Section 508 Self-Evaluation

Web Page Accessibility Questionnaire for Component Web Contacts

The World Wide Web has dramatically changed the way that federal agencies disseminate and collect information. Computer users from all around the world have instant access to federal agencies any time of day. Not only do agencies save countless dollars in printing and mailing costs, they also receive and process accurate information much more quickly and efficiently.

Under the authority of section 508 of the Rehabilitation Act, the Access Board (or 'Architectural and Transportation Barriers Compliance Board') has enacted regulations setting forth accessibility standards for federal agencies' electronic and information technology, including web pages.[1] 36 C.F.R. pt. 1194.

This document is designed to help agencies evaluate whether their Internet pages are accessible to people with disabilities. The Access Board's web accessibility standards appear at the beginning of each of the substantive sections, below, and form the basis for most of the questions. A few of the questions are designed to suggest ways that agencies should make their programs accessible, even where there are barriers on existing agency web pages.
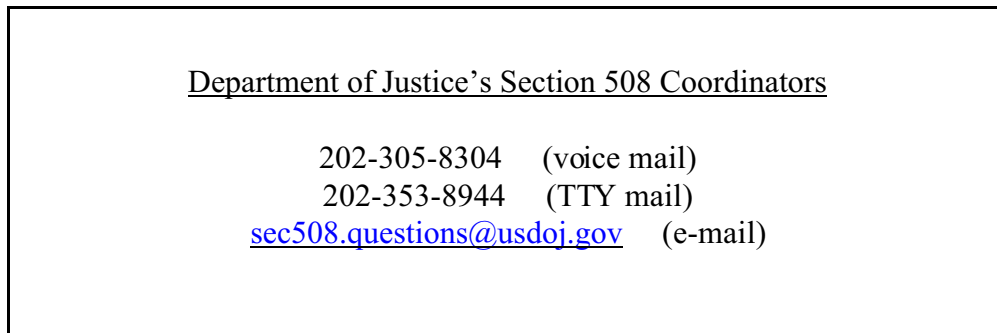
Directions for Component Web Contacts:

1. Make 20 copies of this form.

2. Identify your component's 20 most popular web sites. Choose the 20 pages with the most "hits" per week. If you do not have a way to track usage, pick the top 20 pages in order of hierarchy by looking at those pages accessed most directly from your component's home page. Include Intranet pages among your chosen 20 if they are among your component's 20 most popular pages.

3. For each of your 20 designated web pages, answer the questions listed below.

4. Enter your data on the web site, http://www.508.org. To access the site, use the username and password assigned to you by your Designated Agency Official.

---

[1]To the extent that there are any inconsistencies between this Questionnaire and any policy determination, advisory opinion, or technical assistance provided by the Access Board regarding its own rule, the Access Board's determination should control.

5.	**By April 6, 2001,** submit your data electronically to the Department of Justice by following the instructions on the http://www.508.org website.

6.	Print and retain a copy of your responses.

7.	Inform your Designated Agency Official that you have completed your survey responsibilities.

Please contact us if you have any questions.

<div style="border:1px solid black">

Department of Justice's Section 508 Coordinators

202-305-8304	(voice mail)
202-353-8944	(TTY mail)
sec508.questions@usdoj.gov	(e-mail)

</div>

Selecting the Web Pages to Survey

●	*Understanding the URL/URI*

To access your agency's web site, a computer user uses *browser* software (e.g. Internet Explorer, Netscape Navigator, and others) that takes in specially encoded *web pages* and processes them to be viewable on a computer screen. To receive this information, the user's computer makes a *request* over the Internet to an agency computer called a *web server*. The web server responds to the request by sending the encoded web page back to the user and the user's browser makes the web page viewable on a computer monitor or other device.

When a user's computer makes a request of the web server, there has to be some mechanism for ensuring that the request goes to the *right* web server and requests the correct web page. This task is handled through the *Uniform Resource Locator* (or "URL") or the *Uniform Resource Identifier* (or "URI"). The URL or URI is usually displayed at the top of the browser software each time a web page is accessed. For instance, almost all federal agencies maintain a *home page.* Typing in the URL for this page will take the user to the agency's home page.

Example 1:	Typing in http://www.usdoj.gov will take you to the U.S. Department of Justice's home page.

In addition to a home page, agencies will maintain many other pages identified by different URL's.

> **Example 2:** Typing in http://www.usdoj.gov/crt/ada/adahom1.htm will take you to the U.S. Department of Justice's Americans with Disabilities Act home page.

- ***Pages Generated by Search Engines***

Many agency web sites contain search engines that, when activated, will provide a list of agency resources that matches the text submitted by the user. These search engines operate by invoking a computer program that changes the content of the page based on the user's search term. The URI may change slightly, as different information is passed to the computer program that performs the search. Although the *location* box (used in Netscape Navigator) may seem to reflect a different URI for each search request, the resulting pages should not be considered separate pages. For instance, searching for the word "508" on the U.S. Department of Justice Home Page creates a page with location:

> http://search.usdoj.gov/compass?scope=508&ui=sr&view-template=dojsimple

By contrast, searching for the acronym "ADA" on the same page yields a results page with location:

> http://search.usdoj.gov/compass?scope=ADA&ui=sr&view-template=dojsimple

For purposes of this survey, these two separate searches request the same page. If you are surveying such a page, you should only complete one evaluation for it.[2]

- ***Hyperlinks Within the Same Web Page***

Many links in agency web pages just take the user to another part of the same page. These links are called "name anchors." Name anchors are often used for cross-referencing material on lengthy web pages. Selecting one of these links may cause a change in the URI without changing the actual page that is displayed. Because the identity of the web page does not change, if you are surveying a page with name anchors, only complete one evaluation for it.

---

[2]The examples above use so-called *"URL parameters"* to pass search terms to the agency's server that actually performs the search. Search engines can also pass these terms as "form variables," which would not affect the URI.

- ***Intranet Pages***

Many agencies have reserved certain web pages for the exclusive use of their own employees. These pages are called "Intranet" pages. When choosing your agency's 20 most popular web pages to survey, you may include some Intranet pages.

Q1: **What is the URL/URI of the web page?**

_____

Q2: **What is the best description for the purpose of the page?**

a. Web-based application.
b. Online form for receipt of services or benefits.
c. Other online form.
d. Instructions for receipt of services or benefits.
e. Description of activities of your component.
f. Employment postings.
g. Inherently graphical content (e.g. map or photograph).
h. Other (describe): _____

Q3: **Is the page available on the Internet, an agency Intranet, or some combination?**

a. Intranet. Deployed only over internal Intranet.
b. Internet. Deployed over Internet – available to those outside agency.
c. Combination. Deployed over Internet, but containing some portions that are available only on agency terminals.

Q4: **On a weekly basis, how many times (approximately) is this page accessed by users?**

a.   On a weekly basis, this page is accessed approximately _____ times.

b.   We have no way to track this information.

Overview of HTML

As noted above, *browser* software interprets specially coded web pages and renders the content on the computer's screen. The page itself is generally coded in HTML ("HyperText Markup Language"), which is simply a standard way of representing the format and content of all information (other than text) on a web page with regular keyboard characters. Using HTML, for instance, a web designer can easily insert "tags" to indicate boldface font, align text into tables, or place images (which are separate computer files) in specific locations on a web page. More information about the features of HTML can be found at http://www.w3c.org.

Although HTML is not complicated, many web designers use authoring tools (e.g. Microsoft FrontPage, Macromedia DreamWeaver, etc.) to facilitate rapid web page development. Using one of these programs lets web designers quickly write HTML pages that include carefully aligned graphics and consistent font selection.

Using a web browser, you can easily view a web page's underlying HTML coding. For instance, in Netscape Navigator, selecting *View*, then *Page Source* will reveal the page's HTML source coding. In Internet Explorer, selecting *View*, then *Source* will do the same thing.

Non-Text Elements

"A text equivalent for every non-text element shall be provided (e.g., via 'alt', 'longdesc', or in element content)." 36 C.F.R. § 1194.22(a)

In order to make non-text elements[3] (such as images, multimedia objects, logos, photographs, artwork, Java applets, or other types of web page content that cannot be reduced to ASCII text)

---

[3]For the purposes of this survey, the phrase "non-text elements" does not include HTML "tags" that control all elements other than text, including bold face, paragraph breaks, etc.

readable by some of the assistive technologies – such as screen readers used by people who are blind[4] – it is important to accompany non-text elements with meaningful text labels. Assistive technology cannot describe images, but can convey text associated with those images. For instance, a screen reader cannot communicate whether an image is a weather satellite photo or a sketch of Abraham Lincoln unless these images are accompanied by meaningful text labels.

