Peter Shikli

(503) 570-6831

[info@access2online.com](mailto:info@access2online.com)

Analyst: Alicia B.

Date: 02/15/2018

QA: Juliette M.

Date: 02/17/2018

# **Targeted Accessibility Instruction Sheet**

Client: Avalon School District

Page Responsibility: Contact Us, Make a Suggestion

# Form controls

WCAG Guidelines Covered:

Level A - 1.1.1 Non-Text Content; 1.3.1 Info and Relationships, 1.4.1 Use of Color; 3.2.1 On Focus; 3.2.2 On Input; 3.3.1 Error Identification; 3.3.2 Labels or Instructions;

Level AA - 2.4.6 Headings and Labels; 3.3.3 Error Suggestion; 3.3.4 Error Prevention (Legal, Financial, Data)

## Form Accessibility Guidance Details

Form elements such as text fields, radio buttons, checkboxes and buttons all require a descriptive, accessible name. Additionally, all instructions or cues that are needed to understand or complete the process must be programmatically associated with the appropriate elements so assistive technology will supply the information to users. Different types of form elements use different methods for providing this information.

Buttons need to have a descriptive value which describes their function or purpose. Generic text such as “GO” should be avoided. If a value such as “GO” has been used, a secondary method must be used to provide the information that a sighted user gets from the visual context and placement of the button. Image buttons require a meaningful ALT on the IMG tag to describe the purpose or function of the button.

For example: You have used a search field on the “Make a Suggestion” page. The text input is labeled “Search” and an image button with an eyeglass icon to submit the search.

* Add an ALT on the image button telling users it will submit the search, such as ALT=”Submit Search”.

Other form inputs, such as text boxes, radio buttons and checkboxes, need onscreen text which describes their purpose or function. These text labels must be programmatically associated with the input. This is most commonly achieved with the “LABEL FOR” and “ID” tags and attributes. The onscreen text which describes what is supposed to be entered into the input should be wrapped in ‘LABEL FOR=”abc123”’, and the input should have the ‘ID=”abc123”’ attribute included in the tag.

For example: On the “Contact Us page” there are fields for the user to enter their first and last name. You have not used LABEL FOR and ID to associate the text label with the input field.

* Add LABEL FOR and ID

<LABEL FOR=”fname”>First Name:</LABEL>

<input ID=”fname” TYPE=”text”>

<LABEL FOR=”lname”>Last Name:</LABEL>

<input ID=” lname” TYPE=”radio”>

When there are instructions or explanatory text for sets of inputs, such as a question relating to a set of radio buttons or checkboxes, a fieldset or other mechanism should be used to associate the additional instructions to the individual form inputs.

For example: The “Make a Suggestion” page has a question “What is you gender?” and related radio buttons with the text “Female,” and “Male”. There is no markup to associate the question with the possible responses.

* Add both LABEL FOR/ID on the individual radio buttons as well as a mechanism to associate the question “What is your gender?” with each input, such as fieldset and legend; TITLE; or ARIA. All inputs in a grouping need to have a matching value in the ‘NAME’ attribute

Groups of Inputs Example:

A) Using Fieldset:

<FIELDSET>

<LEGEND> What is your gender?</LEGEND>

<LABEL FOR=”female”>Female</LABEL>

<input NAME=”gender” ID=”female” TYPE=”radio”>

<LABEL FOR=”male”>Male</LABEL>

<input NAME=”gender” ID=”male” TYPE=”radio”>

</FIELDSET>

B) Using the ARIA “describedby” property:

<p ID="question1">What is your gender?</p>

<LABEL FOR="female">Female</LABEL>

<input ARIA-DESCRIBEDBY="question1" NAME="gender" ID="female" TYPE=”radio”>

<LABEL FOR="male">Male</LABEL>

<input ARIA-DESCRIBEDBY="question1" NAME ="gender" ID="male" TYPE=”radio”>

In situations where onscreen text describing the purpose of an input is not desired, such as a text field for a search function placed next to a button with the text “Search,” the input still must be named for assistive technology. A TITLE attribute in the INPUT tag can be used in these situations. Additional methods like using hidden onscreen text or ARIA to describe inputs are acceptable as well.

When an input requires the information be entered in a specific format, those instructions must be provided for users. This can be achieved through onscreen text, a TITLE attribute, or some other method. Similarly, if the field is required for submission of the form, this must be indicated in an accessible way. The color of the text label cannot be the only method of indicating that the field is required. Text in the label, such as “(Required)” or including the attribute REQUIRED=”true” in the input tag is acceptable

If there are fields which require input, or input in a specific format, before a form will be processed the user must be informed if they have made an error in completing these fields. Highlighting the fields which need to be altered in a specific color is not an acceptable method for indicating erroneous fields. The fields must be identified in text with a description of what is required. For example, if a field requiring the date to be entered in mm/dd/yyyy format is incorrectly filled in, an appropriate error suggestion would involve informing the user of the correct format and returning the focus to the date field.

