Accessible Forms and Tables

High Tech Center Training Unit

Of the California Community Colleges at the
Foothill-De Anza Community College District

21050 McClellan Road
Cupertino, CA 95014
(408) 996-4636

www.htctu.net

2006 HTCTU
http://creativecommons.org/licenses/by-nd-nc/1.0/

Table of Contents

ACCESSIBLE DATA TABLES 5
Introduction *5*
Formatting Data Tables 5
The Scope Method *6*
Scope Code 6
Scope Example: 6
Scope Method with Dreamweaver MX 2004 ..... *7*
To insert a table: 7
Scope Method with Composer *8*
The "ID" and "Headers" Method *9*
Summary Attribute: *10*
Summary Example: 10
Data Tables and CSS *11*
CREATING ACCESSIBLE FORMS 13
Introduction *13*
Forms and AccVerify/AccRepair *13*
Step 1 – Select Input Field Type 13
Step 2 – Enter Form Field Information 13
Step 3 – Creating the Form 13
Fixing Forms with AccVerify/AccRepair 14
Accessibility and Form Content *15*
Label Tag 15
Fieldset and Legend Tag 16
Tabindex Attribute 16
Other Form Considerations *17*
Drop Down Menus 17
Title Attribute 17
FORMS AND TABLES RESOURCES 19
W3Schools.com *19*
Webaim *19*
W3C Forms Specifications *19*
Guide to Web-based Forms *19*
Accessible Forms and Tables *19*
ADDITIONAL WEB RESOURCES 21
Accessible Web Design Guidelines 21
Section 508 Standards 21
Usability.gov 21
Web Accessibility Initiative (WAI) Guidelines 21
WebAIM 21
WebABLE 21

Accessible Data Tables

Introduction

The rendering of Web content can vary from web browser to web browser in such a manner that presentation and design originally identified for the page are now altered due to variations in a user's Web browser. To solve this issue, many Web authors utilize tables to control the layout of content and its presentation. While tables may be used for presentation purposes, tables are also used to present data information on a Web page. Tables used for presentation purposes are generally referred to as "layout tables" while tables used to organize information are referred to as "data tables". The following information is focused on the appropriate code for accessible data tables .

Formatting Data Tables

Data tables are exactly what the name implies - a table that contains data about some information sequenced in a specific format. For example, if you have a list of days, a list of appointments on specific days, and a list of specific times for those appointments on specific days, it would make sense to develop a table to display that information. Here is a practical example below:

	Sun.	**Mon.**	**Tues.**	**Wed.**	**Thurs.**	**Fri.**	**Sat.**
9:00 AM	Sleep		Work	Work		Work	Sleep
12:00 PM	Nap	Lunch			Lunch		Nap
2:00 PM	Hamster Feeding			Cat Pruning			Breakfast
5:00 PM				Hospital			Nap
6:00 PM		Football					

The benefit of such a data table is that it provides information in a visually organized format such that one can quickly identify the day of the appointment and time in a simple and easy manner. However, this is the exact problem for some types of assistive computer technologies when communicating the page to the user – these technologies do not "visually" interpret the page. Instead, specific code provides cues that can be communicated to the user to better communicate the necessary details of the data. There are several methods to make data tables more accessible using specific HTML attributes.

The Scope Method

The "scope" method is one method that can be used to improve the accessibility of a data table. The scope method can also be used to meet paragraph G of the US Section 508 Standards for Web-based Internet and Intranet Information and Applications (1194.22). Essentially, the scope method identifies the row and/or column header information for an entire row and/or column. The scope method does not require the Web page author to directly associate each table cell with its appropriate row and/or column header. In other words, with the scope method, the data within the entire row or column is automatically associated with the identified header information.

Scope Code
(from Section 508 Standards, www.access-board.gov)
The first row of each table should include column headings. Typically, these column headings are inserted in <TH> tags. These tags at the top of each column should include the following attribute:

scope="col"

By doing this simple step, the text in that cell becomes associated with every cell in that column. The first column of every table should include information identifying information about each row in the table [row headers]. Each of the cells in that first column are created by either <TH> or <TD> tags. Include the following attribute in these cells:

scope="row"

While this technique dramatically improves the usability of a web page, using the scope attribute does not appear to interfere in any way with browsers that do not support the attribute.