To embed an image into a web page, the designer uses an HTML tag to identify or point to a separate computer file that contains the actual image file. These tags can generally be identified by examining a page's HTML source coding. Images are usually indicated with an <IMG> tag. Java applets are usually embedded using the <APPLET> tag. Another tag commonly used to embed these non-text elements is the <OJBECT> tag. If you notice any of these tags in your HTML source, you likely have non-text elements that require special consideration to ensure that the information they convey is accessible to people with disabilities. For instance, the following tag is used to display an image, which is saved in a separate file called "pictureofme.jpg:"

> **Example 3:** <IMG src="pictureofme.jpg">

While a screen reader cannot "look" at an image and identify all of its features, it can read text descriptions of that image. The same is true for refreshable Braille displays[5] and other types of assistive technology delivery devices. Meaningful text descriptions must accompany non-text elements so that users with disabilities can distinguish important content from merely decorative images. There are several ways of providing this textual information so that it can be recognized by assistive technology devices. For instance, the <IMG> tag can accept an "alt" attribute that will enable a web designer to include text that describes the picture directly in the <IMG> tag. Example 3, above, could be modified to include the "alt" description, "this is a picture of me," as follows:

> **Example 4:** <IMG src="pictureofme.jpg" alt="this is a picture of me">

Similarly, the <APPLET> tag for Java applets also accepts an "alt" attribute, but it only works for browsers that provide support for Java. Because some browsers used by persons with disabilities do not accept Java applets, a better alternative for providing textual descriptions is to simply include the alternative text between opening and closing <APPLET> or <OBJECT> tags. For instance, if a web designer wanted to include an applet called *MyCoolApplet* in a web page,

---

[4]A screen reader is software that reads aloud in an artificial voice the words and punctuation marks that would normally appear on a computer screen. Screen readers are primarily used by people who are blind or who have low vision.

[5]A refreshable Braille display device moves metal points up and down to form the pixels used in Braille lettering. Devices usually display one line of text at a time and allow the user to advance at his or her own speed.

but also include a description that the applet shows a stock ticker displaying the current price of various stocks, she could include it as:

Example 5:   <APPLET CODE="MyCoolApplet.class" WIDTH="200", HEIGHT="100">
This applet displays current stock prices for many popular stocks
</APPLET>

Finally, yet another way of providing a textual description is to include it in the page in the surrounding context. For instance:

Example 6:   Below is a picture of me during my great vacation!<p>
<IMG src="pictureofme.jpg">

Q5:   **Does each non-text element on the page have a text equivalent via "alt" (alternative text attribute) or does the page otherwise include a meaningful description of the non-text element in the text accompanying the non-text element**?

a.   Yes.
b.   Yes and no. Some non-text elements have meaningful text equivalents while others do not.
c.   No, none of the non-text elements have text equivalents.
d.   N/A. There are no non-text elements on the page.

Multimedia

> **"Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation." 36 C.F.R. § 1194.22(b)**

As the World-Wide Web has evolved over the past several years, the content of web pages has also evolved from basic HTML for laying out and formatting text to content that is much richer and more complex. In addition to images and programmatic elements, web developers have also embraced multimedia technologies which allow web pages to include presentations, short movies, or other content with synchronized audio and video tracks. Some developers have also started using such technologies as *RealVideo*, a form of "streaming" multimedia that allows users to watch online videos with relatively little delay.

Generally, multimedia presentations require users to rely on more than one sense, such as combining a reliance on hearing and vision to understand the content of a page. While these advances make web pages more interesting and "layered" for people without disabilities, those with disabilities face new barriers to understanding a page's content.

To ensure that multimedia is accessible to users with visual impairments, agencies should ensure that audio descriptions of video portions are provided in all cases, except where the video portion of a presentation is unimportant to understanding a presentation. For instance, if a multimedia presentation includes a narrator describing an event, it is not necessary to include as an audio description, "The narrator is wearing a blue sweater and brown pants." Audio descriptions are usually selectable, so that a non-disabled user can choose not to turn them on.

To ensure that multimedia presentations are accessible to users with hearing impairments, text captioning should be provided to convey in text any important audio information. Such captioning can be included as part of the video track of the presentation, and can also be selectable.

Finally, agencies should be aware that because multimedia content cannot be rendered using HTML, plug-ins or other applications will be necessary in order to view multimedia content. These other technologies are also subject to the Section 508 rule and are treated separately below.

Q6: **For any multimedia content, is text captioning provided for all audible output and audible output provided for all important visual information?**

a. Yes.
b. Yes and no. Text captioning is not provided for audible output, but audio descriptions are provided for all important visual information.
c. Yes and no. Audio descriptions are not provided for all important visual information, but text captioning is provided for audible output.
d. No, neither is provided.
e. N/A. There is no multimedia content on the page.

Q7: **Are all audio descriptions and text captions synchronized with their associated dynamic content?**

a. Yes, text captioning for audible output and audible output for visual information is completely synchronized with changes in the dynamic content of the page.

b. Yes and no. Text captioning is not completely synchronized with audible output as the dynamic content of the page changes or is not provided, but the audio descriptions are synchronized to the important visual information they describe.

c. Yes and no. Audio descriptions are not synchronized to the important visual information they describe or they are not provided, but text captioning is synchronized with the audible output as the dynamic content of the page changes.

d. No. Both are provided but neither is synchronized.

e. N/A. There is no multimedia content on this page.

Use of Color

> **"Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup." 36 C.F.R. § 1194.22(c)**

Relying on color to communicate important information will affect persons who find it impossible or difficult to discern or differentiate colors, such as persons who are blind, or who have limited sight or color-blindness. Also, persons with cognitive impairments may not be able to fully comprehend or remember the significance assisted to a certain color for use on a web page.

While good web page design often makes full use of color, pages should be fully understandable even by those who cannot discern or differentiate colors. For instance, if a web page has one green button and one red button that are identical except for color, the web page should not state, "press the green button to go to the next page or press the red button to start over." Making this element accessible merely requires the buttons to be labeled: "green" and "red," or "next" and "start over."

Hint: Two simple ways of testing a web page to determine if color is used appropriately is by viewing the page on a black and white monitor, or by printing it out on a black and white printer.

Q8: **Is the page capable of being understood and navigated even if users do not have the ability to identify specific colors or differentiate between colors?**

a. Yes, the page uses color appropriately.
b. No, the page does not use color appropriately.

Style Sheets

"Documents shall be organized so they are readable without requiring an associated style sheet." 36 C.F.R. § 1194.22(d)

One of the difficulties in maintaining multiple web pages on a single site is presenting a consistent "look and feel" to the visitor. Because the appearance of a web page is controlled by a myriad of HTML tags (such as *<b>* for boldface, *<i>* for italics for basic character fonts), creating a multi-page web site that has a consistent style can be challenging. Also, changing that style can be a time-consuming and daunting task.

HTML has evolved to meet this challenge. *Style sheets* allow web designers to create a basic set of rules governing specified web pages to determine how those pages will appear to visitors. More specifically, style sheets allow a web designer to specify rules of document style (e.g. fonts, margins, colors, etc.) once and have these rules applied to elements of document structure (e.g. headings, paragraphs, tables, lists, etc.). For instance, a web designer can specify in a single style sheet that:

- all paragraph headings will appear in large, boldface, blue font
- all paragraph text will appear in small, italic, black font
- all tables will appear with borders and will have large spacing between table cells

When the web designer decides to change the look of the associated web pages – for instance, changing all paragraph headings to black font – the web designer only needs to make a single change in a style sheet to affect the change throughout those pages.

While style sheets provide uniform control over all associated web pages on the site, a designer may want to override these controls for a specific purpose. For instance, the style sheet may be set up to use an italic font for highlighting, but the designer may decide that on a particular page, bold faced type would better accentuate a highlighted word. Alternatively, the designer may want to depart from the style sheet specifications for an entire web page. For instance, a web page that includes large tables may need a smaller font and hairline borders between cells to display all of the information than is called for on the style sheet. Because these variations in style may occur at different "levels" (e.g. a web page, a paragraph, or a specific word), style sheets provide a hierarchy of rules, with specific rules taking precedence over more general rules. Rules affecting a specific element in a web page will take precedence over rules that affect an entire site. Therefore, when a web browser displays a page, it must look for specific rules first, then less specific rules, then general rules, thus giving these standards the term *cascading style sheets*.

Because style sheets 'cascade,' they can occur at different locations in a document. For instance, style sheets that affect an entire web site may be included in a separate file, whereas style sheets affecting a specific web page will usually be included at the beginning of the page. Usually, you can find style sheets by checking the HTML source for the word *style* in an HTML tag, such as *<STYLE TYPE="text/css">*.[6]

Used properly, style sheets can increase the accessibility of web pages for people with disabilities. Some browsers enable users to use style sheets to set their own preferences for how pages are displayed. For instance, a user with low vision may use a style sheet so that regardless of what web pages he visits, all text is displayed in an extra large font with white characters on a black background. If designers set up their pages to override user-defined style sheets, people with disabilities may not be able to use those pages. For accessibility reasons, therefore, it is critical that designers ensure that their web pages do not interfere with user-defined style sheets.

Another potential problem with style sheets is that only the most modern browsers are compatible with them. In some cases, people with disabilities must use older browsers that work well with their assistive technologies. If a web page can only be understood or used by visitors using the latest browsers that support style sheets, some people with disabilities will be unable to use those pages. Therefore, web designers must ensure that their web pages are usable when style sheets are not supported or when support for style sheets is turned off.

---

[6]Netscape Navigator and Internet Explorer also let web designers control document style with scripts. This technology is sometimes referred to as *JavaScript Style Sheets*. Usually, these style sheets can also be identified by the web designer's use of *<STYLE>* tags.

Q9: **If the page uses cascading style sheets or JavaScript style sheets, is it viewable without style sheets or with style sheets turned off or not supported by the browser?**

a. Yes.
b. No.
c. N/A. The page does not use cascading style sheets or JavaScript style sheets.

