Techniques for Web Content Accessibility

The following material has been compiled from the W3 Site on Accessibility:
http://www.w3.org/TR/2013/NOTE-WCAG20-TECHS-20130905/Overview.html#contents

WCAG = Website Accessibility Guidelines

The following items are provided as a guide to some of the simple techniques you must use to make your web pages accessible to every user of your web pages. These items not only describe good web coding practice, they are required by law.

The following Accessibility items can be found in the web page referenced by the link above.

Accessibility: Using images

Description

When using the `img` element, specify a short text alternative with the `alt` attribute. Note. The value of this attribute is referred to as "alt text".

When an image contains words that are important to understanding the content, the alt text should include those words. This will allow the alt text to play the same function on the page as the image. Note that it does not necessarily describe the visual characteristics of the image itself but must convey the same meaning as the image.

Examples

Example 1

An image on a Website provides a link to a free newsletter. The image contains the text "Free newsletter. Get free recipes, news, and more. Learn more." The alt text matches the text in the image.

Example Code:

```
     <img src="newsletter.gif" alt="Free newsletter.
        Get free recipes, news, and more. Learn more." />
```

Example 2

An image on a Web site depicts the floor plan of a building. The image is an image map with each room an interactive map area. The alt text is "The building's floor plan. Select a room for more information about the purpose or content of the room." The instruction to "select a room" indicates that the image is interactive.

Accessibility: Providing link text that describes the purpose of a link

Description

The objective of this technique is to describe the purpose of a link in the text of the link. The description lets a user distinguish this link from links in the Web page that lead to other destinations and helps the user determine whether to follow the link. The URI of the destination is generally not sufficiently descriptive.

Examples

Example 1: Describing the purpose of a link in HTML in the text content of the a element

Example Code:

```
<a href="routes.html">
  Current routes at Boulders Climbing Gym
</a>
```

Accessibility: Identifying the purpose of a link using link text combined with the text of the enclosing sentence

Description

The objective of this technique is to identify the purpose of a link from the link and its sentence context. The sentence enclosing the link provides context for an otherwise unclear link. The description lets a user distinguish this link from links in the Web page that lead to other destinations and helps the user determine whether to follow the link. Note that simply providing the URI of the destination is generally not sufficiently descriptive.

Note: These descriptions will be most useful to the user if the additional information needed to understand the link precedes the link. If the additional information follows the link, there can be confusion and difficulty for screen reader users who are reading through the page in order (top to bottom).

Examples

Example 1:

A Web page contains the sentence "**To advertise on this page, click here.**"

Although the link phrase 'click here' is not sufficient to understand the link, the information needed precedes the link in the same sentence.

Example 2:

In the news summary containing the sentence "**The Smallville Times reports that the School Board chose a 2007 school calendar that starts on August 27.**", the words "**reports that**" are a link to an article in the Smallville Times about the School Board meeting.

Note: Although this example satisfies the Success Criterion, putting information needed to understand the link after the link in this way is awkward for those who are reading through the document with a screen reader.

Accessibility: Supplementing link text with the title attribute

Description

The objective of this technique is to demonstrate how to use a `title` attribute on an anchor element to provide additional text describing a link. The `title` attribute is used to provide additional information to help clarify or further describe the purpose of a link. If the supplementary information provided through the `title` attribute is something the user should know before following the link, such as a warning, then it should be provided in the link text rather than in the `title` attribute.

Because of the extensive user agent limitations in supporting access to the title attribute, authors should use caution in applying this technique. For this reason, it is preferred that the author use technique C7: Using CSS to hide a portion of the link text (CSS) or H30: Providing link text that describes the purpose of a link for anchor elements.

Examples

Example 1: Clarifying the purpose of a link

Example Code:

```
<a
href="http://example.com/WORLD/africa/kenya.elephants.ap/index.html"
   title="Read more about failed elephant evacuation">
   Evacuation Crumbles Under Jumbo load
</a>
```

Example 2: A link that opens in a new window

In HTML 4.01 the `target="_blank"` attribute can be used on an anchor element to indicate that the URI specified by the `href` attribute will be opened in a new window. This example shows using the `title` attribute of the anchor element to provide information that the link will be opened in a new window.

