, but see below for an exception to this rule.

### Iframe

Accessible iframes need a title tag - <iframe title="Map to X" src="https://maps.google.com/..." width="320" height="240" frameborder="0" marginheight="0" marginwidth="0">Loading... </iframe>

### Fonts

12 pt minimum font. Colour contrast as per guidelines all covered by css. Avoid all capitals in a block of text.

Avoid italics

Avoid All capitals and use sentence case, remove whitespace for layout.

### Heading structure

Should have 1 Heading 1 which is probably the same as the <title>

Other headings should follow sequentially and logically. Avoid going from a H2 to H4 but a H4 to H2 is ok.

<h1>Recipes</h1>

<h2>Vegetables</h2>

<h3>Breakfast</h3>

<h4>Tomatoes</h4>

<h4>Mushrooms<h4>

<h3>Lunch</h3>

<h2>Meat</h2>

<h3>Breakfast</h3>

<h4>Bacon</h4>

<h2>Fish</h2>

### Colour contrasts

A good test is to see it in black and white and does it still stand out. Use a colour contrast analyser for criteria on acceptable colour and font size combinations.

### Navigation

#### Page navigation

"Skip to main content" should be provided at the top of each page to enable keyboard only users to skip repetitive navigation or to traverse long pages. Long pages should have anchor links and back to top links for pager navigation

#### Menu navigation

These should all be navigable by keyboard and not have “on-hover” or other specific mouse only controls [see event handling controls.](http://www.w3.org/WAI/GL/wiki/Making_actions_keyboard_accessible_by_using_keyboard_event_handlers_with_WAI-ARIA_controls)

#### Tabbing

Default order of navigation tabbing is as interpreted code.

### Graphics, labels, lines and buttons

* Label – ARIA-label If the label text is visible on screen, authors SHOULD use ARIA-labelledby and SHOULD NOT use ARIA-label.
* When a button is used for a link, put in the Alt text
* <button ARIA-label="Close" onclick="myDialog.close()">X</button>To a screen reader the X may not give out the information that the X is to close the open object.
* When a line is used for decoration only like a text bar line separator, use a null Alt tag (Alt=””)

### ARIA (Accessible Rich Internet Applications)

[Wai-ARIA](http://www.w3.org/TR/wai-aria/usage) should be used, for things like form elements that are required, on focus behaviours. ARIA will be dominant when used. Currently HTML mechanism exists to:

* Identify the role of the DIV as a pop-up menu
* Alert assistive technology when these elements have focus
* Convey accessibility property information, such as whether the pop-up menu is collapsed or expanded
* Define what actions can be formed on the element other than through a device-dependent means through the event handler type (onmouseover, onclick, etc.)

#### Common elements:

* **Properties and states**
  + <input ARIA-required="true">. This property will cause a screen reader to read this input as being required (meaning the user must complete it)
  + <input ARIA-disabled="true">. This property will cause a screen reader to read this input as being disabled, but this state value could easily be changed to false dynamically based on user input.
  + ARIA-haspopup="true"
  + Required field - ARIA-required = true or false
  + Label – ARIA-label If the label text is visible on screen, authors SHOULD use ARIA-labelledby and SHOULD NOT use ARIA-label
* **Roles** for body content, navigation, elements to make keyboard navigation possible etc. A full list of what we can use should be decided upon.
  + role =”banner”
  + role=”main”
  + role=”form”
  + role=”search”
  + role=”alert”
  + role=”link”
  + role=”menuitem”
  + role=”navigation”
* **Managing Focus** for tabbing, visual cues and keyboard shortcuts using container roles.
* Layout tables should use role=”presentation”

A full list of what we can use should be decided upon. There are some good references from [Firefox on ARIA implementation](https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Techniques) , the [Webaim website](http://webaim.org/techniques/aria/) and [WC3 WAI ARIA](http://www.w3.org/TR/wai-aria/states_and_properties#aria-controls) and [ARIA techniques.](http://www.w3.org/TR/WCAG20-TECHS/aria)

### Video

* Must not autoplay and can start, pause and stop.
* Must have captioning
* Nice to have a text script
* Keyboard compliant

### Keyboard behavior

Navigation should be in logical order as the keyboard will follow this order, tabbed 1 first, tabbed 2 second etc.