Q10: **If the page uses cascading style sheets or JavaScript style sheets, is it designed so that it does not interfere with style sheets set by the browser?**

a. Yes, it works without interfering with style sheets set by the browser.
b. No, it interferes with any style sheets that have been set by the browser.
c. N/A. The page does not use cascading style sheets or JavaScript style sheets.

Image Maps

"Redundant text links shall be provided for each active region of a server-side image map." 36 C.F.R. § 1194.22(e)

"Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape." 36 C.F.R. § 1194.22(f)

Image maps are graphic images that encapsulate links to other web pages or which initiate scripts. For instance, Example 7, below, is a geographic map of the United States used as an image map to allow users to select a specific state to locate information only relevant to that state:
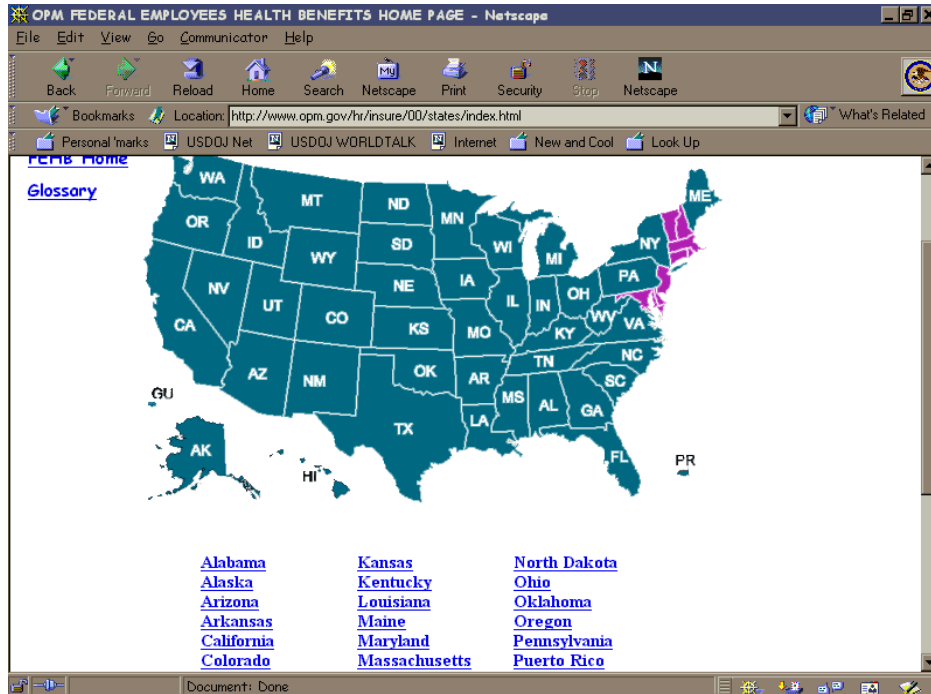
Example 7

Image maps may also include other types of graphic images. Some web designers put text links in images to present a very clean look to their web pages or to embed text messages inside of icons. Example 8 is used frequently by the National Highway Transportation Safety Board as a navigation tool. It is an image of an automobile dashboard that includes links to other web pages, with each link associated with a part of a car's dashboard:

Example 8

Although the image used in Example 8 is not a *geographic* map, it is nevertheless considered an *image* map for purposes of this survey.

Client-side image maps v. server-side image maps. In general, there are two types of image maps: server-side image maps and client-side image maps. When a user selects an area of one of the older *server-side image maps*, the vertical and horizontal coordinates of the cursor location are transmitted back to the web page's server, which then calculates in which region of the image

the user's cursor is located and responds appropriately. By contrast, newer *client-side image maps* use the user's browser to determine which region the user has selected. In general, server-side image maps are less efficient because the web server, which may already be overburdened serving web pages to other users, must calculate how to respond based on the exact coordinates of where a user clicked their mouse button within an image.

While image maps do not present insurmountable problems for people with disabilities if they are designed correctly, it is far easier for many people with disabilities to use client-side image maps than server-side image maps. To make client-side image maps accessible, each region within the map can be assigned an *alt description* that can be read by a screen reader or other assistive technology device. As discussed above, using the *alt attribute* is the technique that is generally used to make ordinary images accessible. The same cannot be said of server-side image maps. Server-side image maps must be accompanied by redundant text links. For instance, in Example 7, above, the agency has made the image map accessible by providing a duplicate list of links – here, the names of states – for each of the areas identified in each region of the map.

To determine whether a web page includes client-side image maps, you can review the page's HTML source coding for any tags that begin with the word *map*. For instance, the tag shown in Example 9 usually indicates a client-side image map:

Example 9:   <MAP NAME="USMap">

Server-side image maps are slightly more difficult to identify within the page's HTML source coding, because there are several tags that will send the horizontal and vertical coordinates of the cursor location back to the server. Some indications that a web page includes a server-side image map are the presence of the *ISMAP* attribute in an image tag or a form input tag specifying a parameter of *IMAGE*. If the HTML source code for the page includes tags of one of the following formats, then the page *may* include a server-side image map:[7]

Example 10:   <IMG SRC="picture.jpg" ISMAP> or
                         <INPUT TYPE="image">

Because it is so much easier for client-side image maps to be made fully accessible to people with disabilities, the Access Board's final rule requires that agencies use client-side image maps instead of server-side image maps, except when the map regions cannot be defined with a n available geometric shape.

---

[7]The presence of either of these tags, however, does not mean that the page necessarily includes a server-side image map. For instance, a web page may include a form that includes an *<INPUT="image"...>* tag that allows a user to select the form by clicking anywhere on the image. In this case, while the horizontal and vertical coordinates are sent back to the server, the server ignores this information when responding.

Q11: **If the page includes any server-side image maps, are duplicate text links provided for all links within the server-side image maps?**

a. Yes, each link in the server-side image maps is duplicated by a separate text link.
b. Yes and no. Some of the links from the server-side image maps are duplicated in separate text links while others are not.
c. No, redundant text links are not provided for any link from the server-side image maps.
d. N/A. The page does not include any server-side image maps.

Q12: **If the page includes any server-side image maps, have you established a timetable to replace the server-side image maps with client-side image maps, except where the regions cannot be defined with an available geometric shape?**

a. Yes.
b. No.
c. N/A. The page does not include any server-side image maps.

Q13: **If the page includes one or more client-side image maps, does each map region have a text equivalent via "alt" (alternative text attribute) or does the page otherwise include a meaningful description of the non-text element in the text accompanying it?**

a. Yes.
b. Yes and no. Some of the non-text elements associated with the image map(s) have text equivalents or a meaningful text description, while others do not.
c. No. None of the non-text elements associated with the image map(s) have text equivalents or meaningful text descriptions.
d. N/A. The page does not include any client-side image maps.

Tables

> **"Row and column headers shall be identified for data tables." 36 C.F.R. § 1194.22(g)**
>
> **"Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers." 36 C.F.R. § 1194.22(h)**

Tables are presentations of information in rows and columns. As shown in Example 6, below, tables are typically used to display numeric or statistical information where numbers occupy *cells* located along specific horizontal rows and under specific vertical columns.

Example 11:

Wages for Summer 2000			
	June	**July**	**August**
Mark		$360.30	$720.59
Judy	$180.15		
Bill	$720.59	$720.59	
Trina	$720.59	$720.59	$720.59

Many web designers also use tables as a way of arranging text into columns or cells, as HTML has traditionally otherwise lacked a way of aligning text in columns.

Orientation of Data in Cells. Tables present unique problems for people with disabilities who use assistive technologies such as screen readers and refreshable Braille displays. In order to make sense of the information in a data cell, the reader must understand the significance of the cell in the overall scheme of the table. A nondisabled user can quickly look at the column and row headings to accomplish this task. Someone who uses certain types of assistive technologies cannot. To fully understand the problem, one must understand how most screen readers (and other types of assistive technologies, such as refreshable Braille displays) work. Screen reading software typically follows standard reading protocols, generally begin to read at the top left of the screen and moving left to right, dropping down 1 row at a time. If someone using a screen reader is confronted with the table in Example 6, he or she would hear the following: "Wages for Summer 2000 June July August Mark $360.30 $720.59 Judy $180.15 Bill $720.59 $720.59 Trina $720.59 $720.59 $720.59." Since some screen readers cannot associate the data with particular

column headings, the reader would be unable to tell, for example, the months in which Mark, Judy, or Bill worked.

Web developers can use several HTML attributes to make most HTML tables accessible to people who use screen readers and other types of assistive technologies. In particular, the "id" and "header" attributes can be used to identify table rows and headers within each cell. Although not all screen readers are currently able to make use of these attributes, more sophisticated screen readers that can do so are being introduced.

<u>Formatting of Text in Tables</u>. Another problem with tables that is frequently encountered by users of screen readers and other types of assistive technologies arises when long text elements such as phrases or sentences are placed within a table for formatting purposes. When the text "wraps" within a table cell, it becomes difficult or impossible for users of some assistive technologies to make sense of it.

Example 12:

The elephants should be fed hay two times a day.	The tigers should be fed raw meat once every three days.	The pandas should be fed bamboo on demand.

An older or simpler screen reader, for example, would ignore the formatting of the table columns and read each line from left to right, then dropping to the next line and reading it left to right, as follows. Reading the table in Example 12, a screen reader would recite: "The The tigers The elephants should be fed pandas should be raw meat should be fed hay two once every fed times a day. three days. bamboo on demand." To avoid this problem, web developers who want to make their pages completely accessible to persons with disabilities should avoid using tables for formatting text. However, the section 508 standards do not prohibit agencies from using tables to format text. Therefore, survey questions 14 and 15, below, do not inquire about the use of tables for formatting text.

