

Web Accessibility Guidelines

For UN Websites

INTRODUCTION	3
A GENERAL ASSEMBLY MANDATE	3
WHAT IS WEB ACCESSIBILITY?	3
COSTS AND BENEFITS	3
AUDIO AND AUDIOVISUAL CONTENT	4
ACCESSIBILITY VALIDATION	4
I - PAGE DESIGN: VISUAL ELEMENTS	5
ALTERNATIVE TEXT	5
DO NOT RELY ON COLOUR ALONE TO CONVEY INFORMATION	6
COLOUR CONTRAST	7
USE MARKUP RATHER THAN IMAGES	8
NO FLICKERING, BLINKING OR MOVING CONTENT	9
II - PAGE DESIGN: CODING	11
ORGANIZE DOCUMENTS SO THEY MAY BE READ WITHOUT STYLE SHEETS	11
FRAMES	11
CREATE VALID DOCUMENTS	13
CONTROLLING LAYOUT AND PRESENTATION	14
LAYOUT TABLES	15
USING RELATIVE UNITS	16
IDENTIFY THE MAIN LANGUAGE	17
III - TEXTUAL CONTENT	19
CHANGES IN THE NATURAL LANGUAGE	19
SIMPLE AND CLEAR LANGUAGE	19
DATA TABLES	20
TITLES	24
LISTS	26
QUOTATIONS	27
METADATA	28
ACRONYMS AND ABBREVIATIONS	29
IV - NAVIGATION	31
MEANINGFUL LINK TITLES	31
SITEMAP	31
AVOIDING POP-UPS AND LINKS OPENING IN NEW WINDOWS	32
CONSISTENT NAVIGATION	33
LOGICAL TAB ORDER	34
PROVIDE INFORMATION ABOUT DOCUMENTS	35
SKIP LINKS	36
V - SCRIPTS AND FORMS	39
PAGES ARE USABLE WHEN PROGRAMMATIC ELEMENTS ARE NOT SUPPORTED	39
ACCESSIBLE DYNAMIC CONTENT	40
NO AUTO-REFRESH	41
NO CLIENT-SIDE AUTOMATIC REDIRECTS	42
LABELS AND FORMS	43
DEVICE INDEPENDENCE.....	43
VI - ACCESSIBLE ALTERNATIVES	46
UP-TO-DATE EQUIVALENTS	46
HTML ALTERNATIVES FOR NON-W3C TECHNOLOGIES	47
ANNEX	49
VALIDATION PROCEDURE FOR DESIGNERS AND DEVELOPERS	49
VALIDATION PROCEDURE FOR CONTENT PROVIDERS.....	55

Introduction

A General Assembly Mandate

In order to make information on the United Nations and its activities available to the widest range of audiences, and in keeping with resolutions of the General Assembly, the Department of Public Information (DPI) is aggressively pursuing accessibility of the United Nations website by persons with disabilities. The objective of DPI is to comply with the Web Content Accessibility guidelines¹ of the World Wide Web Consortium standards, which at the present time are the only standards with worldwide recognition.

Following the adoption of the Convention on the Rights of Persons with Disabilities on 13 December 2006 (resolution 61/106), the General Assembly requested the Secretary-General to progressively implement standards and guidelines for the accessibility of facilities and services of the United Nations System. To assist other departments and offices in producing accessible web content, those guidelines were prepared by a task force composed of members of the Web Services Section.

What is Web Accessibility?

“Web accessibility means that people with disabilities can use the Web. [...] Web accessibility encompasses all disabilities that affect access to the Web, including visual, auditory, physical, speech, cognitive, and neurological disabilities.” (W3C)

Persons with disabilities are faced with many challenges when using the Internet. They use of assistive technologies that help them to access information. For instance, persons with visual impairments may use screen magnifiers or screen readers that read out loud the content of a page or translate it on a Braille display. Persons with mobility impairment may be unable to use a mouse and they surf the web with the keyboard or another assistive device.

Special attention has to be dedicated to web accessibility in the web design process to ensure a site is usable by persons with different types of disabilities, using a wide range of assistive technologies.