<ul>

Tabbed 1<li><a href="/services/">Services</a></li>

Tabbed 2<li class="current"><a href="/articles/">Articles</a></li>

Tabbed 3<li><a href="/resources/">Resources</a></li>

Tabbed 4<li><a href="/community/">Community</a></li>

</ul>

### Forms and form sets

#### Forms

##### Grouping Controls:

Use the <fieldset> and <legend> elements to group and associate related form controls. For formsets, use the <fieldset> and <legend> elements to group and associate related form controls.

<fieldset>

<legend>Enter delivery information here:</legend>

…..field elements here….

</fieldset>

A fieldset grouping control groups a set of related information. Groupings of form controls, typically groups of related checkboxes and radio buttons, sometimes require a higher level description (such as "Shipping Method" for a group of shipping option radio buttons). This descriptive text can be associated to the group of form controls using <fieldset> and <legend>. The <fieldset> identifies the entire grouping and <legend> identifies the grouping's descriptive text. Using <fieldset> and <legend> ensures that the text description is read to screen reader users when the grouping is navigated to.

##### Labels

<label for="name">Name:</label>

<input id="name" type="text" name="textfield">

Matching the for and id values associate the two. The id MUST be unique on each page and only 1 form element to each unique form element.

**Multiple labels** for simple input form like search use ARIA-labelledby, ARIA-describedby, and ARIA-label.

Hidden labels - In general, content should only be hidden from sighted users and made available to screen reader users when content is apparent visually, but not apparent to screen reader users.

* Use <label> element when you can. It has excellent browser and screen reader support, and users can click on the label to select the associated form control.
* Use ARIA-labelledby to overcome the 1:1 limitations of <label>.
* Use ARIA-describedby in addition to a label when you need to associate descriptive text to the form control.

• Use a hidden <label> or title or ARIA-label when a visible text label is not available.

##### Text Areas

<label for="address">Enter your address:</label><br />

<textarea id="address" name="addresstext"></textarea>

##### Check boxes (or radio buttons)

<fieldset>

<legend>Select your delivery options:</legend>

<input id="mail" type="checkbox" name="delivery" value="mail"><label for="mail">Mail</label><br />

<input id="courier" type="checkbox" name="delivery" value="courier"><label for="courier">Courier</label><br />

<input id="pickup" type="checkbox" name="delivery" value="pickup"><label for="pickup">Pickup</label>

</fieldset>

Avoid multiple selects as it can be problematic.

##### Buttons and image buttons

**Button:**

<input type="submit" name="submit" value="Submit Search">

<input type="reset" name="reset" value="Reset">

<button>Activate</button>

**Image button:**

<input type="image" name="submitbutton" Alt="search" role=”button” src="submit.png">

##### Validation and error recovery

If the form requires input validation like “required”, ensure it is located close to the label

<label for="firstname">First Name

<span style="color:red">(required)</span></label><br />

<input type="text" name="firstname" id="firstname" />

Test with keyboard only (Tab, Shift+tab and enter keys.)

* Clear focus indicators (a:focus {style for focus}
* Navigation order (natural is best and only links and form controls are naturally navigatable items)

##### Form Instructions

* Provide **feedback** to users about the results of their form submission, whether successful or not. This includes in-line feedback at or near the form controls, and overall feedback that is typically provided after form submission. Notifications need to be concise and clear. In particular, error messages should be easy to understand and should provide simple instructions on how they can be resolved. Success messages are also important to confirm task completion.
* **Provide instructions** to help users understand how to complete the form and individual form controls. Feedback should be provided using assistive techniques, [please see tutorial on form instructions](http://www.w3.org/WAI/tutorials/forms/notifications/)
* **Validate input** provided by the user, and provide options to undo changes and confirm data entry in case they are unsure (for example “Are you sure you want to order this item?”) It is critical to include form instructions in ways that can be read aloud for validation. This will be explained in [more detail in this tutorial](http://www.w3.org/WAI/tutorials/forms/instructions/).
* Divide **long forms into multiple smaller forms** that constitute a series of logical steps or stages, and inform users about their progress.