Scope Example:

```
<table>
<tr>
<th> </th>
<th scope="col" >Spring</th> <th scope="col" >Summer</th> <th scope="col"
>Autumn</th> <th scope="col" >Winter</th> </tr>
<tr> <td scope="row" >Betty</td> <td>9-5</td> <td>10-6</td> <td>8-4</td><td>7-
3</td>
</tr>
<tr> <td scope="row" >Wilma</td> <td>10-6</td> <td>10-6</td> <td>9-5</td> <td>9-
5</td>
</tr>
<tr> <td scope="row" >Fred</td> <td>10-6</td> <td>10-6</td> <td>10-6</td> <td>10-
6</td>
</tr>
```

</table>

This table would be displayed as follows:

	Spring	**Summer**	**Autumn**	**Winter**
Betty	9-5	10-6	8-4	7-3
Wilma	10-6	10-6	9-5	9-5
Fred	10-6	10-6	10-6	10-6

Scope Method with Dreamweaver MX 2004

Dreamweaver can also be used to create data tables for the visual display of information. Data tables are valuable to show relationships between various pieces of information.

To insert a table:

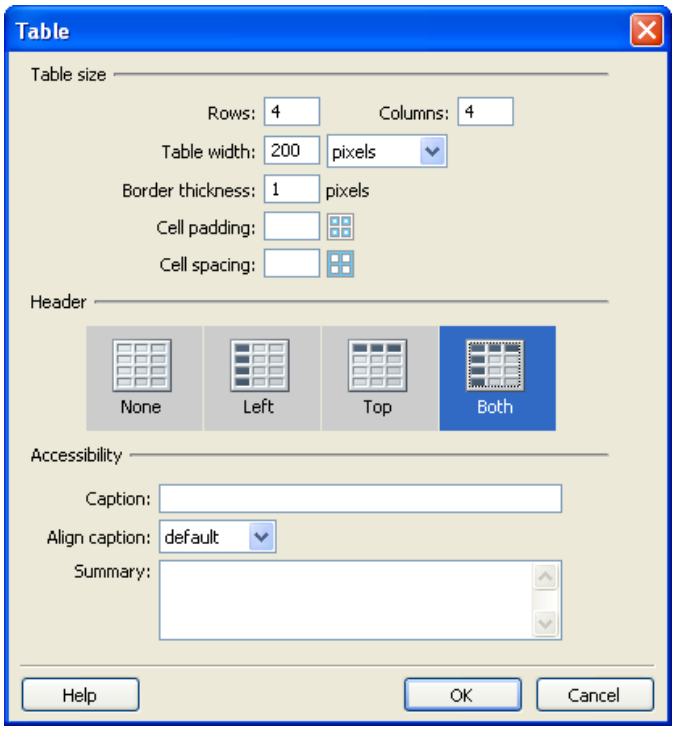
1. Determine the number of rows and columns you will need for your table.

2. Select **Insert** from the menu bar and choose **Table**.

3. Enter the number of rows and columns for your table. You can also determine the Width of your table (it is recommended to use percentages instead of pixels for specifying table width).

4. Select the box with the type of heading information that is appropriate for your data. You should choose a header type for either the row, column, or both.

5. Enter a brief description in the **Summary** field. You do not have to enter information in the **Caption** field, however, it is important to enter a summary of the information contained within the data table.



Scope Method with Composer

Composer can also be used to create data tables for the visual display of information. Data tables are valuable to show relationships between various pieces of information. After you have created a data table using Composer, you can then proceed to add the necessary scope information.

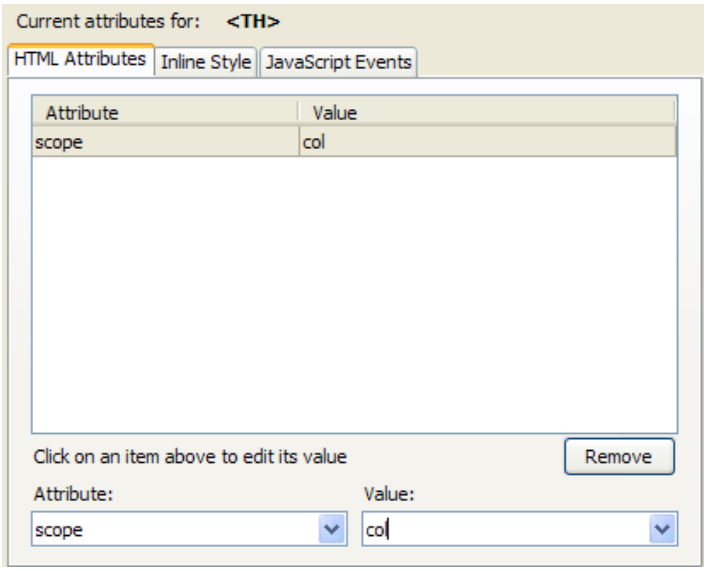
1. Move the cursor to the appropriate header cell and select **Table** from the menu bar. Choose **Table Properties**.