Any form submissions that are for legal commitments, financial transactions, or data storage purposes must allow the user to check, confirm, or reverse the submission. In practice this means that data entered by the user is checked for input errors and the user is provided a chance to correct them; OR there is a method to review, confirm, and correct information before finalizing the submission; OR that the submission can be reversed upon request.

### Form Accessibility Checklist

Form Checklist:

* Buttons have a meaningful descriptive text value. Image buttons have an ALT attribute which describes the function of the button.
* All inputs are appropriately labeled, ideally with onscreen text.
* All instructions or cues are provided in text which is programmatically associated with the input being described.
* When a field requires input this information is not provided by color alone.
* When a field requires input in a specific format this information is provided in text.
* If a field is required and has been left blank or a field has been completed incorrectly, a text description of the error is provided and the field is indicated through non-color means.
* Submissions for legal commitments, financial transactions, and data storage can be checked, confirmed, or reversed.

# Text

WCAG Guidelines Covered:

Level A - 1.1.1 Non-Text Content; 1.3.1 Info and Relationships; 1.3.3 Sensory Characteristics; 1.4.1 Use of Color; Level AA - 1.4.3 Contrast (Minimum); 2.4.6 Headings and Labels; 3.1.2 Language of Parts

## Text Accessibility Guidance Details

When providing text content on webpages it is important to include appropriate markup so that assistive technologies are able to report the information and relationships relayed by the visual presentation of the text. Specifically, heading structure is essential for unsighted users to understand the information hierarchy and easily navigate to different sections of the content. If text has been visually styled as a heading, it must have the accompanying markup. HTML heading markup is provide for levels 1 through 6. Styling can be applied to programmatic heading levels to achieve desired visual formatting. When structuring heading hierarchy, heading levels do not have to be sequential but heading levels should not be skipped. For example, appropriate heading hierarchy for a “Contact Us” page could be:

<H1>Contact Us</H1>

<H2>Portland Offices</H2>

<H3>Hawthorne</H3>

<H2>Eugene Offices</H2>

<H3>Downtown</H3>

<H3>River Road</H3>

Other visual formatting, such as items in a list, emphasized or bold text, specific fonts or colored text used for meaning, etc. must be accompanied by markup that relays this information to unsighted users. HTML markup for lists must be used when items are visually in a list – ordered lists are for numbered items and unordered lists are for related but unordered items; <strong> should be used to present boldface text; <em> should be used to present italicized text and so on. <strong> and <em> elements tell assistive technology to inform the user that the text is particularly important or emphasized. CSS should not be used to present visual content in a way that provides information about relationships or meaning in content unless HTML has been used to make that information available to assistive technology.

Structural markup should not be used to achieve purely visual effects. List and blockquote markup should not be used for indentation and spacing. Headings should not be used for stylistic purposes. Empty paragraphs should not be used to create empty spaces. Additionally, glyphs should not be used without providing a text description of the content. CSS and other markup allow for great flexibility in styling without affecting the relationships or meaning of the text content.

For Example: On the Contact Us page you have used BLOCKQUOTE to indent a paragraph.

* Remove the structural markup and use CSS for visual styling.

Because assistive technologies do not process changes in positioning or styling made with CSS, if CSS has been used to style or position content, the content must maintain its position in relation to other relevant content when CSS is not applied. For example, if one paragraph of text is related to a meaningful image and CSS has been used to place the image in relation to the text, when CSS is not applied the image cannot be displayed out of sequence in a way that obscures the relationship to the paragraph of related text. Additionally, CSS cannot be used to hide or remove content that would be confusing if revealed. Assistive technology will announce the presence of all content on the page, even if it is visually hidden with CSS.

When assigning color to text there are two things to keep in mind. Appropriate luminosity contrast levels between the text and background must be maintained. “Large text” is text that is 18pt (24px) font or larger, 14pt (18.66px) font if bold. Large text and the background must have a luminosity contrast level of 3:1. Small text is 18pt (24px) font or smaller, 14pt (18.66px) font if bold. Small text and the background must have a luminosity contrast level of 4.5:1. If color is used for purely ascetic reasons, contrast is the only accessibility related concern. If, however, the text color indicates information about the text, this information must be provided in another way. For example, if words in red are new additions to the document, a non-color indicator such as the use of a different font must be provided. Additionally, the TITLE attribute can provide advisory or supplemental information for any element and could be used to provide a text description of the visual information.