### Tables

**Screen readers treat layout tables and data tables very differently**. For layout screen readers read in the order presented but for data tables, they will identify the presence of the table including number of columns and row, provide table navigation functionality, read row and column headers, etc. It is therefore vital that data table markup, such as <caption>, <th>, etc. are NEVER used within layout tables, otherwise the screen reader may incorrectly present the table as a data table causing increased overhead and confusion.

~~Set <table WIDTH="100%">~~

Another way to associate data cells and headers is to use the headers and id attributes. This method is NOT generally recommended because scope is usually sufficient for most tables, even if if the table is complex with multiple levels of headers.

Proportional sizing rather than actual sizing should be used for both table types.

[Reference the WCAG2.0 Data and layout table guidelines](http://www.w3.org/TR/2012/NOTE-WCAG20-TECHS-20120103/H43.html)

#### Layout tables

* Table summaries (<table summary="....") **should not be used on layout tables**.
* Tables can be used for layout, screen readers simply read the content of table based on the source code order.
* **Never use data table markup like<caption>, <th>, or <summary> or id attributes**
* Keep them simple

<table role="presentation">

<tr >

<td>Cell 1</td>

<td>Cell 2</td>

<td>Cell 3</td>

</tr>

<tr >

<td>Cell 4</td>

<td>Cell 5</td>

<td>Cell 6</td>

</tr>

<tr >

<td>Cell 7</td>

<td>Cell 8</td>

<td>Cell 9</td>

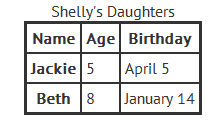
</tr>

</table>

#### Data tables

* **Downloadable data table** is the preferred method. Create a csv file and link to it under the data table. We need to make the exce,l using cut and paste techniques, quickest method isfrom the html preview of the document, and save it as csv.
* **Use a caption** within the table <table> <caption>Shelly's Daughters</caption>, it is generally helpful but not required. Do not use headers in the caption <caption>~~<h2>~~Shelly's Daughters~~<h2/>~~</caption
* Identify Row and Column headers ,using id and headers, add id and headers attributes to table cells so they identify the cells that relate to the headers. This method is complex so scope is preferred AND use the download table data as preference.
* Use scope to set expectation for a heading row and data row, add scope attributes to header cells so they identify the cells that relate to the header
* Note: **SUMMARY Tag for complex data tables**
  + The SUMMARY attribute is placed within the TABLE tag, and is read only by screen readers. It can be used to clarify the organization of a table, or to provide a quick summary of the table’s content. It shouldn’t repeat the information in the CAPTION tag, but can be used to supplement that information.
  + Provide download csv table to avoid excessive formatting.

##### Example Data Table



<table>

<caption>Shelly's Daughters</caption>

<tr>

<th scope="col" id=”name”>**Name**</th>

<th scope="col" id=”age”>**Age**</th>

<th scope="col" id=”bday”>**Birthday**</th>

</tr>

<tr>

<th scope="row" headers=”name” id=”n1”>**Jackie**</th>

<td headers=”age” id=”a1”>**5**</td>

<td headers=”bday” id=”b1”>**April 5**</td>

</tr>

<tr>

<th scope="row" headers=”name” id=”n2”>**Beth**</th>

<td headers =”age” id=”a2”>**8**</td>

<td headers=”bday” id=”b2”>**January 14**</td>

</tr>

</table>

Abbreviations

We can also use abbreviations for things like check list ✓ or tick and p utting the word “Yes” is also acceptable.