2. Under the **Cells** tab, select the **Advanced Edit** button in the lower right portion of the window.

3. In the drop down box under **Attribute**, select "scope". In the drop down box under **Value**, select "col". "Col" is selected as this is the column header.

4. Select **OK** to return to the Table Properties window. Select **Apply** and then **OK** to complete the process for this cell.

5. Repeat for the remainder of the column headers in the table.



Repeat the process for the row headers. Instead of choosing "col" for the value, it is necessary to choose "row" for the row headers.

The "ID" and "Headers" Method

(from Section 508 Standards, www.access-board.gov)

The "id" and "headers" method provides another method to improve the accessibility of data tables when there are several headers for a particular row or column. The "id" attribute identifies the specific header information for the multiple row or column headings. The "headers" attribute is used within each data cell to specific the appropriate row and column headers for that data cell. Essentially, each data cell becomes uniquely identified as to which heading information it is related to in the table.

"ID" and "Headers" Example:

```
<table>
<tr>
<th> </th>
<th colspan="2" id="winter" >Winter</th>
<th colspan="2" id="summer" >Summer</th>
</tr>
<tr>
<th> </th>
<th id="am1" >Morning</th>
<th id="pm1" >Afternoon</th>
<th id="am2" >Morning</th>
<th id="pm2" >Afternoon</th>
</tr>
```

```
<tr>
<td id="wilma" >Wilma</td>
<td headers="wilma am1 winter" >9-11</td>
<td headers="wilma pm1 winter" >12-6</td>
<td headers="wilma am2 summer" >7-11</td>
<td headers="wilma pm2 summer" >12-3</td>
</tr>
<tr>
<td id="fred" >Fred</td>
<td headers="fred am1 winter" >10-11</td>
<td headers="fred pm1 winter" >12-6</td>
<td headers="fred am2 summer" >9-11</td>
<td headers="fred pm2 summer" >12-5</td>
</tr>
</table>
```

This table would be displayed as follows:

	Winter		**Summer**	
	Morning	**Afternoon**	**Morning**	**Afternoon**
Wilma	9-11	12-6	7-11	12-3
Fred	10-11	12-6	9-11	12-5

Summary Attribute:

Not all assistive technology products will interpret data tables using these options in the same manner, but data tables encoded in an accessible format will provide a greater opportunity for those needing assistive technology to gain access to on-line information. Using the <**summary>** tag can also assist screen-reader users in understanding the layout and content presented in a table. It is important to note that not all assistive technology developers support this attribute at the current time.

Summary Example:

1. Design the table using either the **scope** or **id** and **header** attributes.

2. Within the **<table>** tag insert the attribute **summary= "descriptive table summary goes here"**. It is recommended to place this information as close to the start of the table as possible.

3. A good description of the table will include a brief identification of the content displayed in the row, the column, and the cell.

4. If possible, check the table and summary using screen-reader technology.

Data Tables and CSS

The coding to make a data table more accessible does not mean that the table has to "look" plain or cannot be modified visually. With some basic style sheet information, it is possible to turn a basic looking data table into something more visually appealing while still retaining the necessary code for assistive computer technologies.

For some different data table designs, visit:
http://icant.co.uk/csstablegallery/

To incorporate the data table designs using the above templates, select the "Download" hyperlink for the theme you are interested in applying to your content. You may either view the CSS page in the Web browser or it may prompt for a download to your computer. You may then take this CSS information and add it to your own Web page in order to visually format the data table.

Creating Accessible Forms

Introduction

Forms require the input of information from the user that, while visually understandable, may not make sense when using a screen-reader or other assistive computer technology. For instance, a Web-based form may require information about a person's address to complete a purchase. If the address label information is not associated with the form input field, then the information entered by the user may not be accurate as the individual was unaware of the data required for processing.