Instructions and references in text to elements on the page cannot rely on sensory characteristics. Text cannot refer to “the chart at right” or “the red button.” References to sensory characteristics must be paired with a secondary non-sensory identification such as, “the Quarterly Sales Report chart at right,” or “the red ‘Go Search ‘button.” The words “above” and “below” are acceptable since they refer to the culturally understood positions of “previous” and “forthcoming” in the flow of content.

If any text on the page is in a language other than the main language of the page, the text must be marked with an attribute that identifies what language it is. For example, if the page is in English but there is a link to a Spanish version and the link text is in Spanish that link must be identified as “Spanish.” This is achieved by adding the “LANG=”es” attribute to the link element. Here is a list of some of the more common languages - Japanese: ja; German: de; Chinese: zh; French: fr; Spanish: es; Italian: it; Dutch: nl; Portuguese: pt; Finnish: fi; Swedish: sv; Norwegian: no; Danish: da’ Korean: ko; Polish: pl; Russian: ru; Hebrew: he; Hungarian: hu; Greek: el; Turkish: tr; Czech: cs; Thai: th; Arabic: ar; Icelandic: is.

A final aspect that must be approached carefully is blinking and/or scrolling text. The <BLINK> element cannot be used because there is no way for the effect to be turned off. If other methods are used to make content blink or scroll (such as <MARQUEE> or a slideshow) the blinking or scrolling must not occur in the range between three times in one second and 55 times in one second (3Hz-55Hz). Additionally, if any blinking content is present – even if it is below the threshold – which lasts longer than 5 seconds there must be a mechanism to pause, stop, or hide the content. If there is any scrolling content present, regardless of the length of time the content scrolls, there must be a mechanism to pause, stop, or hide then restart the content. There are exceptions if the blinking or flashing occurs on very small areas of the screen. Tools and formulas are available to test for this if desired, but due to the complexity of factors such as screen size, viewing distance and resolution, blinking or flashing is best to avoid.

### Text Accessibility Checklist

Text Checklist:

* Heading structure reflects content hierarchy and has been used appropriately.
* Information about relationships in content provided by visual formatting, such as color or font, has been provided in a non-visual way, such as a TITLE on the formatted element.
* Lists, <BLOCKQUOTE>, and other structural markup have been used appropriately and not for visual styling alone.
* <STRONG> and <EM> are used instead of <B> and <I>.
* The order of the HTML code reflects the visual presentation of the content. CSS has not been used to visually place content in a way that makes the HTML order not meaningful.
* CSS has not been used to hide content which would be confusing if revealed.
* Small text has a luminosity contrast ratio of at least 4.5:1.
* Large Text has a luminosity contrast ratio of at least 3:1.
* Meaning provided through the use of color on text is also provided with a non-color indicator.
* Sensory characteristics alone are not used to refer to content on the page.
* Passages written in a language other than the main language of the page are identified with LANG=”…”
* Blinking and scrolling content has not been used that is 3 Hz-55 Hz. If blinking or scrolling content has been used which meets the blinking threshold and lasts longer than 5 seconds it can be paused, stopped, or hidden and then restarted.

# Links

WCAG Guidelines Covered:

1.4.1 Use of Color; 2.1.1 Keyboard Accessible; 2.4.4 Link Purpose (In Context); Level AA – 3.2.4 Consistent Identification

## Link Accessibility Guidance Details

The most important thing to consider when providing users with links is to include sufficiently descriptive link text. Link text of “Read More,” “Learn More,” “Go,” etc. does not describe the destination of the link. Ideally a user should be able to discern the destination or purpose of the link without needing additional contextual information. It is allowable for the destination of a link to be clarified with information contained in the same sentence, paragraph, list item, table cell or table header cell because these are directly associated with the link itself. Contextual information will be most helpful if it precedes the link. ARIA techniques can also be used to associate additional text on the page with the link.

Links in a set of webpages which have the same destination should be consistently identified across the set of pages. Best practices recommend using identical identifications for links with the same destination, but to meet accessibility guidelines the link text only needs to be consistent, not identical.

When links are embedded within unlinked text the link must have at least two methods to distinguish it from the surrounding, unlinked text. The most common method is to make the link a different, underlined color. If underlining is not used, and difference in color is depended on to differentiate the link, the color of the link as compared to the color of the text must have a luminosity contrast ratio of at least 3:1. Other acceptable methods of differentiation are bold font, different font styles, italics, background highlights, etc.

For Example: On the “Make a Suggestion” page you have a link “Read More about Accessibility”. The link is in a paragraph of text. The unlinked text is 14px with color #333333 and the link text color is #3344DD, producing a luminosity contrast ratio of 1.8:1.

* Add a secondary method to distinguish the link from the surrounding, unlinked text such as underlining.