<tr>

<th scope="row">File an application for subsystem classification</th>

<td><abbr title="Yes">&#10004;</abbr></td>

</tr>

<li> tags OK within a table

##### Best degradation - table or chart with download

Use for complex tables and is the preferred method as the data is open and can be manipulated by the person with an accommodation how they prefer.

Sample Chart with accompanying enlarge and downloadable data chart

<p align="center">

<img src="/sites/default/files/min-wage.png" alt="Chart: Ontario's General Minimum Wage">

</p>

<p class="rteright">

<a href="/sites/default/files/min-wage-lg.png">Enlarge</a>

|

<a href="/sites/default/files/min-wage.csv">Le téléchargement des données disponibles au format CSV</a>

Data download available in CSV format

Data download of *Table 1* available in CSV format

*Description of document*

</p>

##### Advanced Example Data Table

<table width =”100%” >

<caption>Table 3-7: Piping Identification Background and Legend Colour</caption>

<tr>

<th scope="col" rowspan="2">

<p>Contents classification </p>

</th>

<th scope="col" colspan="2">

<p>Background colour </p>

</th>

<th scope="col" colspan="2">

<p>Legend colour </p>

</th>

</tr>

<tr>

<th scope="col">

<p>Colour </p>

</th>

<th scope="col">

<p>CGSB Equivalent\* </p>

</th>

<th scope="col">

<p>Colour </p>

</th>

<th scope="col">

<p>CGSB Equivalent\* </p>

</th>

</tr>

<tr>

<td>

<p>Hazardous </p>

</td>

<td>

<p>Yellow </p>

</td>

<td>

<p>505-101 </p>

</td>

<td>

<p>Black </p>

</td>

<td>

<p>512-101 </p>

</td>

</tr>

<tr>

<td>

<p>Inherently Low Hazard </p>

</td>

<td>

<p>Green </p>

</td>

<td>

<p>503-107 </p>

</td>

<td>

<p>White </p>

</td>

<td>

<p>513-101 </p>

</td>

</tr>

<tr>

<td>

<p>Fire protection </p>

</td>

<td>

<p>Red </p>

</td>

<td>

<p>509-102 </p>

</td>

<td>

<p>White </p>

</td>

<td>

<p>513-101 </p>

</td>

</tr>

<tr>

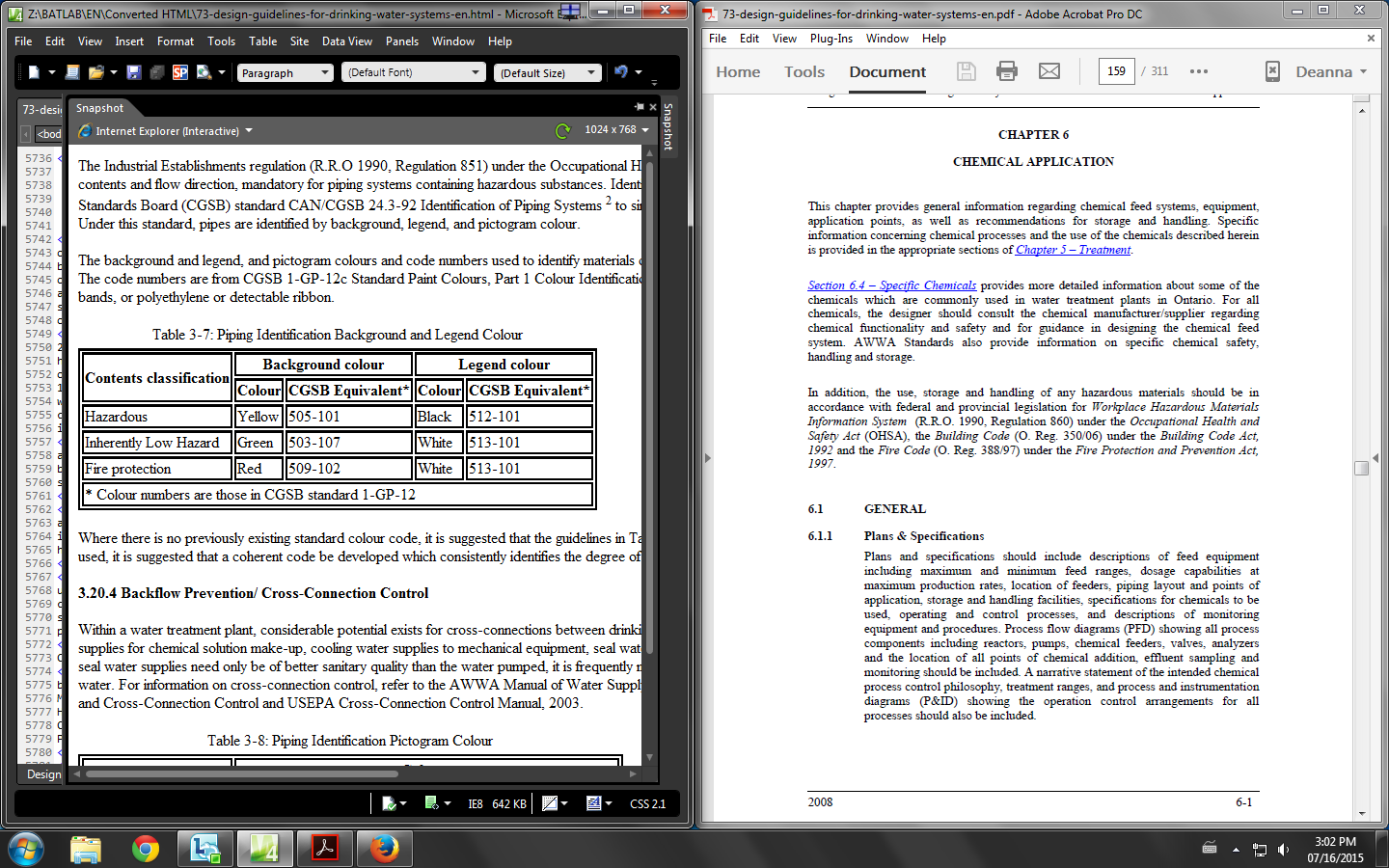
<td colspan="5">

<p>\* Colour numbers are those in CGSB standard 1-GP-12 </p>

</td>

</tr>

</table>



**Data available as an excel download**

Or create the excel table and put it into the documents folder and place a link to it under the table