Making a web-based form accessible can be accomplished using various HTML attributes that associates the query with the form field elements. Care must be taken, however, when creating forms within tables used for presentation purposes. Nesting table content without the appropriate form elements can make it difficult for screen-readers to communicate the on-screen information to the user in a recognizable format.

Forms and AccVerify/AccRepair

The Accessible Form Creator (AFC) is a tool available from within the AccVerify/AccRepair application from Hi-Software. The AFC can be used to create accessible, HTML-based forms without requiring the author to edit the HTML code directly. To start the form creation process, open AccVerify/AccRepair and choose "Tools" on the menu bar. Select "Accessible Form Creator" to start the form authoring tool.

Step 1 – Select Input Field Type
Select the type of form input field you wish to create (e.g., button, text input, etc.). Most form fields all have a similar set of attributes that you will need to identify in order to create a functional (and accessible) Web-based form.

Step 2 – Enter Form Field Information
For Text Input, enter the field name (this is the name="" attribute of the code) and the Label Text. The Label Text is the onscreen information. You can also specify the size of the text input by entering a value in the "Size" field. Once you have entered the necessary form information, press the "+" (plus-sign) button or select "Field" from the menu bar and choose "Add". In the lower window, you can check the actual HTML code created.

Step 3 – Creating the Form
Create the entire form by selecting "File" and "Create Form". You will need to enter the specific form information that allow for the form processing. Identify if you wish to use the "tabindex" attribute as well as "fieldset" tag. More information is available below regarding use of the "tabindex" and "fieldset" content. Press the "Finish" button to complete the form creation process (you may also select "CSS" to view the style sheet information or "Preview" to preview how your form will appear in a browser).

Fixing Forms with AccVerify/AccRepair

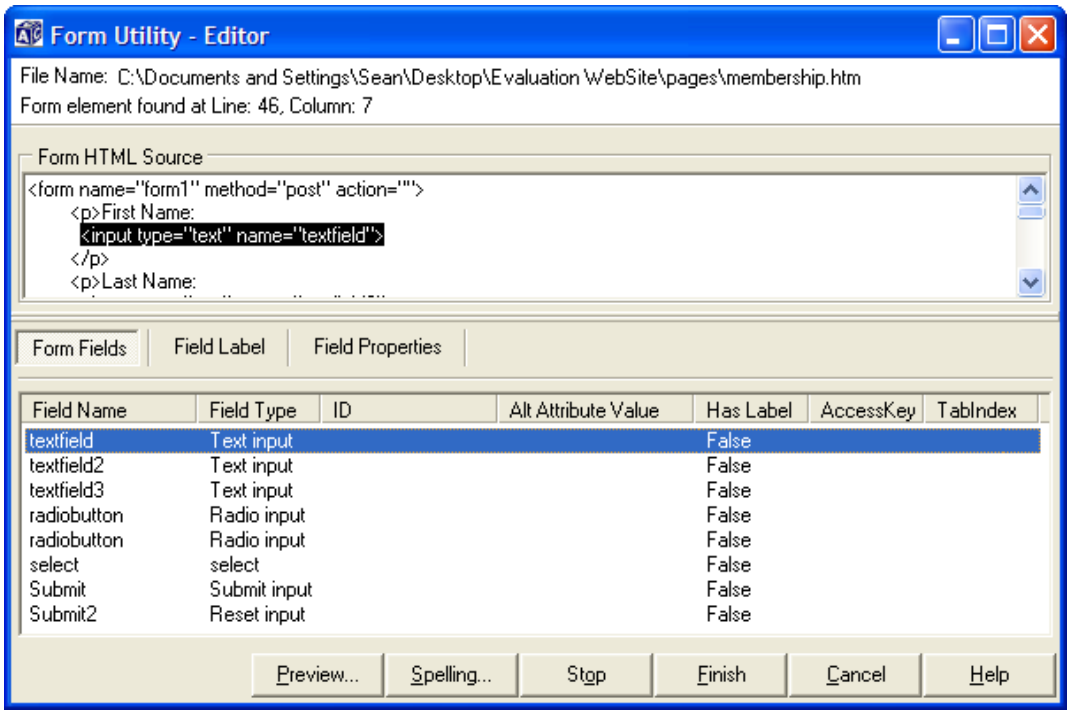
Forms can provide a variety of uses on web pages and most interactive websites have some type of form element present (e.g., Search fields, etc.). Forms can be difficult for some types of assistive computer technology in that the user must interact with the form using two different interactive methods. Usually this requires a user to read the web page and then enter a "forms mode" in order to enter data. However, when in the "forms mode", the user does not have the capability to read the rest of the web page.

