

Simplified Web Accessibility Guide



This publication is the result of a partnership between:



Human Resources
Development Canada

Développement des
ressources humaines Canada



BRITISH
COLUMBIA

Ministry of
Advanced Education



CENTRE FOR CURRICULUM,
TRANSFER & TECHNOLOGY

Office For
**Disability
Issues**

Simplified Web Accessibility Guide

Glenda Watson Hyatt
Soaring Eagle Communications

National Library of Canada Cataloguing in Publication Data

Watson Hyatt, Glenda.

Simplified web accessibility guide

ISBN 0-7726-4682-1

1. Web site development. 2. Computers and the handicapped. I. Centre for Curriculum, Transfer and Technology.

HV1569.5.W37 2001

005.2'76

C2001-960314-2

To Order Copies of this Publication:

Marketing Department
Open Learning Agency
4355 Mathissi Place
Burnaby B.C. V5G 4S8
Telephone: 604-431-3210
Toll Free: 1-800-663-1653
Fax: 604-431-3381
Email: catalogue@ola.bc.ca
Web: <http://www.ola.bc.ca>

Simplified Web Accessibility Guide

Order Number: MN1935
ISBN 0-7726-4682-1

For More on Career and Labour Market Information:

Ministry of Advanced Education
St. Ann's Academy
835 Humboldt Street
PO Box 9888, Stn Prov Govt
Victoria BC V8W 9T6
Telephone: 1-888-664-2256
Web: <http://www.aett.gov.bc.ca>

Human Resources Development
Canada BC/Yukon Region
Library Square, Suite 1400
300 West Georgia Street
Vancouver BC V6B 6G3
Telephone: 1-800-661-1997
Web: <http://www.bc.hrdc-drhc.gc.ca>

Office for Disability Issues
PO Box 9567 Stn Prov Govt
Victoria, British Columbia V8W 9K1
TDD: 250-387-3555
Fax: 250-387-3114
Telephone: 250-387-3813
E-mail: mhrodi@gems4.gov.bc.ca
Web: <http://www.gov.bc.ca/odi/>

Centre for Curriculum, Transfer and
Technology
6th Floor, 1483 Douglas Street
Victoria, B.C. V8W 3K4
Telephone: 250-413-4409
Fax: 250-413-4403
Email: curriculum@c2t2.ca
Web: <http://www.c2t2.ca/>

**Copyright © 2002 Ministry of Advanced Education and
Human Resources Development Canada BC/Yukon Region**

With appropriate acknowledgement, text in this document may be reproduced for non-commercial use without permission. Reproduction for any other purpose requires permission from the Centre for Curriculum, Transfer and Technology.

Contents

Acknowledgements	ii
Foreword	iv
Why Web accessibility?	1
Does it cost more to build an accessible site?	3
What are the Web accessibility principles?	4
How to use this guide?	6
Question 1: Does the Web site contain auditory information and visual images?.....	7
Question 2: Does the site rely on colour alone to convey information?	11
Question 3: Are markup and style sheets used properly?	13
Question 4: Does the Web page clarify natural language usage?.....	16
Question 5: Are tables understandable on various systems?.....	18
Question 6: Are pages featuring new technologies understandable on older systems?	23
Question 7: Can users control scrolling or changing content?	27
Question 8: Are embedded user interfaces compatible with assistive technologies? ..	30
Question 9: Is the page designed for a variety of input devices?	32
Question 10: Are interim accessibility solutions used to ensure that older assistive technology and browsers will operate correctly?.....	35
Question 11: Are current W3C technologies and guidelines used?.....	39
Question 12: Does the page provide context and orientation information?	43
Question 13: Does the site provide clear navigation mechanisms?	46
Question 14: Is the site's content written in clear and simple language?	51
Question 15: Are PDF files only used as an alternate format?.....	53
Appendix A: Summary of Web Accessibility Standards	55
Glossary	59
References	61
Index	62

Acknowledgements

The Simplified Web Accessibility Guide was jointly sponsored and funded by Human Resources Development Canada: BC/Yukon Region, the BC Ministry of Advanced Education, and the BC Ministry of Human Resources: Office for Disability Issues. Project management was provided by the Centre for Curriculum, Transfer and Technology.

The Guide was researched and written by Glenda Watson Hyatt, the principal of Soaring Eagle Communications. Ms. Watson Hyatt is an expert on, and specializes in addressing electronic and online access issues for persons with disabilities. Desktop publishing was provided by Bev Lytton, of Superscript.

We wish to thank the other members of the management and steering committee— Jim Howie from Ministry of Advanced Education, Angie Allard from Human Resources Development Canada, Steve Reid and Sue Buttnor from the Office for Disability Issues, and Keith Dunbar from the Centre for Curriculum Transfer and Technology— for their ongoing support for the development of this Guide.

As well, a special note of appreciation to proponents of the original Accessible LMI Project: Dave Symington, Office for Disability Issues; Liam Arthurs, Ministry of Education; Ayne Meiklem, A.M. Research Services; the GF Strong Rehabilitation Centre; the BC Educational Association of Disabled Students; and the late Larry Warren, Human Resources Development Canada, without whom this project would not have been possible.

Comments about the contents of this publication and suggestions for improving it are welcome. Please address them to:

Frank Jonassen, Director
Office for Disability Issues
Ministry of Human Resources
frank.jonassen@gems1.gov.bc.ca

Doug Moulton, Manager, Strategic Services
Strategic Services and External Relations
Human Resources Development Canada
doug.moulton@hrdc-drhc.gc.ca

Foreword

In the new knowledge economy, timely access to quality information is an essential requirement for effective planning and decision making. This is particularly true in the case of the labour market, where both organizations and individuals are making human capital investment decisions based on information regarding current and future labour market trends and developments; and, increasingly, both information providers and users are relying on the Internet as the primary vehicle for exchanging that information.

Increasing dependence on the Internet as a medium of information and other forms of exchange has brought to the fore a wide range of information quality and related access issues that need to be addressed. Now that the Internet is a primary source of information, the quality and timeliness of that information becomes a major concern for the consumer. As well, access to the Internet for the widest possible number of user groups, including persons with disabilities, becomes an essential requirement for effective research.

This reality has long been recognized by the World Wide Web Consortium (W3C), the voluntary organization that develops and maintains the common standards and protocols without which the Internet could not function. Through their Web Accessibility Initiative (WAI) the W3C has worked hard to develop the Web Content Accessibility Guidelines (WCAG) that directly address the issue of Internet access for persons with disabilities.

In addition, both government and business are moving forward to ensure that access issues are clearly addressed from their various perspectives. In the United States, implementation of the new Section 508 of the U.S. Rehabilitation Act will require that all content and programs on federal Web sites be accessible to persons with disabilities. Section 21 of the UK Disability Discrimination Act of 1999 includes a similar set of provisions, and expectations are that the European Union will be moving in the same direction in the near future.

Closer to home, the Treasury Board of Canada, under its Common Look and Feel Standards and Guidelines for the Internet, has adopted the W3C Web Accessibility Guidelines Priority 1 and Priority 2 checkpoints “to ensure that (Government of Canada) sites can be easily accessed by the widest possible audience.” And, at the time of writing this publication, the Province of British Columbia is also revising its BC Government Internet Standards to be “in alignment with the W3C Web Content Accessibility Guidelines so as to better address issues and mitigate barriers on government Web sites.”

As well, under the aegis of the Forum of Labour Market Ministers, a working group addressing the development and delivery of labour market information across Canada has highlighted guidelines and standards for labour market information products and services, including accessibility considerations, as a critical component of an enhanced pan-Canadian labour market information “system.”

While the term “accessible Web sites” has in the past been interpreted to mean going to “text-only” versions of the more substantial, graphical mainstream Web pages, the W3C, as well as major software developers such as Macromedia, Microsoft and Adobe, are continuing to work to enable application of accessibility design criteria to development of mainstream Web sites, and to highlight the benefits flowing from a greater focus on ease of access to content on the Internet. An accessible Web site that is simple to navigate and contains well organized, clearly labeled content that can be easily indexed by search engines is simply a well-designed Web site that contributes to more effective information transfer and to increased traffic.

For these reasons, the Simplified Web Accessibility Guide is a timely and valuable addition to the literature on building well-designed, accessible Web sites for all Internet users. Presented in simple question and answer format, this publication provides a clear description of the W3C accessibility standards and serves as a simple diagnostic tool to assist Web developers to review the accessibility of their site and to identify improvements that can be made in this regard.

The contents of the Guide are a credit to the knowledge, skills and experience of Glenda Watson Hyatt, whose simple, clear descriptions of the WCAG checkpoints—what they mean, why they are important and how to address them—are a significant contribution to the resources available to help developers produce Web sites that are easier to access by people with a range of disabilities.

The Simplified Web Accessibility Guide follows up from one of the recommendations of the earlier Accessible LMI Project, funded through the Canada/BC Joint Committee for Enhanced Labour Market Information, and is one of a number of jointly developed products and services designed to help end-users in British Columbia make more informed and effective career, learning, and labour market plans and choices.

In helping to build better designed, more functional, and more accessible career, learning and labour market information Web sites, the Simplified Web Accessibility Guide is making a significant contribution to this goal.

Jim Howie
Ministry of Advanced Education
Victoria, January 2002

Why Web accessibility?

Good Web design makes information accessible. By the same token, Web designers must be aware of accessibility issues in order to accommodate people with disabilities. People with disabilities use the Web for the same things others do—education, employment, entertainment, shopping, and banking. In fact, they may rely upon the Web more than others because of their disabilities. The Web has opened many doors for people with disabilities but bad design can slam the door shut again!

People with disabilities can use a range of assistive technologies (specialized software and hardware) with their computers to help them access information. Unfortunately, poorly designed and coded Web sites cause assistive technologies to perform poorly.

Some obstacles that people with disabilities face when surfing the Internet include:

- People with visual impairments and some types of learning disabilities often rely on text-to-speech **screen readers** that read aloud text appearing on the screen. Screen readers cannot read images (graphs, maps, etc.), so information provided in only these formats is not accessible to these individuals. Blinking and scrolling text can also cause problems for screen readers.
- Flickering or flashing designs can cause seizures in people with certain neurological disorders.
- Without captioning, people with hearing impairments cannot appreciate multimedia content such as on-line newscasts, movies, and lectures.
- Without descriptive narration, individuals who are blind miss information portrayed visually.
- For individuals with little or no hand control, using a mouse can be very difficult. Being required to “click” on a tiny area to access information can be an obstacle.
- Inconsistent page layout and poor information design can be disorienting and confusing to any user.

Not adhering to Web accessibility principles excludes segments of the population. For example, individuals with disabilities need equal access to the labour market information provided online. Canadians with disabilities may have legitimate complaints under the Canadian Human Rights Code if denied equitable service

compared with other Canadians. Accessible Web design will provide equal access to the information and opportunities on the Internet.