### Footnotes

**<h2>Title of section</h2>**

<p>

Lorem ipsum dolor sit <a href="#foot\_note1" id="foot\_1">Footnote 1</a>, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet</p>

<p>

<p><a href="#foot\_1" id="foot\_note1">1 Back to the original place</a></p>

</p>

**<h2>Title of next section</h2>**

The ID must start with a letter (a-z or A-Z)

All subsequent characters can be letters, numbers (0-9), hyphens (-), underscores (\_), colons (:), and periods (.).

Each ID must be unique within the document.

### Page summary

Says, “This is an HTML version of Ontario's new poverty reduction strategy. Learn about new and enhanced programs, services and benefits that will reach more vulnerable people and lead more families toward a brighter future.”

Image dimensions are 220px × 150px

### Sample infographic, full-width and alt text.

<p align="center">

<img src="/sites/default/files/youth-jobs-strategy.png" alt="An infographic depicting the amount the government is investing over two years in the Youth Jobs Strategy as $295M, and the number of youth job opportunities created to date as a result of the Youth Jobs Strategy at 20,000.">

</p>

### Images

Sample infographic with Enlarge image

Enlarged images appear to be a width of 949px

<p align="center">

<img src="/sites/default/files/iah.png" alt="An infographic depicting the government’s investments in affordable housing: Over $400 million: Amount being committed over 5 years to extend the Canada-Ontario Investment in Affordable Housing Agreement; Over 10,360: Number of units that are being built or repaired; Over 10,200: Number of low- to moderate-income Ontarians in receipt of rental or down payment assistance; $4 billion: amount committed to affordable housing since 2003">

</p>

<p class="rteright">

<a href="/sites/default/files/iah-lg.png">Enlarge</a>

</p>

Sample image

<p align="center">

<img src="/sites/default/files/disabled2.jpg" alt="">

</p>

Interestingly their photos do not contain ALT tags or have assigned dimensions. The width is set to 100% with actual width of 625px, no standard height. Photos are JPG and charts are PNG. No specific naming convention.