<u>Preformatted Tables Using the <PRE> Tag</u>. HTML browsers do not ordinarily render "whitespace" (e.g. spaces, tabs, carriage returns, etc.) in an HTML document as anything other than a single space. Most of the time, this feature makes page layout very easy as web developers can freely insert spaces and extra lines between words and tags without disturbing the ultimate appearance of the page. Occasionally, however, web developers want greater control over spacing in a document and may create tables by simply adding extra spaces between column entries. HTML's <PRE> and </PRE> tags facilitate this process by formatting everything between these tags as preformatted text. Thus, extra spaces will be interpreted as extra spaces. This means that creating tables will not require careful use of <TABLE> tags (and table row and

table cell tags) and may simply require adding extra spaces between <PRE> tags— thus tables are much simpler to create. The disadvantage is that screen readers cannot be told anything about the structure of the table and the tables are visually far less appealing because all table content must appear in monospace font.

```
Example 13:             Wages for Summer 2000
                   June         July        August
        Mark                    $360.30    $720.59
        Judy        $180.15
        Bill        $720.59    $720.59
        Trina       $720.59    $720.59    $720.59
```

While the contents of this table are identical to Example 11, the HTML code supporting the two tables is radically different and demonstrates the relatively simplicity of preformatted tables. While the standard HTML code used to create Example 11 is relatively complex,[8] a preformatted table is much more straightforward.[9]

---

[8]The HTML source for Example 6 would likely be as follow:

```
<table border=1, cellpadding=2, cellspacing=0>
<tr><td colspan=4 align="center">Wages for Summer 2000</td></tr>
<tr><th id="name"> </th><th id="June" align="center">June</th>
<th id="July" align="center">July</th>
<th id="August" align="center">August</th></tr>
<tr><td header="name" align="center">Mark</td><td header="June"> </td>
<td align="center" header="July">$360.30</td>
<td align="center" header="August">$720.59</td></tr>
<tr><td align="center" header="name">Judy</td>
<td header="June" align="center">$180.15</td>
<td header="July" > </td><td header="August"> </td></tr>
<tr><td align="center" header="name">Bill</td>
<td header ="June" align="center">$720.59</td>
<td align="center" header="July">$720.59</td><td header="August"> </td></tr>
<tr><td align="center" header="name">Trina</td>
<td align="center" header="June">$720.59</td>
<td header="July" align="center">$720.59</td>
<td header="August" align="center">$720.59</td></tr>
</table>
```

This table also demonstrates proper use of the *id* and *header* attributes for making each cell correspond to the appropriate header. A screen reader would have no difficulty with this table.

[9]The coding for the preformatted table would appear as follows:

```
<PRE>                    Wages for Summer 2000
              June                July              August
Mark                              $360.30   $720.59
Judy          $180.15
Bill          $720.59   $720.59
Trina         $720.59   $720.59   $720.59

</PRE>
```

Unfortunately, using the <PRE> tag to make tables also makes them completely inaccessible to users of screen readers and other types of assistive technology because there is no indication to the user that the contents are, in fact, in a table. Furthermore, there is no way to meaningfully associate the information in each "cell" with a particular "column". A screen reader encountering the table in Example 8 would recite: "Wages for Summer 2000 June July August Mark $360.30 $720.59 Judy $180.15 Bill $720.59 $720.59 Trina $720.59 $720.59 $720.59." To avoid this problem, web developers should not use the <PRE> tag to create tables.

Identification of Tables in HTML Code. Tables can be identified in a page's HTML source code by the *<TABLE>* tag. Alternatively, tables are sometimes presented in monospaced font using the *<PRE>* tag.

Appendix A contains detailed instructions and a checklist for coding HTML tables to maximize accessibility.
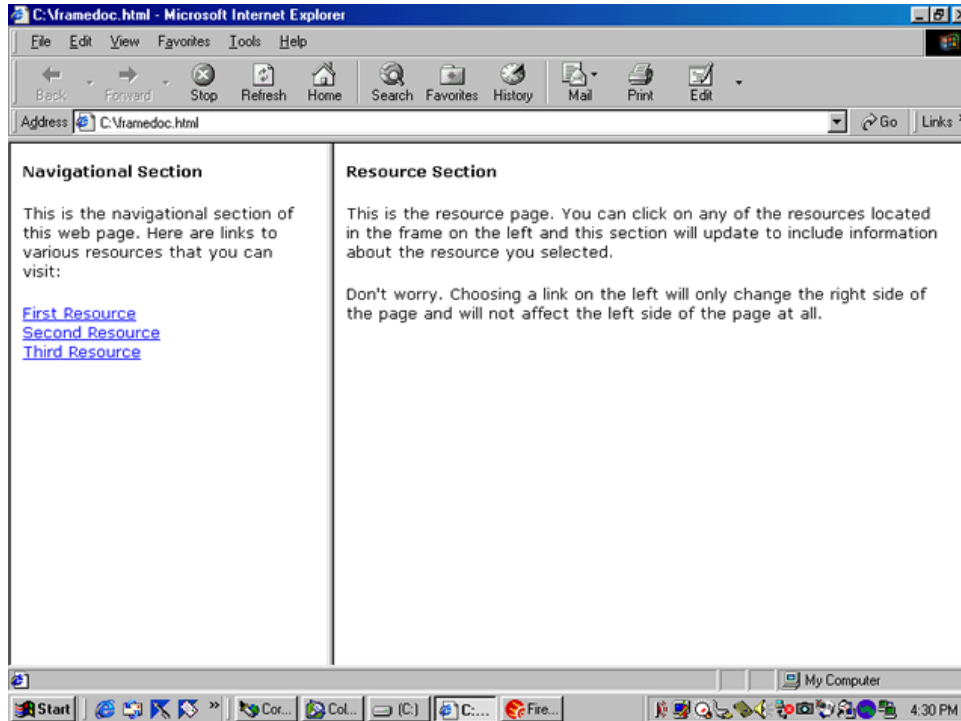
Q14: If the page includes data in tables (either HTML tables or preformatted text tables using the <PRE> tag), and if any of the tables has two or more rows (including header or data cells), does each cell provide identification of row and column headers?

a.   Yes, the page includes tables, some of which have two or more rows, but each cell within the table includes identification of its row and column headers.

b.   No. The page includes tables, at least one of which has two or more rows, but some cells within the table fail to identify row and column headers.

c.   No, but.... The page includes tables, at least one of which has two or more rows, and not every cell within the table identifies row and column headers, but there is meaningful text on the page that conveys the same information as is contained in the table.

d.   N/A. The page does not include any tables or includes tables of single rows only.

Frames

Once a very popular way of formatting information on a browser screen, frames have become less popular as web sites have incorporated new ways of presenting information. Most commonly, frames are used to divide the browser screen into separate areas with each presenting different, but usually related, information.[10]   For instance, as shown in Example 14, a developer may use *framesets* to place navigational links on the left side of the screen ("Navigation Section," in this example) and the substance in a larger frame on the right ("Resource Section," in this example):

---

[10]Less commonly, frames are used to assist scripts. Because the contents of each frame can change independent of the other frames, hiding a script in a so-called "hidden frame" lets a web developer embed a program that works consistently as the user changes from one page to another. Often, hidden frames are not visible on the browser screen but can be found by reviewing the page's HTML source code.

Example 14

In this way, a user can more easily navigate different areas of a web site while keeping track of where he or she is in relation to the overall web site. Frames can also be designed with a newer technology – *inline frames* – that allows for more design control.

To ensure that web pages using frames are accessible to people with disabilities, designers should provide meaningful titles to each of the frames so that users can properly orient themselves. Otherwise, those who use screen readers or other types of assistive technologies will be unable to fully understand the relationships among multiple frames on the same page.

Frames can be identified by examining a page's HTML code. Framesets will be coded with a <FRAME> tag, while inline frames will be coded with an <IFRAME> tag.

Appendix B contains more information about the accessibility implications of frames and detailed instructions for making *framesets* and *inline frames* accessible.

Q15: **If the page uses frames, does each frame have a title that meaningfully describes it?**

a. Yes.
b. No.
c. N/A. The page does not use frames.

Screen Flicker

"Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz." 36 C.F.R. §

Some individuals with photosensitive epilepsy can have a seizure triggered by displays which flicker or flash, particularly if the flash has a high intensity and is within certain frequency ranges. Seizures can be triggered by flickering or flashing in the 2 to 59 flashes per second (Hertz) range. Developers should avoid using strobe light effects – abrupt changes from light to dark – or using flashes in any color range at a rate of 2 to 59 flashes per second. Flashing or

flickering elements are usually added through technologies such as *animated gif's, Java applets,* or third-party *plug-ins or applications*. As discussed above, *Java applets* and *third-party plug-ins* can be identified by the presence of <APPLET> or <OBJECT> tags. *Animated gif's* are images that download in a single file (like ordinary image files), but have content that changes over short periods of time. Like other images, however, they are usually incorporated through the use of the <IMG> tag.

Q16: Does the page include content (such as applets or content requiring plug-ins) that may cause the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz?

a. Yes.
b. No.