In addition to making information easier to access and thereby increasing the site's potential customer/client base, benefits of accessible Web design include:

- Improved usability for all visitors. Consistent navigation makes it easier to find desired content quickly.
- Clear navigation and clear content supports people with low literacy levels.
- Captioning is useful for people with hearing impairments, for people in noisy environments, and for those having difficulty understanding the spoken word due to an accent or foreign language.
- Good colour contrast aids people with colour blindness, people using monochrome monitors, and those who prefer to read from printed pages.
- Providing **text equivalents** (e.g., ALT attributes and captioning), table summaries, and metadata improves search engine listings.
- By using style sheets, colour independence, and avoiding **deprecated elements**, Web content will be more readily available to a changing (and expanding) customer/client base and to any new technologies.
- Web content in alternate formats assists low-bandwidth users.
- Reduced site maintenance. By separating structure and presentation and using style sheets, the look of a site can easily be changed by modifying only one file.
- Accessibility solutions, such as cascading style sheets, can increase the speed of file transfer, thereby reducing server load.
- Keyboard commands assist people with limited hand function and those working in confined spaces with little or no room to operate a mouse (e.g., on planes and in cars).

Good design is good design. Just as sidewalk curb cuts—originally intended for people using wheelchairs—also benefit parents wheeling strollers and individuals on roller blades, accessible Web design benefits more than just people with disabilities. Accessibility and usability are intertwined and are equally important.

Note: Parts of this section are excerpted and adapted from contributions by the author to a report by the Office for Disability Issues on the subject of improving access to government web sites. To view or download and print a copy of that report, see... <http://www.gov.bc.ca/odi/rpts/>

Does it cost more to build an accessible site?

The main impediment to making a site accessible is the perceived cost premium. A common misconception is that by adopting a Web accessibility policy the cost of designing and maintaining a site will dramatically increase. Typically, however, costs are not as high as first anticipated, and although some additional costs might be incurred, the major investment is time. Web sites vary in size and complexity, making it difficult to give an exact cost for accessibility features. Cost factors include:

- When the accessibility policy is implemented, whether it is for a new Web site or to retrofit an existing site. Incorporating accessibility features into the development phase of a new site will be less expensive than adding the features to an existing one.
- Design complexity. Accessibility does not mean sites need to be boring or text-only. For new sites that are more complex in design, accessibility may add relatively little to the cost compared to the overall cost of development. It may even create some longer-term savings in site maintenance and updating.
- Web architecture. Existing Web sites using valid World Wide Web Consortium (W3C) technologies may require relatively little expense to retrofit. Existing Web sites that use invalid markup or proprietary technologies may require more extensive retrofitting and therefore may incur more expense.

Other costs may include:

- Assessing the accessibility of an existing Web site. The initial assessment can be conducted by existing personnel or by external consultants. (Use this Guide as a diagnostic and assessment tool.)
- Training. Personnel may require additional training in HTML, style sheets, etc., particularly with smaller organizations in which Web developers also have other duties.
- Software. Consider purchasing Web design software that better supports the production of accessible Web sites.
- Assistive technologies. Large organizations doing significant amounts of Web developing may want to purchase assistive technologies to test accessibility of sites. Alternatively, include people with disabilities in user test groups.

What are the Web accessibility principles?

To provide a comprehensive look at Web accessibility principles, this Guide covers three sets of guidelines/standards:

1. The Web Accessibility Initiative's "Web Content Accessibility Guidelines 1.0,"
2. The Treasury Board of Canada's "Common Look and Feel Standards and Guidelines for the Internet," and
3. The Province of British Columbia's "B.C. Government Internet Standards."

Web Accessibility Initiative

The Web Accessibility Initiative (WAI) is responsible for implementing the World Wide Web Consortium's (W3C) commitment to making the Web accessible to all people. The WAI is a recognized authority for the development of Web accessibility guidelines. The Web Content Accessibility Guidelines 1.0 (WCAG 1.0) are internationally accepted and many standards and policies are derived from these guidelines.

WCAG 1.0 consists of 14 guidelines, presented in this publication as *Questions*, and checkpoints, presented here as *Solutions*, which are sub-sections explaining how to implement each guideline. Each checkpoint has a priority level assigned based on the checkpoint's impact on accessibility.

Priority
1

Priority 1. A Web content developer *must* satisfy this checkpoint. Otherwise, one or more groups will find it impossible to access information in the document. Satisfying this checkpoint is a basic requirement for some groups to be able to use Web documents.

Priority
2

Priority 2. A Web content developer *should* satisfy this checkpoint. Otherwise, one or more groups will find it difficult to access information in the document. Satisfying this checkpoint will remove significant barriers to accessing Web documents.

Priority
3

Priority 3. A Web content developer *may* address this checkpoint. Otherwise, one or more groups will find it somewhat difficult to access information in the document. Satisfying this checkpoint will improve access to Web documents.

[From WCAG 1.0 <http://www.w3.org/TR/WAI-WEBCONTENT/#priorities>]

Treasury Board of Canada

Under its “Common Look and Feel (CLF) Standards and Guidelines for the Internet” (<http://www.cio-dpi.gc.ca/clf-upe/ae.asp>), the Treasury Board of Canada has adopted the W3C Web Accessibility Guidelines Priority 1 and Priority 2 checkpoints “to ensure sites can be easily accessed by the widest possible audience.” The CLF standards also address additional accessibility issues not covered by the W3C. These additional standards are included in their guide.

Federal institutions listed in Schedule 1, 1.1 and 2 of the Financial Administration Act (<http://laws.justice.gc.ca/en/F-11/index.html>) must comply with the CLF Standards by December 31, 2002.

Province of British Columbia

At the time of writing this guide, the Province is revising its “B.C. Government Internet Standards” to be “in alignment with the W3C Web Content Accessibility Guidelines (*complying with Priority 1 and considering Priority 2 checkpoints*) so as to better address issues and mitigate barriers on government Web sites.”

The standards apply to all government Internet sites, and all agencies, boards and commissions of the B.C. government that fall under the Financial Administration Act (http://www.qp.gov.bc.ca/statreg/stat/F/96138_01.htm). Crown corporations may choose to comply.

How to use this guide?

This Guide explains the Web Content Accessibility Guidelines 1.0 in simplified detail, with additional material included where relevant. Presenting content in a question and answer format, the Guide is designed as a diagnostic tool to assist Web developers in assessing the accessibility of Web sites. The glossary defines terms that are in ***bold italics***.

Questions 1 through 14 correspond to Guidelines 1 through 14 of the WCAG 1.0; and solutions correspond to the WCAG checkpoints. Each solution indicates the WCAG priority, based on the impact on accessibility. Both the Treasury Board and the Province of B.C. require compliance with Priority 1 and Priority 2.

Appendix A summarizes the questions and solutions relevant to the federal and provincial standards.

Question 1: Does the Web site contain auditory information and visual images?

What does this mean?

Auditory information is sound clips (e.g., Real Player, Media Player, QuickTime, and .wav files). Visual information includes images (GIF, JPEG, or PNG formats that may appear as text, drawings, photos, buttons, etc.), maps, graphical buttons, and animated GIFs. Auditory and visual images are used by Web site developers to enhance sites and to present information in formats other than text.

Why does this present a problem?

Some people with visual impairments and learning disabilities use text-to-speech screen readers that read aloud text appearing on the screen. However, screen readers cannot read content presented in an image or graphic format. Also, individuals with older computers or slow Internet connections may surf with image loading turned off and will miss information presented visually.

Likewise, information presented audibly will be inaccessible to individuals with hearing impairments, to those who have turned their speakers off when working in an environment where noise may disturb others (e.g., in an office or library), to individuals having difficulty understanding spoken language or accent, and to those using computers with no sound card.

What are the solutions?

When using auditory information and visual images, providing **equivalent information** is crucial. Equivalent information is content that conveys the same information as the auditory or visual content, such as providing captioning for a movie clip.

Below are some techniques and guidelines for providing equivalent information.

Priority
1

1. Use text equivalents

- For individuals with visual impairments and for those surfing with images turned off, text equivalents replace images or graphics by providing the same information.

- In HTML, the key to providing text equivalents is the ALT attribute. Quite simply, the ALT (meaning “alternative”) attribute provides a brief description that serves as an alternative to an image when the image is not viewed.
- Provide text equivalents for every non-text element, including images (JPEG, GIF, PNG), image map regions, animated GIFs, graphical buttons, etc.

Guidelines for developing ALT attributes:

- As text equivalents, ALT attributes should fulfill the same function as the image.

For example:

```
ALT="Canadian flag"
```

conveys the same information as an image of the Canadian flag when presented to users.

- Ensure the wording of ALT attributes is concise or terse. Avoid starting each ALT with a phrase such as “this button will...” This would unnecessarily slow down individuals using screen readers.
- Consider the length of the ALT text in relation to the size of the image. If possible, the text should display within the image space when images are off. For ALT attributes greater than 150 characters, consider providing a separate description (see next bullet).
- For complex content (e.g., a chart, table, or diagram) where the ALT text does not provide a complete text equivalent, the description can be included in the text of the main Web page, or placed in a separate page for this purpose and referenced by using the LONGDESC attribute or (descriptive) D-link.

The D-link, a convention established by the CPB/WGBH National Center for Accessible Media (<http://main.wgbh.org/wgbh/pages/ncam/>), is a way to link manually to the target of the LONGDESC by linking a D positioned near the image. As most current Web browsers do not yet support the LONGDESC attribute, it is recommended that both LONGDESC and D-link be used.

For example:

```
<IMG SRC="chart.gif" ALT="Chart comparing salaries of the top
five occupations"
LONGDESC="http://www.thismachine.com/salaries.txt">
<A HREF="salaries.txt">D</A>
```

- Do not specify irrelevant alternate text when including images for purely decorative purposes. For instance, when an image of a red ball is used for decorating a heading or paragraph, rather than using:

ALT="red ball"

simply use the empty ALT attribute:

ALT=" "

Even though the ALT attribute is empty, it is still necessary to use one. Otherwise, without an ALT attribute present, a screen reader will simply read "image" when encountering a graphic.

However, when using an image for a bullet in a list of items, an appropriate text equivalent may be:

ALT="*"

This will assist those using screen readers to differentiate listed items.

TIP A good test to determine if a text equivalent is useful is to imagine reading the document aloud over the telephone. What would you say upon encountering this image to make the page comprehensible to the listener?

TIP Some Web design software programs, e.g., WYSIWYG (what-you-see-is-what-you-get) editors, automatically use the file name and size as the default ALT.

For example:

ALT="image1.gif 546 bytes"

This is uninformative and rather boring to listen to when read aloud by a screen reader. Good Web designers make it a practice to add appropriate, informative ALT attributes in each required instance.

Priority
1

2. Include redundant text links with server-side image maps

- Web browsers, when used with a screen reader or when used with image loading turned off, have no way of accessing the hot-spots in a server-side image map. If the links at these hot-spots have no corresponding link in the body of the Web page then whole portions of a Web site can become unreachable.
- For each hot-spot, provide a text link elsewhere. For example, when providing a map of British Columbia with different regions on which users can click for more information, also list regions as hypertext links elsewhere on the page.

Priority
1

3. Provide auditory description and captioning

- When providing audio clips, also provide captions or a transcription.
- When providing video clips, provide captions or transcription as well as an **auditory description** of important visual information. Auditory descriptions

provide information about the action, body language, graphics, and scene changes in the video. Without verbal descriptions of this visual information, people who cannot see (or look at) the visual content are not able to perceive it.

Priority
1

4. Synchronize equivalentents

- Ensure captions and auditory descriptions are synchronized with the multimedia presentation.