Example Code:

```
     <a href="http://example.com/subscribe.html"
          target="_blank"
          title="link opens in new window">
          Subscribe to email notifications about breaking news
     </a>
```

Accessibility: Combining adjacent image and text links for the same resource

Description

This objective of this technique is to avoid unnecessary duplication that occurs when adjacent text and iconic versions of a link are contained in a document.

Many kinds of links have both a text and iconic link adjacent to each other. Often the text and the icon link are rendered in separate links, in part to create a slight visual separation from each other. Visually they appear to be the same link, but they are experienced by many people as two identical links and this can be confusing. To avoid this, some authors omit alternative text from the image, but this would fail Success Criterion 1.1.1 because the text alternative would not serve the same purpose as the graphical link. The preferred method to address this is to put the text and image together in one link, and provide null alternative text on the image to eliminate duplication of text.

Sometimes the text and the icon link are rendered in separate, adjacent table cells to facilitate page layout. Although WCAG 2 does not prohibit the use of layout tables, CSS-based layouts are recommended in order to retain the defined semantic meaning of the HTML table elements and to conform to the coding practice of separating presentation from content. If CSS is used, this technique can be applied to combine the links.

Examples

Example 1

The icon and text are contained in the same `a` element.

Example Code:

```
 <a href="products.html">
```

```
   <img src="icon.gif" alt="" />
   Products page
 </a>
```

Example 2

A link contains an icon and text, and the site help refers to the icon. The `img` has a text alternative which is the name used for the icon in the site help, which describes clicking the home page icon.

Example Code:

```
<a href="foo.htm">
  <img src="house.gif" alt="home page icon"/>
  Go to the home page
</a>
```

Failure Example 3

This example demonstrates a failure to apply this technique. An icon and text link are side by side. The text alternative for the image is the same as the text link beside it, leading to a "stutter" effect as the link is read twice.

Example Code:

```
<a href="products.html">
   <img src="icon.gif" alt="Products page" />
 </a>
 <a href="products.html">
   Products page
 </a>
```

Failure Example 4

This example demonstrates a failure to apply this technique. An icon and text link are side by side. In an attempt to remove the "stutter" the text alternative for the image is null. However, now one of the links has an unknown destination, which is its own link text problem.

Example Code:

```
<a href="products.html">
   <img src="icon.gif" alt="" />
 </a>
 <a href="products.html">
   Products page
 </a>
```

Failure Example 5

This example demonstrates an incorrect implementation of this technique. The icon and text are contained in the same a element. However, the text alternative for the icon is a duplicate of the link text, leading to a "stutter" effect as the description is read twice.

Example Code:

```
<a href="products.html">
  <img src="products.gif" alt="Products page"/>
  Products page
</a>
```

Accessibility: Providing link text that describes the purpose of a link for anchor elements

Description

The objective of this technique is to describe the purpose of a link by providing descriptive text as the content of the `a` element. The description lets a user distinguish this link from other links in the Web page and helps the user determine whether to follow the link. The URI of the destination is generally not sufficiently descriptive.

When an image is the only content of a link, the text alternative for the image describes the unique function of the link.

When the content of a link contains both text and one or more images, if the text is sufficient to describe the purpose of the link, the images may have an empty text alternative. (See Using null alt text and no title attribute on img elements for images that assistive technology should ignore.) When the images convey information beyond the purpose of the link, they must also have appropriate alt text.

Examples

Example 1

Describing the purpose of a link in HTML in the text content of the `a` element.

Example Code:

```
     <a href="routes.html">
       Current routes at Boulders Climbing Gym
     </a>
```

Example 2

Using the `alt` attribute for the `img` element to describe the purpose of a graphical link.