Scripts, Plug-Ins, Applets, and other Programmatic Objects

"When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology." 36 C.F.R. § 1194.22(l)

"When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with §1194.21(a) through (l)." 36 C.F.R. § 1194.22(m)

Web sites can present information as *static* – fixed – content, or as *dynamic* content that can be changed depending on conditions set by the web developer. Dynamic content may be generated through the use of plug-ins, scripts, or other programmatic objects.[11] Each of these methods of generating dynamic content poses some accessibility challenges.

Scripts. Scripts are simply sets of computer instructions that are embedded in a web page's HTML coding. They are typically used by web designers to perform relatively rudimentary functions such as verifying the contents of a form or changing images on a screen when a user passes a cursor over the image. They can be easily identified by reviewing a page's HTML source coding for a tag that begins with the word *"script*:"[12]

    Example 15: <SCRIPT language="JavaScript1.2">

In the past, scripts presented significant difficulties for people with disabilities. These barriers have been significantly reduced as more web browsers used by people with disabilities have been updated to support scripts. However, two common uses of scripts continue to cause accessibility problems. Scripts can be used to change an element – such as when a "rollover gif" is activated. A rollover gif changes an image on the screen when the user moves his or her cursor over the image. Such scripts are inaccessible because the script is used to change an element that does not have functional text associated with it. If a script changes the content of a page without somehow indicating (through text readable by a screen reader) that it has changed the content of the page, the script cannot be made accessible.

Second, a script may create a barrier to accessibility if the *event handler* triggering the script cannot be activated through the keyboard. For example, rollover gif's can often only be activated through mouse movement. People with disabilities who cannot use a computer mouse – such as those who are blind or those who are quadriplegic – cannot access the rollover gif. Fortunately,

---

[11]Another type of dynamic web page not addressed in this document is a page whose content has been changed "on the fly" before being sent over the Internet to the user's computer. Such pages appear to the user to be static. An example of this kind of web page is a page that has been created by a search engine.

Yet another type of feature may appear to be dynamic, but is actually static. For instance, "animated gif's" have a fixed content, but present a moving image on the user's monitor. Since the images don't change from one user to another, they are not considered to be dynamic content and should be treated as images for accessibility purposes.

[12]Alternatively, scripts can also be included in the parameters for an HTML tag. These so-called "inline JavaScripts" can be identified by the word "javascript" as part of the contents of a tag. For instance, the following tag invokes a script to open a new window:

        Open a new page

most event handlers have recently become accessible through keyboard commands, due to improvements in the assistive technology used by people with disabilities.

Plug-ins[13] are third-party "add ons" to a computer browser that allow third-party companies to develop special content not normally available using just HTML. For instance, the multimedia and streaming video that are available in many commercial information service web sites use plug-ins to enable web pages to provide "real time" broadcasts of radio and television programs. Because plug-ins are really separate computer software applications created by third-party manufacturers, the challenges created by plug-ins extend beyond the accessibility requirements for web pages. Reviewers evaluating the accessibility of plug-ins should review the Access Board's section 508 rule as it applies to software, 36 C.F.R. § 1194.21 (available at: www.access-board.gov/sec508/508standards.htm).

Plug-ins can usually be detected by examining a page's HTML for the presence of an <OBJECT> tag. Some plug-in manufacturers, however, may require the use of proprietary tags.

---

[13]More properly, the term "plug ins" is limited to Netscape Navigator. Internet Explorer uses a separate technology called ActiveX controls. Both of these technologies permit third-party companies to design program modules that work in conjunction with a web browser to run proprietary programs that can be customized by web developers. Each manufacturer of web browsers uses a slightly different technology to permit third party developers to "add on" to their web browsers. To simplify the discussion, the term "plug in" is used to refer generically to all such technologies.

Adobe Acrobat's Portable Document Format (pdf)

Many agencies have turned to Adobe Acrobat's portable document format (.pdf) to ensure a more controlled presentation of documents posted to their websites than is available with HTML. To view a pdf document, a web browser must have the Adobe Acrobat Reader plug-in loaded onto his or her computer.

HTML is not designed for extremely precise representations of data on a computer screen or printer, but allows for a degree of flexibility that can benefit users with disabilities. For instance, as most browsers permit users to increase the size of the font displayed on their screens, a user with limited vision can often increase the font size and the web page will reformat itself to meet the user's needs. On the other hand, HTML's lack of precision can be frustrating for users or web designers who want web-transmitted documents to look exactly like printed documents. For instance, HTML cannot ensure that fonts, margins, and page breaks will occur at exactly the same location on every computer screen every time a document is viewed. Similarly, if a handwritten signature must always appear at the same location, HTML cannot be used reliably.

Adobe Acrobat's Portable Document Format (.pdf) is a technology that responds to the lack of precision of presentation inherent to HTML. Documents created in pdf will appear identical on *every* computer monitor and on every computer printer. Page breaks, margins, fonts, and even the placement of graphics will be the same as in the original document.

In general, pdf documents can be created in several different ways and each method has separate implications for accessibility:

1. Scanning a document into pdf creates a so-called "PDF Image Only" file that is essentially a graphic representation of the document and, like a photograph with no associated text, is generally unreadable by screen reader technology.

2. Scanning a document into pdf and then running it through OCR ("optical character recognition") technology converts the text images into searchable text. Such documents must be checked carefully for accuracy.

3. Printing a file directly into pdf format converts the electronic information into a digital representation of the document that is somewhat readable to assistive technology.

4. Writing a document inside Adobe Acrobat can also result in a document that is somewhat readable by assistive technology.

Web developers should try to use the last two methods of creating pdf files and should avoid the first method entirely. Agencies should also test the accessibility of their pdf documents using screen readers before posting them to their web sites. Adobe's accessibility site includes includes the latest recommendations for making pdf files accessible (http://access.adobe.com). Finally, agencies should be careful that non-text content be accompanied by text descriptions in pdf files. Agencies that choose to publish web-based documents in pdf should simultaneously publish the same documents in another more accessible format, such as HTML.

Applets. Applets are special programs written in Java, a computer programming language created by Sun Microsystems. Java allows programs to be run on different machines that support

it.[14] Applets can be written by web developers and downloaded with a web page. Most modern browsers include a JVM (Java Virtual Machine) that will run applets. Applets are different from scripts because they are not included in the HTML content of the web page and are partially compiled to run faster on the user's computer.[15] Applets are also different from content developed for plug-ins because they are written in Java and do not require the addition of third-party software to operate. Java is not the same as JavaScript (discussed above).

Unfortunately, many browsers used by people with disabilities do not work with applets. In addition, while Sun Microsystems has incorporated accessibility features in Java, most Java developers do not make use of these features when using applets. In order to make the content of applets accessible, web developers should provide text alternatives.

Like plug-ins, applets can be identified by the presence of an <OBJECT> tag in the HTML source for a web page. Also, an <APPLET> tag may also signal the inclusion of an applet in a web page.

Q17: **If the page uses scripts, such as JavaScript or scripts in Macromedia Flash content, and if the scripts affect any content displayed to the user, is there equivalent text provided by the page or the script that is accessible to a screen reader?**

a. Yes. The page contains JavaScript or Macromedia Flash content that affects the content displayed to the user, but the page or script contains equivalent text that is accessible to a screen reader.

b. No. While the page contains JavaScript or Macromedia Flash content that affects the content displayed to the user, neither the page nor the script contains equivalent text that is accessible to a screen reader.

c. N/A. The page does not use JavaScript or Macromedia Flash content.

---

[14]Ordinarily, computer programs must be written for the specific type of computer (and operating system) on which they will operate. Computer programs written in Java, however, can run on any computer that includes a JVM (Java Virtual Machine).

[15]More specifically, scripts are simply text files that must be read by a computer and internally converted into a set of instructions that can be run by the computer. Applets, on the other hand, are already converted into "computer language" and are not text files. This partial conversion allows applets to run faster than scripts of comparable complexity.

Q18: **If the web page uses applets, such as downloadable Java applets, does it also contain the same information and functionality in an accessible format?**

a.　Yes, while the page uses applets, it contains the same information and functionality in an accessible format.

b.　No, although the page uses applets, it does not contain the same information and functionality in an accessible format.

c.　N/A. The page does not use any applets.

Q19: **If the page uses other programmatic objects (such as Flash, Shockwave, RealAudio, or RealVideo content), or otherwise requires the use of plug-ins or programmatic support for the browser, does the page include a link to the plug-in or programmatic item required for accessing the content of the page and is that plug-in or programmatic item itself accessible to people with disabilities?**

a.　Yes, the page uses such programmatic objects and includes a link to the plug-ins or other items required for accessing the content of the page and those plug-ins or programmatic items are accessible to people with disabilities.

b.　Yes and no. While the page uses such programmatic objects and includes a link to the plug-ins or other items required for accessing the content of the page, those plug-ins or programmatic items are not accessible to people with disabilities.

c.　No, while the page uses such programmatic objects, it does not include a link to the plug-ins or other items required for accessing the content of the page.

d.　N/A. The page does not use such programmatic objects.

Q20: **If the page includes links to .pdf (Adobe Acrobat's portable document format) files, were those .pdf files created in a way that is likely to maximize their usability for people with disabilities?**

a.   Yes. While the page includes links to .pdf files, the files were created either by "printing to .pdf" or by scanning something into .pdf and then running them through an optical character recognition (OCR) process.

b.   No, the page includes links to .pdf files that were created by scanning files into .pdf and were not put through an OCR process.

c.   N/A. The page does not include any links to .pdf files.