Priority
3

5. Include redundant text links with client-side image maps

- Client-side image maps allow ALT attributes to be added to individual links. However, because some browsers and assistive technologies do not interpret the ALT attributes correctly, include redundant text links for each linked portion of the image map.

What is the bottom line?

The bottom line, as stated in WCAG Guideline 1, is provide ***equivalent alternatives*** to auditory and visual information.

Question 2: Does the site rely on colour alone to convey information?

What does this mean?

Is colour the only way some information can be differentiated? For example, on a course schedule, if the only difference between course start dates and end dates is the colour of the text.

Why does this present a problem?

Persons with visual and cognitive impairments (or those using monochrome monitors) may have difficulty distinguishing differences between colours. In addition, when printing the Web page in black and white, the different colours will be lost, making the information less understandable.

What are the solutions?

Priority
1

1. Use more than colour to convey information

- When using colour to convey information, also convey the information in other formats.

For example, in addition to saying:

Listed in red are the job requirements and the required qualifications are in purple.

use headings or another cue to indicate the different sections.

Similarly, rather than saying:

At any point during the tutorial, you may click the green circle to continue and the red circle to go back.

try saying:

...click the green circle to continue and the red triangle to go back.

- Some Web designers remove the line from beneath hyperlink text, so that only colour differentiates hyperlinks from regular text. If providing hyperlinks with no underline, provide a small graphic or other visual cue to represent a link.

TIP Test whether a Web page still works without colours by examining it with a monochrome monitor, with browser colours turned off, or by printing it on a black and white printer. This will indicate where redundant cues are needed and whether the cues are too small or indistinct to hold up well.

Priority
2

for images

2. Maximize colour contrast

- Enhance readability by maximizing contrast between foreground and background colours.
- Consider these suggestions when choosing colour schemes:
 1. Black on white is the most legible; white on black is reasonably legible; other colours on black are less legible.
 2. Mixing yellow and black is fairly legible.
 3. If using red or green text, make it large and bold enough to be legible in yellow.
 4. Black on red and black on green are not legible, as some people will see them as black on black.
 5. Combining blue and black is legible enough as long as it is not used for fine detail (e.g., paragraph text).
 6. Combinations such as red-blue, green-yellow, green-white, green-gray (MS Windows “button gray”) are poor because they have small brightness contrast and because red/green colour blindness is the most common.
 7. Avoid highly saturated backgrounds.

Priority
3

for text

TIP To test whether colour contrast is sufficient to be read by people with colour blindness or by those with low-resolution monitors, print pages on a black and white printer (with backgrounds and colours appearing in grayscale). Photocopy the copy two or three times to see how it degrades.

- To ensure cross-platform consistency in colour scheme, Treasury Board Standard 6.5 requires that only standard 216-Web-safe colours be used for Web site elements. Refer to http://www.cio-dpi.gc.ca/clf-upe/216_e.asp for the Web-safe colours and codes.

What is the bottom line?

The bottom line, as stated in WCAG Guideline 2, is don't rely on colour alone.

Question 3: Are markup and style sheets used properly?

What does this mean?

Web developers may be tempted to use markup incorrectly in order to achieve a desired effect. Use markup and style sheets as they are designed to be used.

Why does this present a problem?

Using markup for a presentation effect (e.g., using a table for layout or a header to change the font size) makes it difficult for individuals using some assistive technologies to understand the organization of the page or to navigate through it. For example, when using `<BLOCKQUOTE>` to create visual effects, such as indentation and alignment, rather than to markup a quotation, screen readers may make it unintelligible to the user.

What are the solutions?

Priority
2

1. Use style sheets to stylize text

- Style sheets are by far the best solution for text decoration and object placement because they separate content from layout. Unfortunately, style sheets are not consistently supported by all Web browsers. (For instructions on stylizing text without style sheets, see Question 6.)
- Do not use bitmapped images to represent text; instead, use style sheets.

In the example:

Career Information

use style sheets rather than graphics for the stylistic C and I.

Priority
2

2. Validate documents to published formal grammars

- Include a document type declaration (DOCTYPE) in Web pages. This declares the version of HTML used in documents and assists the browser in rendering pages correctly.
- Use checkers to validate documents. For example, see HTML Validation Service (<http://validator.w3.org/>) and CSS Validation Service (<http://jigsaw.w3.org/css-validator/>).

Priority
2**3. Use style sheets to control presentation**

- Distinguish between structural and presentational content and use style sheets to control presentation.

For example:

Structural	vs	Presentational
<ul style="list-style-type: none"> • • (emphasis) • <H1>...<H6> (headings) • <BLOCKQUOTE> • (unordered list) • (ordered list) • <DL> (definition list) • (list item) 		<ul style="list-style-type: none"> • (bold) • <I> (italics) • Style sheets

TIP To determine if content is structural or presentational, create an outline of the document. Each point in the hierarchy denotes a structural change. Use structural markup to mark these changes and presentational markup to make them more apparent visually.

- Use HTML elements correctly. For a complete list of the HTML 4.01 elements and their roles, refer to WCAG 1.0 (<http://www.w3.org/TR/WCAG10-HTML-TECHS/html-techniques.html#index-elements>).

Priority
2**4. Use relative rather than absolute units**

- Individuals use a wide variety of viewing devices, from very large monitors to tiny screens on handheld devices, and the content should flow into the appropriate size. This does not happen when absolute position and length units are used. Users need to be able to change the size of text to compensate for the resolution or size of their device or because of a visual impairment.
- Use the “em” unit to set font sizes.

For example, use:

```
H1 { font-size: 2em }
```

rather than:

```
H1 { font-size: 12pt }
```

- Use relative length units and percentages.

For example:

```
BODY { margin-left: 20%; margin-right: 15% }
```

- Use percentages when defining sizes of tables and frames.

Priority
2

5. Use headers

- Use headers to convey document structure and correctly nest headings.

For example:

```
<H1>Heading</H1>
  <H2>Sub-heading</H2>
    <H3>Another heading level</H3>
  <H2>Sub-heading</H2>
  <H2>Sub-heading</H2>
```

- Do not use headers for font effects; instead, use style sheets.

Priority
2

6. Mark up lists

- Properly use the list structure and list items to create lists.

For example:

```
<UL> unordered list
<OL> ordered list
<DL> definition list
<LI> list item
```

- Do not use these elements to simply provide formatting effects such as indentation; instead, use style sheets.
- For complex ordered lists, compound numbers are more informative than simple numbers. For example, a list numbered 1, 1.1, 1.2, 1.2.1, 1.3, 2, 2.1, provides more context than the same list without compound numbers.

Priority
2

7. Mark up quotations

- Use <Q> to mark up short quotations and <BLOCKQUOTE> for longer ones.
- Do not use these elements to simply provide formatting effects such as indentation; instead, use style sheets.

What is the bottom line?

The bottom line, as stated in WCAG Guideline 3, is use markup and style sheets, and use them properly.

Question 4: Does the Web page clarify natural language usage?

What does this mean?

Indicate changes in **natural language**—spoken, written, or signed human language—in a document. Also, provide expansions of abbreviations and acronyms.

Why does this present a problem?

Identifying changes in the language enables text-to-speech screen readers and Braille displays to automatically switch to the new language to enhance pronunciation. When changes are not marked, the synthesizer will try its best to speak the words in the primary language it works in. Thus, a speech synthesizer that uses English as its primary language would pronounce the French word for car, “voiture”, as “voter”. Natural language markup also allows search engines to find text in a desired language.

What are the solutions?

Priority
1

1. Identify changes in natural language

- Identify changes in the primary natural language of a document’s text or in text equivalents.

For example, in HTML use the LANG attribute:

```
<P>A great motto for living life is <SPAN LANG="la">carpe  
diem</SPAN> -- seize the day!</P>
```

TIP The International Standardization Organization (ISO) defines the language codes at <http://www.oasis-open.org/cover/iso639a.html>.

Priority
3

2. Expand abbreviations

- Use the <ABBR> and <ACRONYM> elements to expand abbreviations and acronyms the first time they occur on a page.

For example:

```
<ABBR TITLE="association">assoc.</ABBR>  
<ACRONYM TITLE="Labour Market Information">LMI</ACRONYM>
```

- Providing the expansion in the main body of the document ensures everyone understands the abbreviation or acronym.

For example:

```
<P>Labour Market Information (LMI) provides.... LMI is useful
for...</P>
```

Priority
3

3. Identify the document's primary natural language

- Specify the primary natural language of the entire document in the <HTML> element.

For example:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML lang="en">
```

What is the bottom line?

The bottom line, as stated in WCAG Guideline 4, is clarify the natural language being used, and transitions to and from other languages.

Question 5: Are tables understandable on various systems?

What does this mean?

As Web developers cannot control how their pages are viewed, tables need to be understandable when viewed in different resolutions or on different systems (e.g., screen readers, Braille displays, or palmtops).

Why does this present a problem?

People using screen readers, palmtops, or other devices, or people viewing only a portion of the page at a time (e.g., when using screen magnification or low-resolution) may have difficulty accessing information from pages with tables. Marking up tables properly provide these technologies with the appropriate information to support navigation in table cells and to access header and other table cell information.

Tables should be used to mark up truly tabular information (data tables) and, when possible, should be avoided to lay out pages (layout tables).

What are the solutions?

Priority
1

1. Identify table headers

- For data tables, identify rows and column headers. Use <TR> to identify rows, <TD> for data cells, and <TH> for headers.

The following example combines several solutions discussed in this question:

Block Transfer for Eagle University

Completed Program	Receiving Institution	Program	Credit
Biomedical Engineering	Raven University	Bachelor of General Sciences	Up to 60 credits
Forestry Technology	Rainbow Institute	Biological Sciences	55 credits

The code for the above table is:

```
<TABLE border="1" summary="This table charts the completed program, the receiving institution, the program, and the credit received.">
```

```
<CAPTION>Block transfer for Eagle University</CAPTION>
```

```

<TR><TH id="header1" ABBR="Completed">Completed
  Program</TH>
  <TH id="header2">Receiving Institution</TH>
  <TH id="header3">Program</TH>
  <TH id="header4">Credit</TH></TR>
<TR><TD headers="header1">Biomedical Engineering</TD>
  <TD headers="header2">Raven University</TD>
  <TD headers="header3"> Bachelor of General Sciences
    </TD>
  <TD headers="header4">Up to 60 credits</TD> </TR>
<TR><TD headers="header1">Forestry Technology</TD>
  <TD headers="header2">Rainbow Institute</TD>
  <TD headers="header3"> Biological Sciences </TD>
  <TD headers="header4">55 credits</TD></TR>
</TABLE>

```

A screen reader might read this table as follows:

Summary: This table charts the completed program, the receiving institution, the program, and the credit received.

Caption: Block transfer for Eagle University

Completed: Biomedical Engineering, Receiving Institution: Raven University, Program: Bachelor of General Sciences, Credits: Up to 60 credits

Completed: Forestry Technology, Receiving Institution: Rainbow Institute, Program: Biological Sciences, Credits: 55 credits

Priority
1

2. Associate data cells with headers

- For complex data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells.

Labeling table elements with the SCOPE or HEADER and the AXIS attributes will enable future browsers and assistive technologies to select data from a table by filtering categories. For example, in the following table there are three axes: funding, location, and program.

Comparison by Region of Government Funding for Programs

	Funded by Federal Government	Funded by Provincial Government
Region A		
Self-employment Programs	25	2
Training Programs	6	12
Region B		
Self-employment Programs	36	4
Training Programs	4	26

The code for the above table is:

```

<TABLE border="1" SUMMARY="This table compares the number
of self-employment and training programs in regions A and B
funded by the federal and provincial governments">
<CAPTION>Comparison by Region of Government Funding for
Programs </CAPTION>
<TR>
  <TH></TH>
  <TH id="header2" axis="funding" abbr="Federal">Funded by
  Federal Government</TH>
  <TH id="header3" axis="funding" Abbr="Provincial">Funded
  by Provincial Government</TH>
</TR>
<TR>
  <TH id="header4" axis="location">Region A</TH>
  <TH></TH>
  <TH></TH>
</TR>
<TR>
  <TD id="header5" axis="program">Self-employment
  Program</TD>
  <TD headers="header4 header5 header2">25</TD>
  <TD headers="header4 header5 header3">2</TD>
</TR>
<TR>
  <TD id="header6" axis="program">Training Programs</TD>
  <TD headers="header4 header6 header2">6</TD>
  <TD headers="header4 header6 header3">12</TD>
</TR>
<TR>
  <TH id="header7" axis="location">Region B</TH>
  <TH> </TH>
  <TH> </TH>
</TR>
<TR>
  <TD id="header8" axis="program">Self-employment
  Program</TD>
  <TD headers="header7 header8 header2">36</TH>
  <TD headers="header7 header8 header3">4</TH>
</TR>
<TR>
  <TD id="header9" axis="program">Training Programs</TD>

```

```

<TD headers="header7 header9 header2">4</TD>
<TD headers="header7 header9 header3">26</TD>
</TR>
</TABLE>

```

A screen reader may read the table as follows:

Summary: This table compares the number of self-employment and training programs in regions A and B funded by the federal and provincial governments

Caption: Comparison by Region of Government Funding for Programs

Region A, Self-employment Programs, Federal: 25, Provincial: 2

Region A, Training Programs, Federal: 6, Provincial: 12

Region B, Self-employment Programs, Federal: 36, Provincial: 4

Region B, Training Programs, Federal: 4, Provincial: 26

Priority
2

3. Avoid tables for layout

- Use style sheets for layout and positioning.
- When layout tables are unavoidable, ensure tables make sense when **linearized**. Cells should make sense when read in order and should include structural elements (that create paragraphs, headers, lists, etc.) so the page makes sense after linearization.

For example:

Welcome to Career Corner!	
Navigation link 1	Banner Ad
Navigation link 2	Main content area filling the center part of the window.
Navigation link 3	
Navigation link 4	

This simple table produces the following text when unstacked or linearized:

Welcome to Career Corner!

Navigation link 1

Navigation link 2

Navigation link 3

Navigation link 4

Banner Ad

Main content area filling the center part of the window.

TIP A simple way of viewing a page in linear form is to view it in the Lynx text browser. The Lynx viewer at <http://www.delorie.com/Web/lynxview.html> provides a rendering of the page as if the browser were Lynx.

Priority
2

4. Avoid structural markup for formatting

- Do not use structural markup for formatting text. For example, do not use <TH> to display a non-table heading centered and in bold.
- Use style sheets to mark up presentation (see Question 3).

Priority
3

5. Provide table summaries

- <SUMMARY> describes the table's structure and purpose when the table itself cannot be seen. It is not normally displayed to visual users.
- Similarly, <CAPTION> assists in understanding data tables. Captions are displayed visually, usually above the table.

For example:

```
<TABLE border="1" summary="This table charts the completed
  program, the receiving institution, the program, and the credit
  received.">
<CAPTION>Block Transfer for Eagle University</CAPTION>
```

Priority
3

6. Abbreviate long headers

- If column headers are long, use the ABBR attribute in the <TH>.

For example:

```
<TR><TH id="header1" ABBR="Completed">Completed
  Program</TH>
```

What is the bottom line?

The bottom line, as stated in WCAG Guideline 5, is ensure tables have necessary markup to be transformed gracefully by assistive devices.

Question 6: Are pages featuring new technologies understandable on older systems?

What does this mean?

Web developers are often eager to use the newest technologies. They are even encouraged to do so as using new technologies generally solve problems caused by older ones. And after all, being on the cutting edge is fun and exciting! However, developers still need to make their pages work with older browsers.

Why does this present a problem?

Some older browsers or *assistive technologies* may not support the newest technologies. Also, users may choose to turn features off to speed up download time.

What are the solutions?

Priority
1

1. Make documents readable without style sheets

- Ensure that pages are readable and usable without style sheets for browsers that do not support them. This means organizing content in a logical order to ensure display is meaningful without style sheets.

TIP One way of testing readability is to place style information in a separate file, rather than inline, and temporarily rename that file while testing.

- Making documents readable without style sheets is important even though solutions outlined in Question 3 suggest using style sheets. Style sheets are by far the best solution for text decoration and object placement but they are not consistently supported by all browsers.
- Be aware that organizing pages so they may be read without style sheets may require use of deprecated elements. As will be discussed in Question 11, avoid using deprecated elements when possible, but when using them, do so with caution.

Priority
1

2. Update dynamic content

- **Dynamic content** is content that changes over time or in response to user actions. When dynamic content changes, equivalent information must also be updated. For example, with a timed series of images depicting various careers, text descriptions must change with the images.
- Frame sources should be HTML files (not images) so text alternatives change correctly.

The following inaccessible example illustrates why frame sources cannot be images:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
  Frameset//EN">
<HTML>
  <HEAD>
    <TITLE>A bad frameset document</TITLE>
  </HEAD>
  <FRAMESET cols="100%" title="Static frameset">
    <FRAME name="badframe" src="bcmap.gif" title="Map of
      British Columbia">
  </FRAMESET>
</HTML>
```

If a link causes a new image to be inserted into the frame:

```
<P>Visit the British Columbia's Interior <A target="badframe"
  href="interior.gif" title="Interior Region">for unique
  employment opportunities</A>.</P>
```

the initial title of the frame ("Map of British Columbia") will no longer match the current content of the frame ("Interior Region").

Correct coding looks like this:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
  Frameset//EN">
<HTML>
  <HEAD>
    <TITLE>A correct frameset document</TITLE>
  </HEAD>
  <FRAMESET cols="100%" title="Evolving frameset">
    <FRAME name="goodframe" src="map.html" title="Map of
      British Columbia">
  </FRAMESET>
</HTML>
<!-- In map.html -->
<P><IMG src="bcmap.gif" alt="Map of British Columbia">
```

Priority
1

3. Make pages usable with programmatic objects turned off

- Programmatic objects include scripts, applets and other plug-ins. Some individuals turn off scripts or cannot download plug-ins, and some browsers do not support scripts. Content presented in these formats may be inaccessible to these individuals.
- Avoid using scripts that are essential to page content (as opposed to trivial, e.g., adding colour effects when the mouse is moved over an object) so the page will still function when the scripts are turned off.

TIP

Check if the page is usable with scripting turned off by turning off JavaScript in your browser.

To turn off JavaScript in Internet Explorer 5.0 or higher:

1. Go to **Tools** → **Internet Options** → **Security** → **Internet**.
2. Select **Custom Level** button at the bottom.
3. Scroll down to **Scripting** under **Active Scripting** and select the radio button **Disable**.

For Netscape Navigator:

1. Go to **Edit** → **Preferences** → **Advanced**.
2. Uncheck **Enable JavaScript**.

- Use <NOSCRIPT> to provide alternate content. If pages are unusable, provide an alternative accessible page.

Priority
2

4. Create independent event handlers

- Event handlers define how scripts react to events caused by something the user does (e.g., the mouse moves, a key is pressed, the document is loaded, etc.).
- Event handlers must be **device independent**—users must be able to interact with a browser (and the document it renders) using the devices suiting their needs (e.g., mouse, keyboard, voice input, sip-and-puff device, etc.).
- Use event handlers that work for both mouse users and keyboard users (e.g., onClick). Alternatively, for a single event, use a handler that works for mouse users (e.g., onMouseDown) *and* a handler for keyboard users (e.g., onKeyDown).
- Do not write event handlers that rely on mouse coordinates because this will prevent device-independent input.

TIP

To test that event handlers are input device independent, unplug the mouse and navigate the page simply with the keyboard.

Priority
2

5. Ensure accessible dynamic content

- Ensure that dynamic content is accessible or provide an alternative presentation or page.
- For scripts, use <NOSCRIPT> to provide alternate content.
- For frames, provide a <NOFRAME> option for those who cannot read framed information. Giving the message “upgrade your browser” is no help to someone whose browser simply doesn’t handle frames. <NOFRAME> should contain meaningful content with links to the other pages that can be accessed without frames.
- Treasury Board Standard 6.6 mandates using frames only as an alternate format. Users must be given a choice between <FRAME> and <NOFRAME>.

What is the bottom line?

The bottom line, as stated in WCAG Guideline 6, is ensure that pages are accessible even when newer technologies are not supported or features are turned off.

Question 7: Can users control scrolling or changing content?

What does this mean?

Web developers often use blinking or scrolling text or animation to draw attention to particular content. Also, content can be set to automatically change or redirect users to another page within a specified length of time. When using such features, ensure they can be paused or stopped by users.

Why does this present a problem?

Some individuals with visual impairments or cognitive disabilities cannot read moving text quickly enough and most screen readers are not able to read moving text at all. Individuals with mobility impairments may not be able to respond quickly or accurately enough to respond to moving objects. Movement can be annoying or distracting to any reader.

What are the solutions?

Priority
1

1. Eliminate flickering screens

- Features, such as applets, plug-ins, scripts, and animated images and movies, may cause the screen to flicker. Quick motion and sudden changes in colour can have this effect.

TIP People with photosensitive epilepsy can have seizures triggered by flickering or flashing in the 4 to 59 flashes per second (Hertz) range with a peak sensitivity at 20 flashes per second. Quick changes from dark to light (like strobe lights) can also trigger seizures. [From WebAIM www.Webaim.org/tutorials/usercontrol]

American legislation, Section 508 of the Rehabilitation Act further specifies not using frequencies between 2 and 55 Hz. [From <http://www.access-board.gov/508.htm>]

Priority
2

2. Avoid blinking text

- Use other ways to highlight text, such as presentation markup. For example, the following style sheet properties can be used to style text:

- Case: “text-transform” (for uppercase, lowercase, and capitalization)
- Underline, overline: “text-decoration”
- If text must be highlighted by causing it to flash on and off, then use the style sheet ‘text-decoration: blink’ attribute. Using style sheets will allow users to stop the effect.
- Do not use the <BLINK> and <MARQUEE> elements as they are not defined by W3C (refer also to Question 11).

TIP Some screen readers are unable to read blinking text. They may stick on the text and read it repeatedly, or they may freeze the entire computer system.

Priority
2

3. Make moving content stoppable

- Moving content, such as scrolling text, can be distracting and interfere with an individual’s ability to focus on relevant content, or the individual may read more slowly than the scrolling speed. It is important that users be able to freeze the movement of text.
- When content must move, provide a mechanism within a script or applet to allow users to freeze motion or updates. Use style sheets with scripting to enable users to turn off or override the motion.

Priority
2

4. Avoid auto-refresh

- Do not cause pages to auto-refresh with “HTTP-EQUIV=refresh” until various Web browsers and assistive technologies allow individuals to turn off the feature. In the following inaccessible example, the page would refresh or reload itself every 60 seconds:

```
<HEAD>
  <META http-equiv="refresh" content="60">
</HEAD>
<BODY>
  <P>Main Page</P>
</BODY>
```

- Automatically refreshing pages can be disorienting to some users. Instead, when using a continually updated page, inform the users that they should reload the page often.
- If using auto-refresh is unavoidable, provide a single line of text at the top of the page stating that the page contains changing information and will

automatically reload itself after a certain amount of time. Also, provide a link to the next screen for those browsers that don't support the feature.

Priority
2

5. Avoid auto-redirect

- Individuals accessing a page with a screen reader or who read slowly may need more time to read the page. If the page auto-redirects unexpectedly it may prevent access to content. It can also disrupt a browser's history of visited pages.
- As with auto-refresh, provide a single line of text at the top of the page stating that the site has moved and provide a hyperlink to the new page.

For example:

The *Career Corner* has moved. Please come visit us at <http://www.newsite.com>. Be sure to update your bookmarks.

What is the bottom line?

The bottom line, as stated in WCAG Guideline 7, is give the user control of time-sensitive content changes.

Question 8: Are embedded user interfaces compatible with assistive technologies?

What does this mean?

Java applets, media players (e.g., Windows Media Player, RealVideo, QuickTime), Flash animations, and Shockwave scripts require user interaction separate from the HTML-coded Web pages. These interfaces need to follow accessible design principles, such as device-independent access to functionality, keyboard operability, self-voicing, etc. Failing this, an accessible alternative should be included in the HTML pages describing the function of the elements and providing an equivalent function.

Why does this present a problem?

Some assistive technology cannot interpret scripted or programmed Web content, making the content inaccessible or unusable to some users.

What is the solution?

Priority
1

if functionality
is important,
otherwise

1. Create directly accessible interfaces

- Make these features directly accessible by following accessible principles or by providing an equivalent alternative.

Priority
2

TIP For further information about accessible interfaces, see W3C's User Agent Accessibility Guidelines (<http://www.w3.org/TR/UAAG10/>) and Authoring Tool Accessibility Guidelines (<http://www.w3.org/TR/ATAG10/>).

TIP WebAIM compares the accessibility of embedded and standalone versions of three media players in "To Embed or Not To Embed - A Comparison of Media Player Technologies" (<http://www.Webaim.org/Articles/embeddedmp.php>).

TIP Resources to assist you in making applets accessible:

- Java Accessibility (<http://trace.wisc.edu/world/java/java.htm>) – Trace R&D Center IBM Guidelines for Writing
- IBM Guidelines for Writing Accessible Applications Using 100% Pure Java (<http://www-3.ibm.com/able/snsjavag.html>) – IBM Accessibility Center
- Active Accessibility, Microsoft (<http://www.msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28000544>)

What is the bottom line?

The bottom line, as stated in WCAG Guideline 8, is give the user access to embedded user interfaces.

Question 9: Is the page designed for a variety of input devices?

What does this mean?

It is said that surfing the Net is as simple as “point and click” but for individuals who use an input device other than a mouse, it is not that simple. Web developers need to design for a variety input devices. The key here is device independence—enabling users to successfully use the Web page, regardless of how they interact with the computer.

Why does this present a problem?

Not all users can use a mouse or other pointing device. Some users rely on keyboard, alternative keyboard, or voice input to navigate links and activate form controls. Elements that will require alternatives include image maps (server-side and client-side), plug-ins (Flash, Shockwave, etc.), and form elements (text boxes, buttons, etc.).

What are the solutions?

Priority
1

1. Favour client-side image maps

- Where possible, use client-side rather than server-side image maps because client-side image maps support the ALT attribute.
- Provide redundant text links for each active region on the image map. For example, when providing a map of British Columbia with clickable regions, also list the regions as hypertext links.

Priority
2

2. Provide device-independent elements

- Make sure that non-HTML scripts or objects are operable without a mouse. For example, when using a Flash movie introduction, provide a text link outside the movie to allow users to access the content of the site without using the Flash movie.

Priority
2

3. Specify logical event handlers

- Event handlers invoke scripts when specific events occur, usually because of something the user does (e.g., the mouse moves, a key is pressed, the

document is loaded, etc.). For a script to react to an event, event handlers must be defined, such as `onChange` and `onClick`.

- Device-dependent handlers specify the actual device in the event handler name, such as `onMouseOver` or `onKeyPress`.
- Logical (or generic) event handlers are interested in what happened to the element (e.g., has gained or lost focus or has been selected), not what caused it to happen (e.g., keyboard or mouse). Logical event handlers include `onSelect`, `onFocus`, and `onBlur`.
- HTML Techniques for Web Content Accessibility Guidelines 1.0 (<http://www.w3.org/TR/WCAG10-HTML-TECHS/>) provide these suggestions for event handlers that do more than simply change the presentation of an element:

1. Use application-level event triggers. In HTML 4.01, application-level event attributes are `ONFOCUS`, `ONBLUR` (the opposite of `ONFOCUS`), and `ONSELECT`.

Note: These attributes are designed to be device-independent, but are implemented as keyboard specific events in current browsers.

2. If device-dependent attributes must be used, provide redundant input mechanisms (e.g., specify two handlers for the same element):
 - Use `ONMOUSEDOWN` with `ONKEYDOWN`
 - Use `ONMOUSEUP` with `ONKEYUP`
 - Use `ONCLICK` with `ONKEYPRESS`

Note: There is no keyboard equivalent to double-clicking (`ONDBLCLICK`) in HTML 4.01.

3. Do not write event handlers that rely on mouse coordinates since this prevents device-independent input.
- To make pop-up windows accessible to keyboard users, add the event triggers `ONFOCUS` and `ONBLUR`.

For example, rather than using:

```
<a href="foo.html" onMouseOver="blah()" >
```

change to:

```
<a href="foo.html" onMouseOver="blah()" onFocus="blah()" >
```

Priority
3

4. Create logical tab order

- Use `<TABINDEX>` to define the logical order in which elements will receive focus when individuals use the keyboard to navigate the Web page.

For example:

```
<FORM action="submit" method="post">
  <INPUT tabindex="2" type="text" name="lastname">
  <INPUT tabindex="1" type="text" name="firstname">
  <INPUT tabindex="3" type="submit" name="submit">
</FORM>
```

Here, individuals using the keyboard to navigate will input their first name before their last name.

- These elements support the TABINDEX attribute:
 - <A>
 - <INPUT>
 - <SELECT>
 - <AREA>
 - <OBJECT>
 - <TEXTAREA>
 - <BUTTON>

Priority
3

5. Provide keyboard shortcuts

- Provide keyboard shortcuts so that users may combine keystrokes to navigate links or form controls on a page.

For example:

```
<A accesskey="C" href="contents.html">Table of Contents</A>
```

- Visually indicate an ACCESSKEY by underlining, bolding, or capitalizing the letter defined as the ACCESSKEY.
- These elements support the ACCESSKEY attribute:
 - <A>
 - <INPUT>
 - <LEGEND>
 - <AREA>
 - <LABEL>
 - <TEXTAREA>
 - <BUTTON>

However, the ACCESSKEY is not widely used because of the following limitations:

- Browsers have their own keyboard shortcuts; for example, Netscape assigns Alt H to Help. Considering the number of browsers (Netscape, Internet Explorer, Lynx, Opera, pwWebSpeak, etc.), each with their own keyboard shortcuts, the list of shared, unassigned keyboard combinations is extremely limiting. The numbers 0-9 are probably the only safe ACCESSKEY combinations to use.
- Web site developers feel there is no visually sensible way to inform visitors of keystrokes shortcuts.
- The ACCESSKEY *may* work in an Intranet setting, i.e., within one organization where the platform and browser type are the same throughout.

What is the bottom line?

The bottom line, as stated in WCAG Guideline 9, is design for device-independence.

Question 10: Are interim accessibility solutions used to ensure that older assistive technology and browsers will operate correctly?

What does this mean?

Until assistive technologies and browsers catch up with newer Web design features or capabilities, use interim solutions (as described below) to ensure accessibility.

Why does this present a problem?

Some assistive technologies and older browsers may not correctly handle some of the newer Web design features or elements, making these capabilities inaccessible. For example, older browsers may not allow users to navigate to empty edit boxes, and older screen readers may read lists of consecutive links as one link.

What are the solutions?

Priority
2

1. Announce new windows

- Users who cannot see or who do not notice that the current window has changed or that a new window has popped up, may be disoriented by the change.
- Inform users when using pop-up windows, opening a new window, or changing the focus to a new <FRAME>.

For example:

Try this [Skills Inventory](#). (This will open a new window.)

The code may look like:

```
<P>Try this <A HREF="skills.htm" TARGET="_blank">Skills
Inventory</A>. (This will open a new window.) </P>
```

Priority
2

2. Associate form controls and labels

- Users who cannot see may not be able to determine which label applies to which form control and therefore will not know what data to enter. Proper positioning of the label helps to avoid this.

- For each form control, place its label in a <LABEL>. A label is attached to a specific form control using the FOR attribute. The value of the FOR attribute equals the value of the ID attribute.

For example:

```
<LABEL for="firstname">First name:
  <INPUT type="text" id="firstname" tabindex="1">
</LABEL>
```

- Labels for radio buttons and checkboxes should appear after the radio button or checkbox.

For example:

Choose your area of interest:
 Personal Development Management Skills

- Labels for text fields appear before the text field.

For example:

First name
 Last name

- Labels for groups of any form control should appear before the group.

For example:

Choose a city: Prince George Nelson
 Choose a program: Arts Science

Priority
3

3. Provide linear text alternatives

- Text presented in columnar format may cause difficulties for individuals using screen readers, as well as for some browsers, depending upon the font size, window size, and screen resolution.
- When providing text in a columnar format, also provide a linear text alternative.

TIP To gain a better understanding of how a screen reader may read a table, cover the table with a piece of paper and move it down slowly, reading the table line-by-line.

Priority
3

4. Include default text in form controls

- Some assistive technologies will miss a form control if there is no text in it. It will not inform the user that the control is there or allow the user to input data.

Placing default text in the control forces the assistive technology to read the control.

- Default text for <TEXTAREA> controls is located between the opening and closing tags.

For example:

```
<FORM action="http://www.careercorner.com/prog/text-read"
  method="post">
  <P>
    <TEXTAREA name=yourname rows="20" cols="80">Please
      enter your name here.
    </TEXTAREA>
    <INPUT type="image" name=submit src="button.gif"
      alt="Submit">
  </P>
</FORM>
```

Be sure to include text equivalents for images used as “submit” and “reset” buttons.

- Default text for <SELECT> controls (drop-down lists and list boxes) is accomplished by making one of the options active with a SELECTED attribute.

For example:

```
<SELECT name="department">
  <OPTION value="1" selected>Accounting</OPTION>
  <OPTION value="2">Management</OPTION>
</SELECT>
```

- Radio button groups should have an option selected with the CHECKED attribute of the appropriate radio button.

For example:

```
<INPUT type="radio" name="salary-type" value="exempt">
  Exempt
<INPUT type="radio" name="salary-type" value="non-exempt"
  checked> Non-Exempt
```

- To enhance the accessibility of forms, the Treasury Board Standards suggest providing alternate methods of contact or submission, e.g., via email, phone, or TTY (teletype for the deaf).

Priority
3

5. Separate adjacent links

- Use non-linked, printable characters between adjacent links to enable screen readers to pause. This will help individuals with visual impairments to distinguish where one link phrase ends and the next begins. The visual cue

may also help people with cognitive disabilities to identify the break between links.

For example:

[\[Home\]](#) [\[Training Programs\]](#) [\[Career Information\]](#) [\[Resources\]](#)

The code would look like this:

```
<P>[<A HREF="home.html">Home</A>] [<A  
  HREF="training.html">Training Program</A>] [<A  
  HREF="career.html">Career Information</A>] [<A  
  HREF="resources.html">Resources</A>]</P>
```

- Images or bulleted or numbered lists are also good choices for separating links.

What is the bottom line?

The bottom line, as stated in WCAG Guideline 10, is use interim solutions.

Question 11: Are current W3C technologies and guidelines used?

What does this mean?

Use technologies and guidelines specified by the World Wide Web Consortium (W3C) (<http://www.w3c.org>) because they are reviewed for accessibility issues and have accessibility features built in.

Why does this present a problem?

This presents a problem only if a page is not designed to W3C specifications, which are developed in an open, industry consensus process. When non-W3C technologies are used, the user may be required to use plug-ins or stand-alone applications that are not compatible with assistive technologies.

What are the solutions?

Priority
2

1. Use current W3C technologies

- Use the latest version of W3C technologies when suitable for a task.
- When using HTML, Treasury Board Guideline 1.1 recommends (but does not mandate) that HTML 4.0 Strict or newer W3C adopted languages be used as the standard for new and revised Web pages.
- Current W3C technologies include:
 - MathML for mathematical equations
 - HTML, XHTML, XML for structured documents
 - RDF for metadata
 - SMIL to create multimedia presentations
 - CSS and XSL to define style sheets
 - XSLT to create style transformations
 - PNG for graphics

TIP The latest W3C technologies are available from the W3C Technical Reports and Publications page (<http://www.w3.org/TR/>).

- When HTML or other W3C technologies cannot be used, Treasury Board Standard 1.2 requires that users be given information on how to obtain

alternate versions, e.g., print, Braille, audio, Portable Document Format (PDF) (see Question 15 for more on PDF files).

- The Treasury Board provides this example of explanatory text to include before the inaccessible element (optional language is enclosed between square brackets):

If the following [information, content, document, application, form, interactive questionnaire, animation, multimedia presentation] is not accessible to you, please contact [name, email, phone number, TTY number, mailing address or other appropriate contact information] for [assistance, explanations, alternate formats such as regular print, large print, Braille, audio cassette or other appropriate format].

Priority
2

2. Avoid deprecated features

- When possible, avoid use of deprecated elements as browsers may not support them in the future.
- Be aware that organizing pages so they are readable without style sheets, as required under Question 6, may require use of deprecated elements, e.g., , but they should be used with caution.
- The following are elements that have been removed in HTML 4.0 and some recommended alternatives:

Removed	Recommended Alternative
<ul style="list-style-type: none"> • <APPLET> • <BASEFONT>, <CENTER>, , <STRIKE>, <U> for formatting • <DIR>, <MENU> to define lists • <ISINDEX>, for simple form input • <LISTING>, <PLAINTEXT>, <XMP> for formatting fixed-width text 	<ul style="list-style-type: none"> • <OBJECT> • style sheets for these formatting effects • list elements such as or and style sheets to refine the formatting • an <INPUT> element within a <FORM> • <PRE>

TIP

Refer to HTML 4.0 Changes (<http://www.w3.org/TR/1998/REC-html40-19980424/appendix/changes.html#h-A.1.1.2>) for a complete listing of changes from HTML 3.2 to HTML 4.0.

Priority
3

3. Provide appropriate content

- The Core Techniques for Web Content Accessibility Guidelines 1.0 provides the following strategies to help users to select the appropriate content:
 1. Include links to other versions of content, such as translations. For example, the link “Refer to the French version of this document” links to the French version.
 2. Indicate content type or language through markup (e.g., in HTML use TYPE and HREFLANG).
 3. Use content negotiation to serve content per the client request. For example, serve the French version of a document to clients requesting French.
 4. See the W3C Internationalization document on HTTP Content Negotiation for more information.

[From <http://www.w3.org/TR/WCAG10-CORE-TECHS/#content-negotiation>]

- Treasury Board Guideline 1.2 recommends (but does not mandate) using the <MEDIA> element in cascading style sheets for pages explicitly designed to provide information to alternate technologies such as hand-held, print, Braille, and audio devices. This element allows documents to render more appropriately on these technologies. It can also reduce download times by allowing user agents to ignore inapplicable rules.

Priority
1

4. Provide alternative pages

- Only as a last resort, when a page cannot be made accessible, provide a link to a separate page that is accessible—one that uses W3C technologies, has equivalent information, and is updated as often as the original page. This can be done by providing a prominent link that says:

[Text-only version](#)

- Use an alternative page only when necessary, as it is time consuming updating two sets of pages. Unless they are updated as regularly as the original pages, alternative pages become inaccessible.
- Treasury Board Standard 1.3 requires Web pages offering information in alternate formats to provide a hyperlink to a site for obtaining the necessary software.

For example:

Interested in this job? For more information:

- Corporate Information ([HTML](#), [PDF](#))
- Complete Job Description ([HTML](#), [PDF](#))
- Application ([HTML](#), [PDF](#))

Note: to read the PDF version you need [Adobe Acrobat Reader 4.0 \(or later\)](#) on your system. If the Adobe download site is not accessible to you, you can download Acrobat Reader from the [text-only page](#). Alternatively, you can have the file converted to HTML or ASCII text by using one of the [access services](#) provided by Adobe.

What is the bottom line?

The bottom line, as stated in WCAG Guideline 11, is use W3C technologies and guidelines.

Question 12: Does the page provide context and orientation information?

What does this mean?

Providing context and orientation information, grouping elements, and providing information about the relationships between elements can aid all users in navigating and comprehending Web content.

Why does this present a problem?

Complex pages can be confusing to the user, making content inaccessible.

What are the solutions?

Priority
1

1. Use frame titles

- Use the TITLE attribute with the <FRAME> element to describe the content of the frame. This enables users who are not able to access multiple frames simultaneously to pick which frame to view.

For example:

```
<FRAMESET ROWS="20%, 80%">
  <FRAME SRC="navigate.html" TITLE="Navigation Bar">
  <FRAME SRC="content.html" TITLE="Main Content">
</FRAMESET>
```

- Treasury Board Standard 6.6 mandates using frames only as an alternate format—users must have a choice between <FRAME> and <NOFRAME>.

Priority
2

2. Provide frame descriptions

- If the purpose of the frames and how they relate to each other is not obvious by the frame titles alone, use LONGDESC to provide a further description.

Building upon the previous example:

```
<FRAMESET ROWS="20%, 80%">
  <FRAME SRC="navigate.html" TITLE="Navigation Bar"
    LONGDESC="frame-desc.html#navbar">
  <FRAME SRC="content.html" TITLE="Main Content"
    LONGDESC="frame-desc.html#main">
</FRAMESET>
```

A [frame-desc.html](#) might say something like:

#Navbar - this frame provides links to the major sections of the site: Career Information, Job Opportunities, and Resources.

#Main - this frame displays the currently selected selection.

Priority
2

3. Manage information

- Use headings to structure documents (see Question 3).
- Use list elements (e.g., and) and nest them as needed.
- Some browsers will display grouped drop-down lists with expanding and collapsing levels of detail to simplify comprehension and navigation. Use <OPTGROUP> to group long lists of selections into a hierarchy.

For example:

```
<FORM action="http://example.com/prog/someprog"
  method="post">
<SELECT name="higher">
  <OPTION label="0" value="0" selected> Choose an
    institution</OPTION>
  <OPTGROUP label="colleges">
    <OPTION label="0" value="0">--COLLEGES--</OPTION>
    <OPTION label="c1" value="c1">Big Ben College</OPTION>
    <OPTION label="c2" value="c2">College of Blue
      River</OPTION>
    <OPTION label="c3" value="c3">Eagle College</OPTION>
  </OPTGROUP>
  <OPTGROUP label="universities">
    <OPTION label="0" value="0">--UNIVERSITIES--</OPTION>
    <OPTION label="u1" value="u1">University of the Red
      Herring</OPTION>
    <OPTION label="u2" value="u2">Rainbow
      University</OPTION>
  </OPTGROUP>
</SELECT>
</FORM>
```

The screenshot shows a web browser window with a dropdown menu. The menu is open, displaying a list of options. The first option is "Choose an institution" with a small downward arrow to its right. Below this, there is a separator line followed by the text "--COLLEGES--". Underneath, three options are listed: "Big Ben College", "College of Blue River", and "Eagle College". Another separator line follows, with the text "--UNIVERSITIES--". Below this, two more options are listed: "University of the Red Herring" and "Rainbow University". The "Choose an institution" option is highlighted with a blue background.

Note: HTML 4.0 does not permit nested <OPTGROUPS>. While a future version may add this capability, it is advisable to create only a single grouping level with this element.

- Grouping form controls provides additional context information by explicitly showing that they relate in some way. Use <LEGEND> to label each <FIELDSET>.

For example:

```
<FORM action="http://somesite.com/adduser" method="post">
  <FIELDSET>
    <LEGEND>Personal information</LEGEND>
    <LABEL for="firstname">First name:</LABEL>
    <INPUT type="text" id="firstname" tabindex="1">
    <LABEL for="lastname">Last name:</LABEL>
    <INPUT type="text" id="lastname" tabindex="2">
    <!--...more personal information...-->
  </FIELDSET>
  <FIELDSET>
    <LEGEND>Address</LEGEND>
    <LABEL for="street">Street address:</LABEL>
    <INPUT type="text" id="street" tabindex="3">
    <LABEL for="city">City:</LABEL>
    <INPUT type="text" id="city" tabindex="4">
    <!--...more address information...-->
  </FIELDSET>
</FORM>
```

Priority
2

4. Associate form controls with labels

- For each form control, place its label in a <LABEL> element (see Question 10).

What is the bottom line?

The bottom line, as stated in WCAG Guideline 12, is provide context and orientation information.

Question 13: Does the site provide clear navigation mechanisms?

What does this mean?

Web developers should provide clear and consistent navigation mechanisms—orientation information, navigation bars, site maps, etc.—to increase the likelihood that users will find what they are looking for.

Why does this present a problem?

Without clear and consistent navigation, all users, regardless of cognitive, visual, or other disability may have a difficult time navigating a complex or confusing Web site.

What are the solutions?

Priority
2

1. Clearly indicate text links

- Make a hypertext link informative when read out of context, whether it is on its own or as part of a sequence of links. For example, rather than using the word “more”, use full descriptions like “more about career information” or “full text of the report”. Make link text succinct.
- Avoid using “click here” as clicking refers to using the mouse, which is not possible for everyone, and it is meaningless when read out of context.
- Do not use the same link phrase to describe links that point to different URLs. Same link text should point to the same resource.

For example:

Bad form: For ` more information`
`on question 2`

For ` more information`
`on question 3`

Better: For ` more information on`
`question 2`

For ` more information on`
`question 3`

- Use the TITLE attribute to provide more information about links, when needed.

For example:

```
<A href="application.html">The application is available in
HTML</A>,
<A href="application.pdf" title="Application in PDF">PDF</A>,
<A href="application.txt" title="Application in text">plain
text</A>
```

Priority
2

2. Provide metadata

- Metadata—information about data—provides information to users, and is used by search engines and for site maps, content ratings, etc.
- Relevant HTML elements include:
 - <!DOCTYPE>
 - <LINK>
 - <TITLE>
 - <META>
 - <ADDRESS>

Note: Do not use <META HTTP-EQUIV="Refresh"> to auto-refresh or auto-redirect (see Question 7, Solution 4).

TIP Refer to <http://www.w3.org/TR/html401/struct/global.html> for information on the global structure of an HTML document.

- Treasury Board Standard 6.3 requires that all Government of Canada Web sites adopt the following five metatags as a metadata standard for description of Web resources: Title, Originator, Language of Resource, Date, and Controlled Subject.

Priority
2

3. Provide general layout information

- Provide a site map or table of contents to assist users in navigating the site.
- Site maps are often presented as outlines, composed of nested lists, showing the hierarchy of pages.

For example:

- ▶ [What's New](#)
- ▶ [About Us](#)
 - [Who We Are](#)
 - [Contact Us](#)
 - [Become A Member](#)
- ▶ [Job Seekers](#)
 - [Work News & Features](#)
 - [Job Links](#)
 - [Resources](#)
- ▶ [Employers](#)
 - [Human Resources News](#)
 - [Resources](#)

Reminder: Place separating characters (printable characters, images, bullets or numbers) between adjacent links.

Priority
2

4. Use consistent navigation mechanisms

- Typical navigation mechanisms include navigation bars, site maps and tables of contents. Using these mechanisms consistently increases the usability of the site information.

Priority
3

5. Include navigation bars

- Provide consistent navigation bars to assist users in moving about the site.
- Treasury Board Standard 6.1 requires that all Government of Canada Web pages include the common menu bar at the top of each page. The menu bar must contain, in this order: Language (English/French) for bilingual sites only, Contact Us, Help, Search, and Canada Site.
- Treasury Board Standard 6.2 requires an institutional menu bar, similar in design and placement to the common menu bar. Plain language is also required for buttons that identify an organization's programs and services.

Priority
3

6. Skip related links

- As navigation bars are generally at the top of the page, individuals using screen readers must listen to the navigation links before getting to the main content. Similarly, people using the keyboard must tab through the links to get to the main content. This can be time-consuming and repetitive.
- Grouping related links into a unit (e.g., a navigation bar) enables a skip feature to be implemented by one of these methods:
 1. Placing hypertext links at the beginning of related links (e.g., navigation bars) to enable users to "skip to main content" or "skip navigational links".
 2. Using <MAP> to group links and identify the group with TITLE.

For example:

```
<BODY>
<MAP title="Navigation Bar" name="navbar">
  <P>
    [<A href="#home">Bypass navigation bar</A>]
    [<A href="home.html">Home</A>]
    [<A href="career.html">Career Info</A>]
    [<A href="job.html">Jobs</A>]
    [<A href="sitemap.html">Site map</A>]
  </P>
</MAP>
```

```

<H1><A name="home">Welcome to Career
  Corner!</A></H1>
<!--main content of page -->
</BODY>

```

[\[Bypass navigation bar\]](#) [\[Home\]](#) [\[Career Info\]](#) [\[Jobs\]](#) [\[Site map\]](#)

Priority
3

7. Provide a variety of search functions

- Provide search functions that satisfy varying skill levels and preferences. Search engines might include a spellchecker, “best guess” alternatives, query-by-example searches, similarity searches, etc.
- Users with spelling disabilities and users unfamiliar with the language of the site will have a difficult time finding what they need if the search requires perfect spelling.

Priority
3

8. Implement front-loading

- Commonly referred to as “front-loading”, place distinguishing information at the beginning of headings, paragraphs, lists, etc. to decrease the amount of information readers must process in order to find important information. This is especially helpful for people accessing information linearly (e.g., via screen readers).

As an example, this list front-loads the links:

1. [Web Accessibility Initiative](#), the definitive source for accessibility standards.
2. [Common Look and Feel Standards](#), the Government of Canada’s Internet policy developed by the Treasury Board.
3. [Bobby](#), a free service for evaluating the accessibility of sites.

Priority
3

9. Provide information about document collections

- If the document (Web page) is part of a collection, provide information that identifies this document’s location within the collection. This can be achieved by:
 1. Including a table of contents, site index, or site map.
 2. Using page counters (e.g., page *x* of *y*) to indicate the position in relation to the whole set.
 3. Using <LINK> with REL or REV to indicate the URL of the next and previous pages for extra navigation information.

For example, the <HEAD> of this chapter may include:

```

<LINK rel="Contents" href="toc.html">
<LINK rel="Question 12" href="q12.html">
<LINK rel="Question 14" href="q14.html">
<LINK rel="Glossary" href="glossary.html">
<LINK rel="Index" href="index.html">

```

The text-based browser Lynx will build a small menu of links at the top of the page that reads like this:

```
# Contents Question 12 Question 14 Glossary Index
```

Priority
3

10. Skip ASCII Art

- As screen readers read aloud every character in **ASCII art** (images created with text characters and symbols), provide a link enabling users to jump over it.
- If using ASCII art, provide a text equivalent.

TIP Signature Files

Avoid or minimize ASCII art in signature files at the bottom of email messages.

For example, in the email signature below, a screen reader will sound each asterisk:

```
*****
```

```
Jo Anybody, Career Counsellor
```

```
Email: janybody@xxx.com
```

```
*****
```

What is the bottom line?

The bottom line, as stated in WCAG Guideline 13, is provide clear navigation mechanisms.

Question 14: Is the site's content written in clear and simple language?

What does this mean?

Ensure that documents are clear and simple so they are easily understood.

Why does this present a problem?

Inconsistent page layout, unrecognizable graphics, and jargon and complex language can hinder comprehension and usability. Written information can be difficult for individuals who have cognitive and learning disabilities, for individuals whose first language differs from the site's content, and for individuals whose primary means of communication is sign language. Using clear and simple language promotes effective communication.

What are the solutions?

Priority
1

1. Use simple language

- The Core Techniques for Web Accessibility Guidelines 1.0 provides these tips on writing style:
 1. Strive for clear and accurate headings and link descriptions. Use link phrases that are succinct and that make sense when read out of context or as part of a series of links (some users browse by jumping from link to link and listening only to link text). Use informative headings so that users can scan a page quickly for information rather than reading it in detail.
 2. State the topic of the sentence or paragraph at the beginning of the sentence or paragraph (i.e., front-loading). This helps those people skimming visually and those skimming with screen readers, which means jumping from heading to heading, or paragraph to paragraph and listening to enough words to determine whether the current chunk of information interests them.
 3. Limit each paragraph to one main idea.
 4. Avoid slang, jargon, and specialized meanings of familiar words, unless defined within your document.

5. Favour words that are commonly used. For example, use “begin” rather than “commence” or use “try” rather than “endeavour”.
6. Use active rather than passive verbs.
7. Avoid complex sentence structures.

Priority
3

2. Supplement text

- Use visual and auditory cues to enhance comprehension. Provide text equivalents.
- The Core Techniques for Web Accessibility Guidelines 1.0 provides these examples:
 1. A chart of complex data, such as sales figures of a business for the past fiscal year.
 2. A translation of the text into a Sign Language movie clip. Sign Language is a very different language than spoken languages. Some people who communicate via American Sign Language are not able to read American English.
 3. Pre-recorded audio of music, spoken language, or sound effects may also help non-readers who can perceive audio presentations.

Priority
3

3. Use consistent presentation style

- Consistent page style allows users to locate or skip navigation mechanisms so they can more easily find important content.
- A consistent page design should include:
 1. Navigation bar and buttons in the same location on each page.
 2. Consistent colours, text styles, heading locations, graphics style, etc.
 3. Consistently placed supplementary text (copyright, promotional, etc.), identifying images and other navigation or end-of page features.

[From WebAIM <http://www.Webaim.org/tutorials/simple#14.3>]

- Treasury Board Standard 6.4 requires that all pages include a date at the bottom to signal the end of the page and to indicate how current the content is. Dates use the all-numeric date display (YYYY MM DD) and use one of the following formats: Date published, Date modified, or Last updated.

What is the bottom line?

The bottom line, as stated in WCAG Guideline 14, is ensure that documents are clear and simple.

Question 15: Are PDF files only used as an alternate format?

What does this mean?

Portable Document Format (PDF) is a popular method for electronically distributing documents worldwide. Documents saved in this format retain the same fonts, graphics, colours and layouts as when originally created.

The Treasury Board of Canada Standard 1.2 requires using PDF files (minimum version 2.1) only as an alternate format. That is, users must have a choice which format they use—HTML version, Word document, text (ASCII), or PDF.

Why does this present a problem?

Because some PDF files are simply scanned images of printed pages, people using screen readers are not able to read these documents. Even though Adobe offers free PDF-to-HTML or PDF-to-ASCII text conversion services (see Online Conversion Tools for Adobe PDF Documents (<http://access.adobe.com/onlinetools.html>), not all converted documents are accessible. Tables, graphs, pictures and columnar layout can also cause problems.

What are the solutions?

1. Use PDF files as alternate format

- Provide documents in various formats, e.g., HTML and PDF.
- Read the booklet, “How to Create Accessible Adobe PDF Files” (<http://access.adobe.com/booklet.html>).

What is the bottom line?

The bottom line is provide PDF files only as an alternate format to HTML.

Appendix A: Summary of Web Accessibility Standards

	Web Content Accessibility Guidelines 1.0 Priority	Treasury Board of Canada Common Look and Feel Standards	B.C. Government Internet Standards*
Question 1: Does the Web site contain auditory information and visual images?			
1. Use text equivalents	1	✓	✓
2. Include redundant text links with server-side image maps	1	✓	✓
3. Provide auditory description and captioning	1	✓	✓
4. Synchronize equivalents	1	✓	✓
5. Include redundant text links with client-side image maps	3		
Question 2: Does the site rely on colour alone to convey information?			
1. Use more than colour to convey information	1	✓ See also Standard 6.5	✓
2. Maximize colour contrast	2 for images, 3 for text	✓ See also Standard 6.5	✓
Question 3: Are markup and style sheets used properly?			
1. Use style sheets to stylize text	2	✓	✓
2. Validate documents to published formal grammars	2	✓	✓
3. Use style sheets to control presentation	2	✓	✓
4. Use relative rather than absolute units	2	✓	✓
5. Use headers	2	✓	✓
6. Mark up lists	2	✓	✓
7. Mark up quotations	2	✓	✓

* At the time of writing this Guide, British Columbia is revising its Internet standards to be “in alignment with the W3C Web Content Accessibility Guidelines (complying with Priority 1 and considering Priority 2 checkpoints).”

	Web Content Accessibility Guidelines 1.0 Priority	Treasury Board of Canada Common Look and Feel Standards	B.C. Government Internet Standards*
Question 4: Does the Web page clarify natural language usage?			
1. Identify changes in natural language	1	✓	✓
2. Expand abbreviations	3		
3. Identify the document's primary natural language	3		
Question 5: Are tables understandable on various systems?			
1. Identify table headers	1	✓	✓
2. Associate data cells with headers	1	✓	✓
3. Avoid tables for layout	2	✓	✓
4. Avoid structural markup for formatting	2	✓	✓
5. Provide table summaries	3		
6. Abbreviate long headers	3		
Question 6: Are pages featuring new technologies understandable on older systems?			
1. Make documents readable without style sheets	1	✓	✓
2. Update dynamic content	1	✓	✓
3. Make pages usable with programmatic objects turned off	1	✓	✓
4. Create independent event handlers	2	✓	✓
5. Ensure accessible dynamic content	2	✓	✓
Question 7: Can users control scrolling or changing content?			
1. Eliminate flickering screens	1	✓	✓
2. Avoid blinking text	2	✓	✓
3. Make moving content stoppable	2	✓	✓
4. Avoid auto-refresh	2	✓	✓
5. Avoid auto-redirect	2	✓	✓
Question 8: Are embedded user interfaces compatible with assistive technologies?			
1. Create directly accessible interfaces	1 if functionality is important, otherwise 2	✓	✓

	Web Content Accessibility Guidelines 1.0 Priority	Treasury Board of Canada Common Look and Feel Standards	B.C. Government Internet Standards*
Question 9: Is the page designed for a variety of input devices?			
1. Favour client-side image maps	1	✓	✓
2. Provide device-independent elements	2	✓	✓
3. Specify logical event handlers	2	✓	✓
4. Create logical tab order	3		
5. Provide keyboard shortcuts	3		
Question 10: Are interim accessibility solutions used to ensure that older assistive technology and browsers will operate correctly?			
1. Announce new windows	2	✓	✓
2. Associate form controls and labels	2	✓	✓
3. Provide linear text alternatives	3		
4. Include default text in form controls	3		
5. Separate adjacent links	3		
Question 11: Are current W3C technologies and guidelines used?			
1. Use current W3C technologies	2	✓	✓
2. Avoid deprecated features	2	✓	✓
3. Provide appropriate content	3	See Guideline 1.2	
4. Provide alternative pages	1	✓	✓
Question 12: Does the page provide context and orientation information?			
1. Use frame titles	1	✓ See also Standard 6.6	✓
2. Provide frame descriptions	2	✓	✓
3. Manage information	2	✓	✓
4. Associate form controls with labels	2	✓	✓
Question 13: Does the site provide clear navigation mechanisms?			
1. Clearly indicate text links	2	✓	✓
2. Provide metadata	2	✓ See also Standard 6.3	✓
3. Provide general layout information	2	✓	✓

	Web Content Accessibility Guidelines 1.0 Priority	Treasury Board of Canada Common Look and Feel Standards	B.C. Government Internet Standards*
4. Use consistent navigation mechanisms	2	✓	✓
5. Include navigation bars	3	See Standards 6.1 & 6.2	
6. Skip related links	3		
7. Provide a variety of search functions	3		
8. Implement front-loading	3		
9. Provide information about document collections	3		
10. Skip ASCII Art	3		
Question 14: Is the site's content written in clear and simple language?			
1. Use simple language	1	✓	✓
2. Supplement text	3		
3. Use consistent presentation style	3	See Standard 6.4	
Question 15: Are PDF files only used as an alternate format?			
1. Use PDF files as alternate format	n/a	See Standard 1.2	

Glossary

ASCII art

Images created with text characters and symbols, including emoticons, e.g., the smiley face :)

Assistive technology

Software or hardware created or modified to enable people with disabilities to use computers. Examples of assistive technologies that may be used while surfing the Web include the following:

- **Alternative keyboards and switches/pointing devices:** A variety of input methods used in place of the standard keyboard and mouse. Used by individuals with limited hand function or severe mobility impairments.
- **Refreshable Braille display:** Hardware in which pins raise and lower to display Braille, a tactile system using raised dots to represent letters and numbers. Used by individuals with visual impairments.
- **Scanning software:** Software that highlights selection choices (e.g., letters, menu items, links, phrases) one at a time. User selects desired item by hitting a switch when the desired item is highlighted or announced. Used by individuals with physical or cognitive disabilities.
- **Screen magnifier:** Software that magnifies a portion of the screen for easier viewing. Used by individuals with low vision.
- **Screen reader:** Software that interprets what is displayed on screen and directs it to either speech synthesis for audio output or refreshable Braille for tactile output. Used by individuals who are blind or have reading disabilities.
- **Speech recognition software:** Software that allows input using voice for data entry and limited software commands. Used by individuals with limited or no hand function, or with learning disabilities.

Auditory description

A verbal description providing information about actions, body language, graphics, and scene changes in a video. Descriptions must be synchronized with the auditory track of the video, usually during natural pauses in the auditory track. Used by people who are blind or have low vision; may also be used as a low-bandwidth equivalent on the Web.

Deprecated elements

HTML elements or attributes replaced by newer ones. Web developers should avoid using these elements as they may become obsolete in future versions of HTML. Refer to <http://www.w3.org/TR/1998/REC-html40-19980424/appendix/changes.html#h-A.1.1.2> for a complete listing of changes from HTML 3.2 to HTML 4.0.

Device independent

User may interact with the Web page or browser with a preferred input or output device (e.g., mouse, keyboard, Braille, voice, head pointer, or other assistive technology). For example, if a form control can only be activated with a mouse or other pointing device, someone who is using the page without sight, with voice input, with a keyboard, or who is using some other non-pointing input device will not be able to use the form.

Dynamic content

Content that changes over time or in response to user actions.

Equivalent information (or Equivalent alternative)

Essential in accessible Web design since certain types of content may not be accessible to all users (e.g., video, images, audio, etc.). Content is equivalent to other content when both fulfill essentially the same function or purpose. For example, ALT="Canadian flag" conveys the same information as an image of the Canadian flag when presented to users.

Linearized table

Table contents unstacked or extracted from the row/column layout and presented in simple order as a block of text.

Natural language

Spoken, written, or signed human language such as English, French, or American Sign Language.

Screen reader

Software that interprets what is displayed on screen and directs it to either speech synthesis for audio output or refreshable Braille for tactile output (see also *assistive technology*).

Text equivalent

Provides a written substitute for graphic and audio information.

References

- B.C. Internet Services. "B.C. Government Internet Standards." January 2001.
http://www.gov.bc.ca/bcis/down/WebStandards_Jan_2001.pdf
- Center for Applied Special Technology (CAST). "Bobby 3.2."
<http://www.cast.org/bobby>
- Paciello, Michael G. "Web Accessibility for People with Disabilities." CMP Books, 2000.
- Training and Standards Sub-Committee of the Canada/BC Joint Committee for Enhanced Labour Market Information, "The Accessible LMI Project: Addressing Labour Market Information Issues for Persons with Disabilities." 1998.
<http://workinfonet.bc.ca/lmisi/jointcom/AccessLMI/Index.htm>
- Treasury Board of Canada Secretariat. "Common Look and Feel Self-Assessment Guide." http://www.cio-dpi.gc.ca/clf-upe/guide/guidepr_e.asp
- Treasury Board of Canada Secretariat. "Common Look and Feel Standards and Guidelines for the Internet." http://www.cio-dpi.gc.ca/clf-upe/apr_e.asp
- WebAIM. "Tutorials." <http://Webaim.org/tutorials/>
- World Wide Web Consortium. "Techniques for Web Content Accessibility Guidelines 1.0." November 2000. <http://www.w3.org/TR/WCAG10-TECHS/>
- World Wide Web Consortium. "Web Content Accessibility Guidelines 1.0." May 1999. <http://www.w3.org/TR/WAI-WEBCONTENT/>

Index

<

<ABBR>	16, 19, 22
<ACCESSKEY>.....	34
<ACRONYM>	16
<BLINK>	28
<BLOCKQUOTE>.....	13, 14, 15
<CAPTION>.....	18, 20, 22
<FORM>	34, 37, 40, 44, 45
<FRAME>	26, 35, 43
<HEAD>	24, 28, 49
<LABEL>.....	34, 36
<LINK>.....	47, 49
<MARQUEE>.....	28
<OPTGROUP>	44
<SELECT>.....	34, 37
<SUMMARY>	22
<TABINDEX>.....	33, 34
<TEXTAREA>.....	34, 37
<TITLE>	47

A

ACCESSKEY	34
Adobe	42, 53
ALT	2, 8, 9, 10, 32, 60
ASCII art	50, 59
assistive technology	1, 3, 10, 13, 19, 23, 28, 30, 35, 36, 37, 39, 56, 57, 59, 60
auditory description	9, 55, 59

B

B.C. Government Internet Standards	4, 5, 55
--	----------

C

captioning.....	1, 2, 7, 9, 10, 22, 55
colour	2, 11, 12, 25, 27, 52, 53, 55
Common Look and Feel Standards and Guidelines for the Internet	4, 5, 55

D

deprecated elements.....	2, 23, 40, 60
device independent	25, 30, 32, 33, 34, 57, 60
dynamic content.....	24, 26, 56, 60

E

equivalent alternative	10, 30, 60
equivalent information.....	7, 24, 41, 60
event handler	25, 32, 33, 56, 57

F

forms.....	22, 32, 34, 35, 36, 37, 40, 45, 57, 60
frame.....	15, 24, 26, 43, 44, 57

H

hot-spot.....	9
hyperlink.....	11, 29, 41
hypertext	9, 32, 46, 48

I

image map	8, 9, 10, 32, 55, 57
-----------------	----------------------

L

linearized table.....	21, 60
LONGDESC	8, 43

N

natural language	16, 17, 56, 60
navigation mechanisms.....	46, 48, 50, 52, 57, 58

P

PDF	40, 41, 42, 47, 53, 58
-----------	------------------------

S

screen reader..	1, 7, 8, 9, 13, 16, 18, 21, 27, 28, 29, 35, 36, 37, 48, 49, 50, 51, 53, 59, 60
scripts.....	25, 26, 27, 28, 30, 32, 33
style sheets..	2, 3, 13, 14, 15, 21, 22, 23, 27, 28, 39, 40, 41, 55, 56

T

tables	2, 8, 13, 15, 18, 19, 20, 21, 22, 36, 48, 53, 56
text equivalents	2, 7, 8, 9, 16, 37, 50, 52, 55, 60
TITLE	43, 46, 48
Treasury Board ...	4, 5, 6, 12, 26, 37, 39, 40, 41, 43, 47, 48, 49, 52, 53, 55

W

Web Content Accessibility Guidelines.....	4, 5, 6, 10, 12, 14, 15, 17, 22, 26, 29, 31, 33, 34, 38, 41, 42, 45, 50, 52, 55
---	---