### Callout boxes format

#### Narrow Aligned Right

<div class="fact-box right" style="width: 30%; margin: 5px 0px 10px 10px;">

<p>

</p>

</div>

#### Narrow Aligned Left

<div class="fact-box left" style="width: 30%; margin: 5px 30px 40px 0px;">

<p>

</p>

</div>

#### Full width

<div class="fact-box left" style="width: 95%; margin: 5px 0px 10px 0px;">

<p>

</p>

</div>

### Line Breaks

<br />

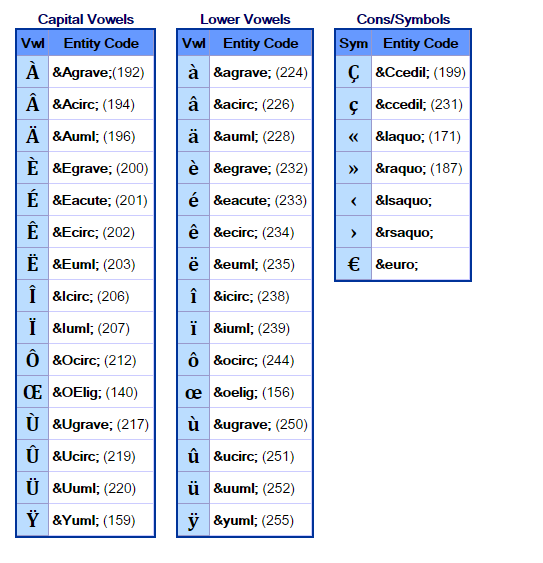
### Heading or paragraph indents

Use class=”rteindent1”

<h4 class="rteindent1"> Building on progress:</h4>

<p class="rteindent1"></p>

### French codes



## [Pdf Techniques and ARIA](http://www.w3.org/WAI/GL/WCAG20-TECHS/pdf.html)

## Disability considerations:

* Low vision and no vision
* Colour blindness
* Dexterity issue like carpal tunnel, tremors, rheumatism
* Learning disabilities
* Dyslexia
* Traumatic injuries
* Hearing imparement

Most assistive technologies for people with motor disabilities either work through the keyboard or emulate the functionality of the keyboard.

Knowing this, developers can focus on making their content accessible to the keyboard, and ensure that the site is navigable with as few keystrokes as possible.

The key for true accessibility is to produce things in more than one format and have navigation available in more than one way, designed with clear direction/focus, usability, graceful degradation and simplistic checks (are you sure?)

### Assistive Technology software solutions:

* Jaws – Speaks the words on a screen, expensive but well known
* NVDA – Free and in practice works in a very similar manner to Jaws
* Zoomtext – magnifies the screen for low vision users. Disadvantage is that you cant see all the screen so prompts need special attention to bring your focus to the ask (for example an “Are you sure?” confirmation button may get missed
* DNS – Dragon Naturally Speaking is used to speak commands into the computer to control input instead of (or to enhance) the keyboard

## Testing Approach

1. Check the source code in achecker.ca
   * 1. Copy source code
     2. Paste into achecker.ca on the paste HTML Markup tab
     3. Select options of WCAG 2.0 (Level AA) and Enable HTML Validator
     4. Check out the Known Problems, Likely Problems Potential problems and HTML problems.
   1. Wave toolbar
      1. Check the Errors, Features and Alerts
   2. WC3 Validator validate by direct input and check the error messages
   3. Total validator
2. Correct and re-check
3. Do a visual check

## Troubleshooting

French or strange characters showing up on conversion:

Make sure language is French, keyboard may have to be French and on Dreamweaver check the settings below. Both of these work.