Adding specific HTML content to the form input fields and the associated screen text can greatly improve the user experience. It is necessary to manually start the repair process for forms when using the AccVerify/AccRepair application.

1. Select the web content you wish to evaluate and run verification on the selected content. The AccRepair Wizard will not automatically repair forms. However, it is necessary to run the verification in order to the program to detect the presence of forms.

2. Select **Tools** from the menu bar and choose "Form Repair Utility".

3. The Form Utility Editor will launch and allow the web author to make the necessary changes to the form fields. Identify the field name in the list of form fields and select the "Field Label" button.



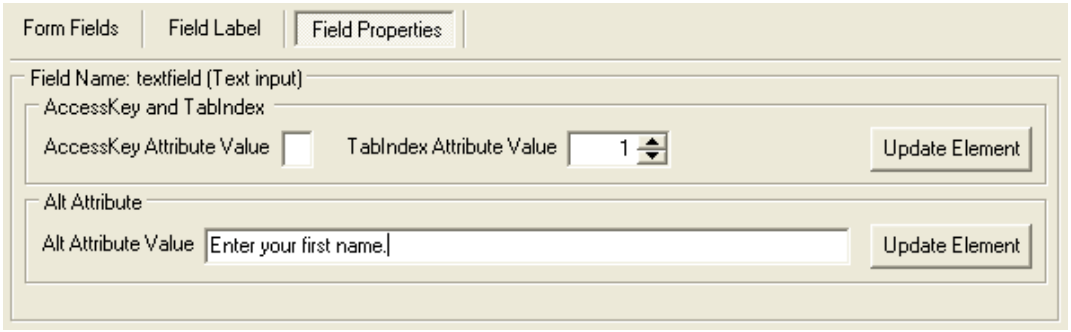
4. Highlight the appropriate text content that you wish to be associated with the form field identified in step 3.

```
<form name="form1" method="post" action="">
      <p>First Name:
        <input type="text" name="textfield">
      </p>
      <p>Last Name:
```

5. Press the "Set Label" button to input the correct <label> tag attribute. Notice the additional HTML code that is inserted into the source code window.

```
<form name="form1" method="post" action="">
      <p><label for="AutoId1">First Name:</label>
        <input id="AutoId1" type="text" name="textfield">
      </p>
      <p>Last Name:
```

6. Select the "Field Properties" button to add information about the tabindex and access key values. You may also add alternative text to improve the user experience with some types of assistive computer technology.



7. Return to the "Form Fields" button and add the necessary information for another form field. Press **Finish** to save the changes and move to another form.

Accessibility and Form Content

If you use another Web authoring tool (Dreamweaver, FrontPage, etc.) to create form content, there are several HTML tags that need to be included in order to support improved form accessibility. The <label> tag should be used for all forms to identify the relationship between the form input element and the onscreen text. In addition to supporting assistive computer technologies, the use of the <label> tag increases the "hot" region for a user to select a form field option.

Label Tag

This tag associates the onscreen text content to the form input field. The result is that when focus is moved to the specific form input field, then assistive computer technology will "read" the onscreen text associated with the form input field. Additionally, if the

user clicks on the onscreen text, this will move focus to the specific form input field. If you are coding this tag by hand, it is important to place the <label> attribute next to the text and form field element you are associating.

```
<form name="form1" method="post" action=" ">
<label for=”fname”>First Name:</label>
<input type=”text” name=”firstname” id=”fname”>
<br>
<label for=”lname”>Last Name:</label>
<input type=”text” name=”lastname” id=”lname”>
<br>
<input type=”checkbox” name=”happyday” id=”hday”>
<label for=”hday”>I am having a happy day.</label>
</form>
```

Fieldset and Legend Tag

The <fieldset> tag draws a box around the set of form input elements that you wish to relate to one another. The <fieldset> HTML tag can also be used with the <legend> tag to create a caption for the form input elements that exist within the <fieldset> attribute. This is particularly useful for radio buttons and checkboxes as assistive computer technology can "read" the question (i.e., legend) content in addition to the various choices.

```
<form name="form1" method="post" action=" ">
<fieldset>
<legend>Enter Legal Name Information</legend>
<label for=”fname”>First Name:</label>
<input type=”text” name=”firstname” id=”fname”>
</fieldset>
</form>
```