Example Code:

```
<a href="routes.html">
   <img src="topo.gif" alt="Current routes at Boulders Climbing Gym"
/>
</a>
```

Example 3

Using an empty alt attribute when the anchor (`a`) element contains text that describes the purpose of the link in addition to the `img` element. Note that the link text will appear on the page next to the image.

Example Code:

```
     <a href="routes.html">
       <img src="topo.gif" alt="" />
       Current routes at Boulders Climbing Gym
     </a>
```

Example 4

A site allows users to provide feedback on products, when the user is logged in, by clicking on the "Feedback" link in a product detail page. Other users or the product manufacturer are able to provide a response to any feedback. The feedback link displays an icon before the "Feedback" text when a response to the user's feedback is available. The help information describes this icon as a talking bubble containing quotation marks and includes the icon itself as an example. The icon's text alternative in the help text is "Response received icon". The same text alternative is used in the product detail pages (when a response is available) to allow identification of this icon through multiple modalities.

Example Code:

```
     <a href="prod_123_feedback.htm">Feedback <img src="response.gif"
     width="15" height="15" alt="Response received icon" /></a>
```

Example 5

A link contains text and an icon, and the icon provides additional information about the target.

Example Code:

```
<a href="WMFP.pdf">
Woodend Music Festival Program
<img src="pdficon.gif" alt="PDF format"/>
</a>
```

Example 6

The “MyCorp” company’s annual report is made available on the corporate website as a PDF file, and the annual corporate budget is made available as an Excel file on the web site.

Note: Many users prefer to know the file type when opening a file that results in opening a new application to view the file, so it is often regarded as useful to include this additional information. However, this is not required for compliance with this success criterion.

Example Code:

```
    <p>
    <a href=”mycorp_report.pdf”>MyCorp Annual Report (pdf)</a><br />
    <a href=”mycorp_budget.xls”>MyCorp Annual Budget (Excel)</a>
    </p>
```

Accessibility: Using alt attributes on images used as submit buttons

Description

For input elements of type 'image', the `alt` attribute of the `input` element is used to provide a functional label. This label indicates the button's function, but does not attempt to describe the image. The label is especially important if there are multiple submit buttons on the page that each lead to different results.

The `input` element is used to create many kinds of form controls. Although the HTML and XHTML DTDs permits the `alt` attribute on all of these, it should be used only on image submit buttons. User agent support for this attribute on other types of form controls is not well defined, and other mechanisms are used to label these controls.

Examples

Example 1

An `input` element with an `alt` attribute

Example Code:

```
  <form action="http://example.com/prog/text-read" method="post">
    <input type="image" name="submit" src="button.gif" alt="Submit" />
  </form>
```

Accessibility: Creating a logical tab order through links, form controls, and objects

Description

The objective of this technique is to provide a logical tab order when the default tab order does not suffice. Often, G59: Placing the interactive elements in an order that follows sequences and relationships within the content is sufficient and this technique is not necessary. It can be very easy to introduce usability bugs when setting the tab order explicitly.

In some cases, the author may want to specify a tab order that follows relationships in the content without following the order of the interactive elements in the code. In these cases, an alternative order can be specified using the `tabindex` attribute of the interactive element. The `tabindex` is given a value between 0 and 32767.

When the interactive elements are navigated using the tab key, the elements are given focus in increasing order of the value of their `tabindex` attribute. Elements that have a `tabindex` value higher than zero will receive focus before elements without a `tabindex` or a `tabindex` of 0. After all of the elements with a tabindex higher than 0 have received focus, the rest of the interactive elements are given focus in the order in which they appear in the Web page.

Examples

Example 1

A genealogical search form searches for marriage records. The search form includes several input fields for the bride and the groom. The form is marked up using a data table that includes the fields of the groom in the first column and the fields of the bride in the second column. The order in the content is row by row but the author feels it is more logical to navigate the form column by column. This way, all the groom's criteria can be filled in before moving on to the bride's criteria. The `tabindex` attributes of the input fields are used to specify a tab order that navigates column by column.