Electronic Forms

"When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues." 36 C.F.R. § 1194.22(n)

In addition to *distributing* information, the Internet provides agencies with an incredible resource for *collecting* information. Much of this information collection is done with online forms.

For example, one very popular type of online form is the search engine. Search engines allow users to type their search requests into a box and submits them. The web server receives the information on the form – the search request – and uses it to process an appropriate response.

In addition to search engines, more complex forms are used on many websites. For instance, when shopping online, users may be asked to submit their names, addresses, credit card numbers, and other information.

HTML Forms. In order for an online form rendered in HTML (HTML form) to be accessible, each input element, whether it be a text box, radio button, submission button, or other element must be laid out in a way that makes sense to those who use screen readers and other types of assistive technologies. In addition, the HTML form's instructions must also be presented in an easy-to-use manner. Often times, HTML forms are inaccessible because the text label identifying a particular radio button or other screen element does not appear contiguous to the

element it describes. Someone who uses a screen reader may not be able to discern which of the buttons or elements is the appropriate selection.

In general, elements in HTML forms should appear contiguous to their descriptions. It is difficult to give more specific guidelines, however, because there are so many ways for web designers to format them. The only sure way to check the usability of an HTML form is to test it with assistive technology.

Other Online Forms.   A number of other types of online forms have been developed to overcome the limited control HTML allows over page layout. HTML forms cannot be used where an online form is required to look exactly like its paper counterpart, as the appearance of HTML forms will vary slightly depending upon the user's computer hardware and software. Non-HTML forms designed to be completed and submitted online are used with plug-ins or special third-party software. The accessibility challenges of plug-ins and other programmatic elements is addressed, separately, above.

Q21: **If the page includes one or more electronic forms that is designed for completion online, does each form permit users of assistive technology to access the information, field elements, and functionality required for completion and submission of the form including all directions and cues?**

a.   Yes, the page contains one or more forms designed to be completed online. Each such form complies with all of the accessibility requirements that are the subject of this questionnaire and has been tested for usability by persons using assistive technologies.

b.   Yes, but... The page contains one or more forms designed to be completed online. While each such form complies with all of the accessibility requirements that are the subject of this questionnaire, one or more of them has not been tested for usability by persons using assistive technologies.

c.   Yes, but... The page contains one or more forms designed to be completed online. We have tested each of the forms using assistive technology, but we are not sure that each such form complies with all of the accessibility requirements that are the subject of this questionnaire.

d.   No. The page contains one or more forms designed to be completed online, but at least one of these forms is inaccessible to people with disabilities, at least in one respect.

e.   N/A. The page does not contain any forms designed to be completed online.

Q22: If the page contains one or more forms that is designed to be completed online but that is inaccessible to people with disabilities in some respect, does the page include an alternate accessible form or a link to an alternate accessible form?

a.   Yes. While the page contains one or more forms that is designed to be completed online but that is inaccessible to people with disabilities in some respect, the page includes an alternate accessible form or a link to an alternate accessible form.

b.   No. The page contains one or more forms designed to be completed online but that is inaccessible to people with disabilities in some respect and the page does not contain an alternate accessible form or a link to an alternate accessible form.

c.   N/A. The page does not include one or more forms that is designed to be completed online or it does include such forms, but they are accessible to people with disabilities in all respects.

Skipping Repetitive Navigational Links

"A method shall be provided that permits users to skip repetitive navigation links." 36 C.F.R. § 1194.22(o)

Web developers routinely place a host of routine navigational links at a standard location – often across the top, bottom, or side of a page. If a nondisabled visitor returns to a web page or site and knows that she wants to view the contents of that particular page instead of selecting a navigation link to go to another page, she may simply look past the links and begin reading wherever the desired text is located.   For those who use screen readers or other types of assistive technologies, however, it can be a tedious and time-consuming chore to wait for the assistive technology to work through and announce each of the standard navigational links before getting to the 'good stuff.' In order to alleviate this problem, the section 508 rule requires that when repetitive navigational links are used, there must be a mechanism for users to skip repetitive navigational links.

> **Q23:** **If the page includes navigational links to other web pages within the same website, is there a link allowing users of screen readers to skip over those links?**
>
> a. Yes.
> b. No.
> c. N/A. The page does not contain any navigational links to other web pages within the same website.

Timed Responses

> **"When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required." 36 C.F.R. § 1194.22(p)**

For security reasons, some web pages are designed to "time out" automatically if a user does not take some action within a prescribed time period. For instance, commercial online banking services commonly use this technique to make sure that their customers do not leave sensitive financial information on their computer screens where others may view or tamper with that information – or where hackers can 'grab' the streaming data off the Internet.

Pages that require users to navigate or negotiate elements within a fixed amount of time can present challenges to some people with disabilities.[16] For instance, someone with extremely low vision may be a slower-than-average reader. A page may "time out" before he is able to finish reading it. Likewise, people with disabilities may require additional time to complete a form. Many forms, when they "time out" automatically, also delete whatever data has been entered. The result is that someone with a disability who is slow to enter data cannot complete the form. Designers should ensure that when their pages "time out," they alert the user before the time expires and give him an opportunity to request additional time.

---

[16]Because HTML does not itself support such timed responses, incorporating such timed responses requires the use of other programming elements, such as scripts. The challenges created by scripts and other programming elements are discussed above.

Q24: If the page requires users to respond within a fixed amount of time before the user is "timed out," is the user alerted that he or she will be timed out and given sufficient time to indicate that more time is required before actually being timed out?

a. Yes.
b. Yes and no. While the page warns users that they are about to be timed out, it does not give them an option to extend the length of time that the page will be kept open.
c. No. The page will time out users but does not provide prior warning or the ability to extend the length of time it will be kept open.
d. N/A. The page does not "time out" users, no matter how long a page is kept open.

Text-Only Pages

"A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes." 36 C.F.R. § 1194.22(k)

If a 'mainstream' web page is inaccessible to people with disabilities and cannot be made accessible, the Access Board's section 508 standards require that the agency maintain an equivalent text-only web page. Text-only alternatives to inaccessible pages allow people with disabilities to have full access to all information available to others. Such pages must contain equivalent information or functionality as the primary pages and must be updated as often.

Q25: **Taking into consideration your responses to the previous questions, if the reviewed page likely contains barriers to access for people with disabilities, do you have an alternative text-only page that contains the same information and is updated as often as the reviewed page?**

a.   Yes.

b.   Yes and no. While the page appears to contain barriers to access for people with disabilities and we have established a text-only alternate page, the text-only alternate page does not include the same information or is not updated as often as the reviewed page.

c.   No. While the page appears to contain barriers to access for people with disabilities, we do not have an alternate text-only page.

d.   N/A. The page does not appear to contain any barriers for people with disabilities, so we do not have an alternate text-only page.

e.   N/A. Even though the page does not appear to contain any barriers to access for people with disabilities, we still maintain an alternate text-only page that contains the same information and is updated as often as the reviewed page.

Subjective Evaluations

To determine whether your web pages are accessible to people with disabilities, you should have someone who is familiar with assistive technologies such as screen readers use those technologies to test your pages. Alternatively, developers may wish to view their web pages using a text-only web browser, such as the public domain browser Lynx (available at www.lynx.browser.org). These methods may uncover previously unrecognized barriers to access, such as text labels that do not sufficiently describe an image, or page layout that results in incomprehensible text when processed through a screen reader.

Many agencies have employees who use such technologies and who can assist with this testing. Those agencies that do not currently employ anyone who uses assistive technologies may want to seek testing assistance from a local Independent Living Center or other disability rights group.

Q26: **Test this page for accessibility to persons with disabilities using assistive technology (including a screen reader, the Lynx public-domain text-only browser, or the IBM Home Page Reader). Describe the results of this test:**

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Q27: **Describe the accessibility successes and challenges you identified during your evaluation of this page and any plans you have for addressing any problems on this and similar pages within your website:**

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Appendix A

<u>Making Web-Based Tables Accessible</u>

This document provides a quick checklist for making information in tables or charts accessible to users with disabilities. Following the checklist is a discussion of the accessibility issues surrounding tables. This document concludes with a discussion of specific HTML tags and attributes that should be used for making documents accessible and an example of an accessible table.

<u>Discussion of Problem</u>

Electronic information poses special difficulties for persons with disabilities. For instance, blind users and users with visual impairments may be unable to read or have difficulty with reading information displayed on a computer monitor.

To use computer information, people with visual impairments often use screen reading software. Screen readers "read" out loud the information displayed on a computer screen. Similarly, refreshable Braille display devices translate the visual display into a touch-discernable Braille output. Screen readers and Braille display devices have little difficulty with unformatted text on web pages.

Where the formatting of text conveys special meaning, screen readers may encounter difficulties, unless special care is taken by the web page designer. Web tables create several serious problems for screen readers, as described below.

<u>Problems with Data Tables</u>

Tables present unique problems for users with disabilities. In a table, the column and row that a particular cell corresponds to is very important to understanding the information in that cell. All users find that very large tables are confusing; it is too easy to become "lost" in a table when column and row headings become obscured. This problem is greatly compounded for screen reader users because it may be nearly impossible to meaningfully associate row and column information within a particular cell in anything but the smallest tables.