### Link Accessibility Checklist

Link Checklist:

* Link text describes the purpose or function of the link.
* Links with the same destination are consistently described across sets of pages.
* At least two methods are used to differentiate linked text from surrounding unlinked text.

# Tables

WCAG Guidelines Covered:

Level A - 1.3.1 Info and Relationships

## Simple Table Accessibility Guidance Details

Ideally, table markup should only be used for displaying data intended to be presented in a tabular format. Using tables for layout purposes is unnecessary given the flexibility of CSS container positioning. If table markup has been used for layout purposes, the ROLE=”presentation” attribute and value should be included in the TABLE tag so that assistive technology does not interact with it as a data table. Use markup to create a table an image of a table cannot be presented to the user of assistive technology

Tables presenting tabular data need to include appropriate markup in the header cells. Typically, the first row and often the first column of cells in a table are header cells. The information in a header cell describes the content of the cells below or to the right of the header cell. Header cells need to be coded with <TH> tags so assistive technology passes the information they contain to the appropriate cells in the body of the table. For instance, in the simple table example the “Sales Associates,” “Payroll Total Q1,” “Payroll Total Q2,” “Salary Q1,” “Commission Q1,” etc., cells are the appropriate headers. Additionally, the contents of header cells should be unique. ([See Figure 1 for an example of the use of TH tags](#_(Figure_1)_Example))

If a visual title has been provided for the table CAPTION markup should be used. The CAPTION is placed inside the TABLE tag before the markup of the first row and associates the title with the table. The CAPTION can be styled as desired. For example, <CAPTION style=”text-align: center; font-size: large; font-weight: 600;”> Acme Quarterly Sales Q1-Q2</CAPTION>

Depending on the complexity of the table, different methods are used to tell assistive technology that the information in the header cell relates to a specific data cell. For simple tables containing up to two rows of headers and up to two columns of headers the SCOPE attribute is used. The SCOPE attribute is assigned either a row or column value (SCOPE=”row” or SCOPE=”col”); information in column headers is applied top to bottom and information in row headers is applied left to right. ([See Figure 1 for an example of the use of the SCOPE attribute](#_(Figure_1)_Example))

For example: On the Make a Suggestion page, there is a table that has <th> cells without (SCOPE=”row” or SCOPE=”col”).

* Change the th cells from <th>Ideas</th> to <th SCOPE=”col”>Ideas</th>

## (Figure 1) Example of a simple table with TH and SCOPE

**Acme Quarterly Sales Q1-Q2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Sales Associate**  TH SCOPE=”col” | **Payroll Totals Q1**  TH SCOPE=”col” | | **Payroll Totals Q2**  TH SCOPE=”col” | |
| **Salary Q1**  TH SCOPE=”col” | **Commission Q1**  TH SCOPE=”col” | **Salary Q2**  TH SCOPE=”col” | **Commission Q2**  TH SCOPE=”col” |
| **Region 1**  TH SCOPE=”row” | **Susan B.**  TH SCOPE=”row” | $7,500 | $3,400 | $7,750 | $4,200 |
| **Robert G.**  TH SCOPE=”row” | $7,250 | $4,500 | $7,500 | $4,150 |
| **Region 2**  TH SCOPE=”row” | **Kyle P.**  TH SCOPE=”row” | $8,400 | $5,200 | $8,650 | $4,800 |

### (Figure 1) HTML Example

<table border="1">

<caption> Acme Quarterly Sales Q1-Q2</caption>

<tr>

<td rowspan="2"></td>

<th scope="col" rowspan="2">Sales Associate</th>

<th scope="col" colspan="2">Payroll Totals Q1</th>

<th scope="col" colspan="2">Payroll Totals Q2</th>

</tr>

<tr>

<th scope="col">Salary Q1</th>

<th scope="col">Commission Q1</th>

<th scope="col">Salary Q2</th>

<th scope="col">Commission Q2</th>

</tr>

<tr>

<th scope="row" rowspan="2">Region 1</th>

<th scope="row">Susan B. </th>

<td>$7,500</td>

<td>$3,400</td>

<td>$7,750</td>

<td>$4,200</td>

<tr>

<th scope="row">Robert G. </th>

<td>$7,250</td>

<td>$4,500</td>

<td>$7,750</td>

<td>$4,150</td>

</tr>

<tr>

<th scope="row">Region 2</th>

<th scope="row">Kyle P. </th>

<td>$8,400</td>

<td>$5,200</td>

<td>$8,650</td>

<td>$4,8000</td>

</tr>

</table>

### Table Accessibility Checklist

Table Checklist:

* Tables have not been used for layout purposes.
* CAPTION has been used to associate the title of a table with the contents of the table.
* TH has been used for all header cells.
* In simple tables SCOPE has been used on all header cells.