Example Code:

```
<form action="#" method="post">
 <table summary="the first column contains the search criteria
  of the groom, the second column the search criteria of
  of the bride">
 <caption>Search for marriage records</caption>
 <tr>
   <th>Search criteria</th>
   <th>Groom</th>
   <th>Bride</th>
 </tr>
 <tr>
  <th>First name</th>
  <td><input type="text" size="30" value="" name="groomfirst"
      title="First name of the groom" tabindex="1"></td>
  <td><input type="text" size="30" value="" name="bridefirst"
       title="First name of the bride" tabindex="4"></td>
 </tr>
 <tr>
  <th>Last name</th>
  <td><input type="text" size="30" value="" name="groomlast"
      title="Last name of the groom" tabindex="2"></td>
  <td><input type="text" size="30" value="" name="bridelast"
      title="Last name of the bride" tabindex="5"></td>
 </tr>
 <tr>
  <th>Place of birth</th>
  <td><input type="text" size="30" value="" name="groombirth"
      title="Place of birth of the groom" tabindex="3"></td>
  <td><input type="text" size="30" value="" name="bridebirth"
      title="Place of birth of the bride" tabindex="6"></td>
 </tr>
</table>
</form>
```

Example 2

A Web page contains a search field in the upper right corner. The field is given tabindex="1" so that it will occur first in the tab order, even though it is not first in the content order.

Example 3

`Tabindex` values need not be sequential nor must they begin with any particular value. The values do not have to be unique. Elements that have identical `tabindex` values are navigated in the order they appear in the character stream.

In sections of the content where the tab order follows the content order, it can be less error prone to give all elements the same tabindex value rather than specifying a different number for each element. Then it is easy to rearrange those elements or add new elements and maintain a logical tab order.

Example Code:

```
<a href="xxx" tabindex = "1">First link in list</a>
<a href="xxx" tabindex = "1">Second link in list</a>
<a href="xxx" tabindex = "1">Link that was added long
  after the original list was created</a>
<a href="xxx" tabindex = "1">Third link in list</a>
  ...
<a href="xxx" tabindex = "1">Twentieth link in list</a>
```

Accessibility: Giving users advanced warning when opening a new window

Description

The objective of this technique is to provide a warning before automatically opening a new window or tab. Opening new windows automatically when a link is activated can be disorienting for people who have difficulty perceiving visual content, and for some people with cognitive disabilities, if they are not warned in advance. Providing a warning allows the user to decide it they want to leave the current window, and the warning will help them find their way back, if they do decide they would like to go to the new window. It will help them understand that the "back" button will not work and that they have to return to the last window they had open, in order to find their previous location.

Examples

Example 1: Including the warning in the text describing a control

The name or label that describes a control can include the warning about opening in a new window.

Example Code:

```
     <a href="knitting.html" target="_blank
          All about Knitting (opens in new window)
     </a>
```

Example 2: Using CSS to provide a warning before opening a new window

The code below uses CSS to provide a warning before opening a new window.

Example Code:

```
              <html>
```

```
                <head>
                <title>Pop-Up Warning</title>
                <style type="text/css">
                body {
                margin-left:2em;
                margin-right:2em;
                }
                :focus { outline: 0; }
                a.info {
                position:relative;
                z-index:24;
                background-color:#ccc;
                color:#000;
                text-decoration:none
                }
                a.info:hover, a.info:focus, a.info:active {
                z-index:25;
                background-color:#ff0
                }
                a.info span {
                position: absolute;
                left: -9000px;
                width: 0;
                overflow: hidden;
                }
                a.info:hover span, a.info:focus span, a.info:active
span {
                display:block;
                position:absolute;
                top:1em; left:1em; width:12em;
                border:1px solid #0cf;
                background-color:#cff;
                color:#000;
                text-align: center
                }
                div.example {
                margin-left: 5em;
                }
                </style>
                </head>
                <body>
                <h1>Pop-Up Warning</h1>
                <p> This is an example of an <a class="info"
                href="popup_advisory_technique.html" target="_blank">
                <strong>External link</strong><span>Opens a new
                window</span></a>
                </p>
                </body>
                </html>
```