Tabindex Attribute

The <tabindex> attribute allows a Web developer to set the order in which form fields (and other elements) are navigated when using the "tab" keyboard button. This insures the correct form field is selected when the user moves through the on-line form using the keyboard instead of the mouse. Navigation proceeds from the lowest tab index value to the next greater tab index value on the page and if the sequence is not set correctly, could result in the focus "jumping" around the screen. If the tabindex attribute is not used, then the form fields will be navigated by their order set in the HTML code. While it is more effective to set the correct HTML order as opposed to using the tabindex attribute, it does remain an option when necessary.

```
<form name="form1" method="post" action=" ">
<fieldset>
<legend>Name Information</legend>
<label for=”fname”>First Name:</label>
```

```
<input type=”text” name=”firstname” id=”fname” tabindex=”1”>
<br>
<label for=”lname”>Last Name:</label>
<input type=”text” name=”lastname” id=”lname” tabindex=”2”>
</fieldset>
</form>
```

Other Form Considerations

Drop Down Menus

Drop down menus provide the flexibility of allowing the user to select from a list of options without the list of information occupying space on a web page. These menus may be developed using the <select>, <optgroup>, and <option> tags. When coupled with a JavaScript function, it is possible to automatically redirect the user to another web page or institute a specific function by selecting one of the drop-down menu options. However, this prevents a user from using the arrow keys alone to scroll through the list of option because the first item selected will initiate the automatic function. While it is possible to access these menus using a combination of keyboard commands, it is generally recommended to provide a "Go" button that directs the user to the appropriate Web page.

Title Attribute

When developing forms for a web page it can be difficult to insure that the form will be easily accessible by all individuals. This is particularly true when creating forms within tables to maintain structure and layout properties from one machine to the next. The use of the <title> tag can be incorporated into the HTML coding to improve the ability of a screen-reader to communicate the on-screen information. For instance, the following example is a form investigating the population of birds during two different times of the year:

	Northern Hemisphere	Southern Hemisphere
Summer		
Winter		

To use the <title> tag, place the code <title=”descriptive information goes here”> within the <input> tag of the form field element. The <title> attribute would then provide the screen-reader user with more information about the information to enter into the form input field. However, the reliability of the <title> tag being supported by assistive computer technology is not consistent and the following method may not provide an equivalent usage experience.

A second method would be to use the <label> tag that provides the necessary label information as to what content should be entered into the field. Using CSS, it is then possible to "move" the actual text off the Web page.

Notes:

Forms and Tables Resources

W3Schools.com

http://www.w3schools.com/html/html_forms.asp

Contains a number of different web building tutorials of varying difficulty levels. Can edit and try out different coding combinations and see the result immediately. W3Schools.com provides information on HTML, CSS, XHTML coding, and JavaScript tutorials.

Webaim

http://www.webaim.org/techniques/forms/

A useful website that demonstrates how inaccessible web-based forms can be and the different workarounds that can be incorporated into the HTML coding. This site also contains an example of how a screen-reader will render an on-line form audibly.

W3C Forms Specifications

http://www.w3.org/TR/REC-html40/interact/forms.html

The website for information about the HTML language and HTML definitions and specifications. Information at this site describes how to implement certain HTML tags in order to achieve the desired result.

Guide to Web-based Forms

http://www.denison.edu/websrv/tutorials/forms/

Informative site on the basics and some advanced concepts to creating web-based forms. Does not contain specific accessibility information.

Accessible Forms and Tables

http://www.jimthatcher.com/webcourse8.htm

This site has been developed by Jim Thatcher to identify the different methods of making HTML more accessible. The site contains example markup and details on how assistive technology will interact with properly coded web pages.

Additional Web Resources

Accessible Web Design Guidelines

Microsoft: http://www.microsoft.com/enable/dev/web/default.htm

WebAIM: http://www.webaim.org/howto/frontpage.php

EASI: http://www.isc.rit.edu/~easi/

Section 508 Standards

http://www.section508.org

http://www.access-board.gov

Usability.gov

http://www.usability.gov

Web Accessibility Initiative (WAI) Guidelines

http://www.w3c.org/WAI/

WebAIM

http://www.webaim.org

WebABLE

http://www.webable.com/