To illustrate this problem, imagine having a large table with 10 columns and 10 rows. In each of the 100 cells of the table are statistics that are aligned under particular column headings and across from particular row headings. A sighted user reading this table could easily associate any cell with its related column headings. A visually impaired user using a screen reader would have a very different experience. Because a screen reader literally reads just the contents of each cell from top to bottom and left to right, it would not take long before the meaning of each cell would be lost. Overcoming this problem calls for a way to indicate with which column heading and row heading a particular cell is associated.

Several HTML attributes make this association of headings and cells possible. Web developers should use the "id" and "header" attributes for table cells and headers. Although not all screen readers and web browsers can "read" these attributes, future browsers and screen readers will likely be able to use them.

Problems with Formatting Text in Tables

Phrases, sentences, and other long textual elements that "wrap" in table cells also pose special problems for users of older screen readers. Older screen readers may read the first line of each adjacent cell before moving to the next row of text in each cell. For instance, if text in two cells are arranged as follows,

This is text in column one.   This is text in column two.
This column includes   This column includes
information relating to   information relating to
column one.   column two.

Then, the text may be read by an older screen reader as:

This is text in column one. This is text in column two. This column includes This column includes information relating to information relating to column one. column two.

Someone listening to this recitation would not be able to make sense of it. Therefore, special care should be taken when using tables for formatting text or any tables that contain text that wraps within a single cell.

Problems Using Preformatted Tables with the <PRE> Tag

HTML browsers do not ordinarily render "whitespace" (e.g. spaces, tabs, carriage returns, etc.) in an HTML document as anything other than a single space. Most of the time, this makes page layout very easy. Occasionally, however, web developers want easy control over spacing in a document and may create tables by simply adding extra spaces between column entries. HTML's <PRE> and </PRE> tags facilitate this process by formatting everything between these tags as preformatted text. Thus, extra spaces will be interpreted as extra spaces.

Unfortunately, using the <PRE> tag to make tables also makes them completely inaccessible to screen readers because there is no indication to the user that the contents are, in fact, in a table. Furthermore, there is no way to meaningfully associate the information in each "cell" with a particular "column". Therefore, web developers should NOT use the <PRE> tag to create tables under any circumstances.

<u>Always Check Your Pages With Assistive Technology Prior to Publication</u>

Before publishing your web pages to the Internet, if at all possible, check your web pages using assistive technology to be sure that they render properly. Remember that users with disabilities may have a very different experience from other users when encountering complex elements like tables. Therefore, web pages with tables deserve extra attention prior to publication.

<u>HTML Table Attributes</u>

Given the special difficulties created by tables, their use should be reserved for situations where they are absolutely necessary. The HTML source for a table includes the following elements, arranged in order of containment:

Table HTML Tags: <TABLE>....</TABLE>
Tag Description: The opening <TABLE> tag and closing </TABLE> tag delineate the beginning and end of a table. Everything between these tags is included in the table.

Row HTML Tags: <TR>....</TR>
Tag Description: The opening <TR> tag and closing </TR> tag mark the boundaries of a row in a table. Everything between these tags is included in a particular row of a table. These tags must be contained within the <TABLE> and </TABLE> tags, above.

Cell HTML Tags: <TD>....</TD>
Tag Description: The opening <TD> tag and closing </TD> tag mark the beginning and end, respectively, of a cell in a table. Ordinarily, these tags are always included in a particular row, delineated by <TR> and </TR> tags, above.

Header HTML Tags: <TH>....</TH>
Tag Description: These tags are best thought of as a special type of <TD> and </TD> tags. The <TH> and </TH> tags mark the beginning and end of a "header" cell.

The <TABLE> tag supports a special attribute called "summary" that is used to provide information about the contents of a table to a non-visual browser.[17] This HTML 4.0 specification may not be supported by all browsers at this time. However, as browser software incorporates more HTML 4.0 specifications, this attribute will serve a valuable function for users with disabilities.

There are special attributes of the <TH> and <TD> tags that allow screen readers to meaningfully associate a particular cell's contents with the header column that it belongs to. These attributes are also part of the HTML 4.0 specification and may not be supported by all browsers. These attributes,[18] and a discussion of how to use them, follows:

Attribute:	id
Used with Tag:	<TH>
Description:	Use the "id" attribute to "name" a particular header column.

Attribute:	headers
Used with Tag:	<TD>
Description:	Use the "headers" attribute in a particular cell to associate the contents of that cell with the "id" used in the associated header cell. Then, the screen reader may voice the contents of the associated header cell, followed by the contents of the cell marked with the "headers" attribute.

---

[17]The Web Accessibility Initiative (WAI) Guidelines of the World Wide Web Consortium (W3C) – which differ in important respects from the Access Board's section 508 rule – also recommend use of the "caption" attribute for each table, which also provides a general caption before the table. Given the limited support for this attribute, however, web developers should be cautious about using this tag to caption their tables.

[18]The W3C's WAI Guidelines also recommend use of the "axis" attribute of the <TH> and <TD> tags. This attribute is used to establish potentially complex relationships between cells of a table. While this is an exciting new development, the "axis" attribute is not supported by most browsers and should not be relied upon by web developers as a means of making tables accessible. A related attribute, called the "scope" attribute, also associates information in a particular cell with other columns, groups of columns, rows, or cells. We believe, however, that this attribute is also not widely supported.

For instance, the following HTML source file for a table may look like:

```
<TABLE summary="The following courses are offered in the mathematics department">

<TR>
<TH id="header1">Course</TH>
<TH id="header2">Professor</TH>
<TH id="header3">Credits</TH>
</TR>

<TR>
<TD headers="header1">Differential Geometry</TD>
<TD headers="header2">Smith</TD>
<TD headers="header3">4</TD>
</TR>

<TR>
<TD headers="header1">Algebraic Topology</TD>
<TD headers="header2">Green</TD>
<TD headers="header3">3</TD>
</TR>

<TR>
<TD headers="header1">Algebraic Geometry</TD>
<TD headers="header2">Thomas</TD>
<TD headers="header3">2</TD>
</TR>

</TABLE>
```

To a popular browser, this may be rendered as:

Course	Professor	Credits
Differential Geometry	Smith	4
Algebraic Topology	Green	3
Algebraic Geometry	Thomas	2

The speech output from a screen reader rendering this table may be equally comprehensible:

> Summary: The following courses are offered in the mathematics department
> Course: Differential Geometry, Professor: Smith, Credits: 4
> Course: Algebraic Topology, Professor: Green, Credits: 3
> Course: Algebraic Geometry, Professor: Thomas, Credits: 2

<u>Table Accessibility Checklist for Use with Dept. of Justice Survey</u>

☐ Avoid using tables simply for formatting paragraphs of text. Some screen readers have trouble reading table cells that include multi-line text.

☐ Do not use the <PRE> tag for laying out tables.

☐ Use the <TABLE> tag's "summary" attribute to include a meaningful description of the contents of the table. This attribute will likely be supported by browsers in the future.

☐ Use the following attributes consistently for headers and data cells in the table:

   ☐ Use the "id" attribute of the <TH> tag to identify columns. This attribute will likely be supported by browsers in the future.

   ☐ Use the "headers" attribute of the <TD> tag to link cells to those columns. This attribute will likely be supported by browsers in the future.

☐ Check your web page using a screen reader prior to publication. If you are unable to do so, read the contents of the table out loud to someone who has not read it and is not looking at the screen. Read the table as you would a book, starting at the upper left corner, reading each line in its entirety before going to the next line. If they can understand the table, it is likely that it is accessible.

Appendix B

Making Web-Based Frames Accessible

Several years ago, Netscape introduced support for frames and, almost immediately, commercial and private web developers revamped their web sites to take advantage of the new technology. Since then, frames have become a standard part of HTML, though frames are less commonly used on commercial websites.

Frames let a web developer divide the browser's screen into several different areas, each displaying a different web page. Each of the different frames can act independently of other frames or can be designed to control the contents of other frames. For instance, a common use of frames is putting navigational links in one frame and the contents of those links in another frame. Using this method enables a user to scroll through a list of possible navigational destinations and choose different options without disturbing the list of navigational links. Every time the user chooses a different link, the contents of a different frame are refreshed.

Like other web technologies, frames present both a challenge and an opportunity for accessibility. The difficulties created by frames most directly affects those who use screen readers, because it may be very difficult for a screen reader user to determine from which frame the screen reader is reading. If frames are properly labeled, however, they may actually improve accessibility by creating an easy-to-reach area which is specifically designed for navigating a website and which is unaffected by visiting different areas on a website.[19]

Identifying whether a web page uses frames and whether it includes features to make those web pages accessible involves some understanding of how frames are implemented. In designing a web page that uses frames, a web designer must either use a *frameset* or an *inline frame.* By far, ordinary *framesets* are more common, because they are supported by all frame-compatible browsers. However, *inline frames* offer a greater level of flexibility and may be used because of a web designer's formatting needs.

Designing a web page using a *frameset* involves two types of web pages. First, there is a web page that serves as a container for all of the other frames. HTML has very specific rules that limit the content of this container document (called the *frameset*). The frameset usually only contains tags identifying how each frame should be laid out and the content of those frames. Second, each of the separate frames within the frameset is actually a distinct web page. The content of those individual web pages is what the user will see on his or her computer screen. HTML is much more liberal in what is permitted within the non-container web pages. For

---

[19]At one time, frames created significant difficulties for screen readers and users with disabilities, because only newer, inaccessible browsers supported frames. Because frames have become an accepted part of websites and because of advances in technology, however, almost all browsers and screen readers now support frames.
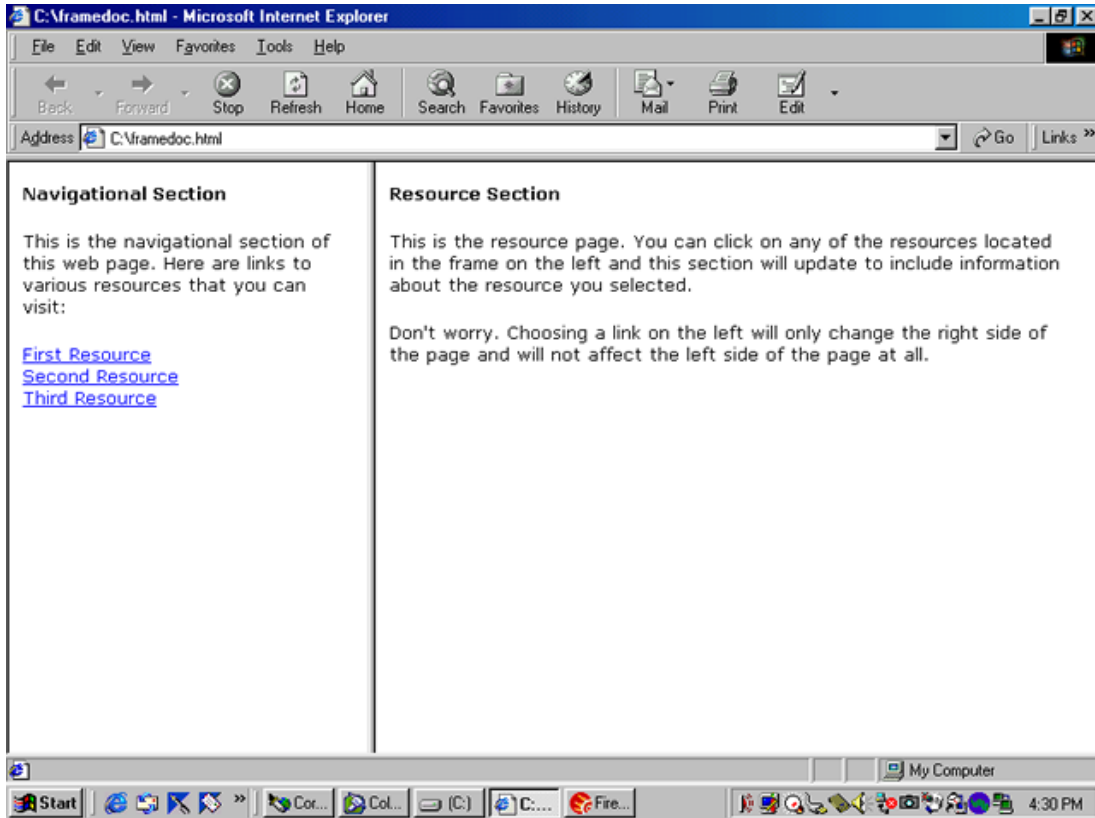
instance, suppose that a web designer is designing a page using a frameset that will direct users to resources. He may design the page so that on the left side will be a frame with navigational links and on the right side will be a frame that displays the ultimate content from each those links with a default frame containing general information. The web designer can then create one file that includes just the navigational links and call it "navlinks.html" and create another file that includes the general information called "geninfo.html".[20] Then, constructing a single web page that includes the contents of these files as separate frames would include only the following frameset:

```
<FRAMESET COLS="30%, 60%">
<FRAME SRC="navlinks.html" NAME="navlinks">
<FRAME SRC="geninfo.html" NAME="contents_page">
</FRAMESET>
```

Depending on the actual contents of the *navlinks.html* and *geninfo.html* files, the resulting frameset web page may look like:

---

[20]This example also illustrates the use of the *name* attribute that is used in each *<FRAME>* tag. The purpose of the *name* attribute is to provide an easy way of identifying the different regions (frames) of the browser window – a critical step when the web page links to new content.

Because the HTML standards require web pages using frames to have this format, determining whether a web page uses a frameset requires looking for either the <FRAMESET> or <FRAME> tags. Making the frame portion of these web pages accessible is extremely easy. Both the <FRAMESET> and <FRAME> tags support an attribute called *title* that can be used to describe the contents of the frameset or frame. For instance, if a designer wanted screen reader users to know that a page uses frames and that one frame includes navigational links and the other frame displays the content of those links, the web designer would only have to make the bold-faced additions to the frameset document:

<FRAMESET COLS=”30%, 60%” **TITLE=”document about national parks using frames”**>

<FRAME SRC=”navlinks.html” NAME=”navlinks” **TITLE=”navigational links section”**>
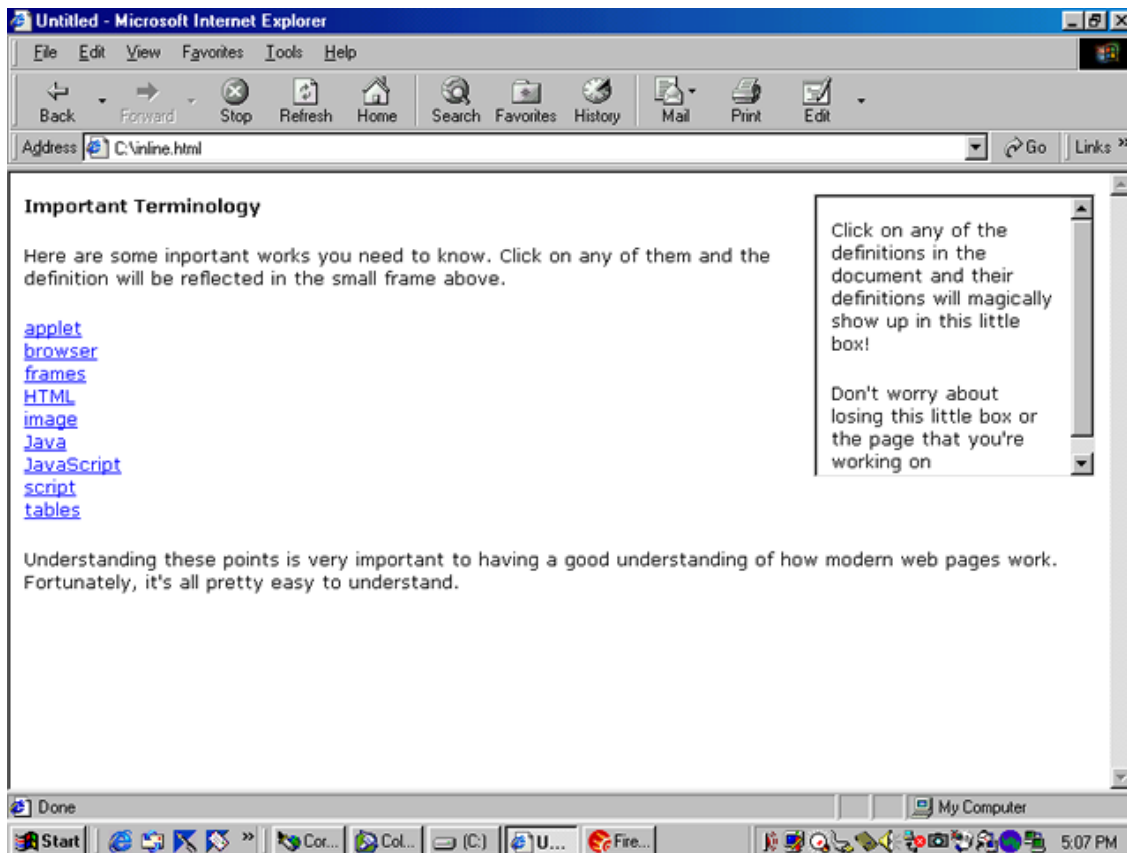
<FRAME SRC=”geninfo.html” NAME=”contents_page” **TITLE=”contents section”**>

</FRAMESET>

Inline frames technology is a new way to offer greater flexibility in web page design. Unlike a *frameset* web page that divides up the entire screen into a separate regions, inline frame technology enables web designers to place frames of any size at any location on the screen. Creating an inline frame involves nothing more than including an *<IFRAME>* tag within the content of a regular web page. There is no need for a *<FRAMESET>* tag, because the entire web page is effectively the container for the frame, just as a web page can be a container for images or other content. An *<IFRAME>* tag can be very simple:

<IFRAME WIDTH=200, HEIGHT=200, SRC="info.html" ALIGN="right">

Such a tag can be used for creative web design. For instance, the following page uses a tiny frame in the upper right hand corner to display the definitions of words selected from the main document by the user:

Making this inline frame accessible is as simple as making a frameset web page accessible. Like the *<FRAME>* tag, the *<IFRAME>* tag also supports the *title* attribute that can be used to pass information to a screen reader about the contents of that frame. For instance, if the web designer wanted to identify the frame as including definitions for terms selected in the main document, the web designer would only need to modify the *<IFRAME>* tag with the boldfaced language:

```
<IFRAME WIDTH=200, HEIGHT=200, SRC="info.html" ALIGN="right"
TITLE="definitions section">
```

Because *inline frame* technology is a relatively recent addition to the HTML standards, it is only supported by the very latest browsers.