Costs and Benefits

Making a website accessible to disabled persons requires technical expertise in web design and accessibility issues.

Creating accessible pages takes more time. However, it is much more productive to consider this aspect at the start of a project than to modify an existing, inaccessible site. It should also be noted that creating accessible websites carries other benefits:

¹ The present guidelines are based on the Web Content Accessibility Guidelines 1.0 of the W3C.

- They are more user-friendly because they strive to give users the greatest level of control on how they wish to access content;
- they are optimized for display on any type of device, such as mobile phones or PDAs;
- they are easier to access with low-bandwidth connection;
- they are optimized for search engine indexation which contributes to an increased visibility in search results;
- they are easier to maintain: redesigns and language versions can be implemented quicker.

Audio and Audiovisual Content

In order to ensure full access, audio and audiovisual content should be closed-captioned. However, due to increased resource requirements for this aspect, DPI has not yet been able to tackle the issue of audio and audiovisual content accessibility. Therefore, those guidelines will not address the issues related to audio and video content.

Accessibility Validation

There are a number of automatic validation tools that can provide partial accessibility validation. They can be used for preliminary validation purposes. A list of such tools is referenced on the W3C website.

Once the issues identified with an automated tool are solved, it is necessary to complete the evaluation manually. The annex of this document details a procedure for manual validation.

I - Page Design: Visual Elements

Alternative Text

WAI Definition (Checkpoint 1.1)

Provide a text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content). *This includes:* images, graphical representations of text (including symbols), image map regions, animations (e.g., animated GIFs), applets and programmatic objects, ASCII art, frames, scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video.

Explanation

Provide a text equivalent for every non-text element as listed above. The text equivalent can be in the form of an "alt" attribute or a "longdesc" attribute.

Rationale

Some types of information may not be directly accessible to everyone. Images, for example, are not directly accessible by someone who cannot see, and audio files are not directly accessible by someone who cannot hear. Certain users will therefore require that the equivalent information or functionality be provided in text format in order to be compatible with assistive technologies.

Technique

Use an "alt" and/or a "longdesc" attribute on every "img" element or other non-text object inserted in the HTML code.

The "alt" attribute gives a short description of the image. It is the most common way of providing an alternative text description.

"alt" tag for a photo of New York:

```

```

The "longdesc" attribute provides a link to a longer description of the image. This is particularly useful for images that convey a lot of information such as charts.

"alt" attribute and "longdesc" for a pie chart

```

```

If an image does not convey any information and only serves a decorative purpose, add an empty "alt" attribute. Decorative elements can be defined in a CSS style sheet.

“ALT” attribute for a decorative image
`` (*there should be no space between the quotes*)

When an image is used in a form to replace a submit button, use the “alt” attribute.

“alt” attribute for an image serving as a form button
`<input class="img_button" type="submit" name="Submit" value="Submit" alt="submit">`

Validation

Using the Firefox Web Developer’s Toolbar (plugin):

Go to **Images** > **Outline images without “alt” attribute**

Go to **Images** > **Replace images with “alt”**

Success Criteria

No image exists without an “alt” attribute.

The information conveyed through the use of images is replaced by descriptive “alt” attribute. No information is missing.

There are no “alt” attribute with meaningless information (such as “spacer” or “blue line”). “Meaningless information” is considered information that is not fully descriptive of the element.

Do Not Rely on Colour Alone to Convey Information

WAI Definition (Checkpoint 2.1)

Ensure that all information conveyed with colour is also available without colour, for example from context or markup.

Explanation

If colour is necessary, include a text label in conjunction with colour to communicate information effectively. The user should be able to clearly identify and understand an element on the page without the use of colour.

Rationale

One in twelve people suffers from some sort of colour deficiency. Some cannot distinguish red, green or blue. Others have difficulties distinguishing foreground from background, if the contrast is insufficient.

Technique

Give other visual or textual clues.

Below are some common situations which **should be avoided**:

1. Buttons using colour as the only descriptive label.

Click on the green button to continue. Click on the red button to stop the registration process.



Complement the colour cues with text:



2. Avoid similar statements: “All fields marked in red are obligatory”. Instead use other visual clues: “All fields marked with * are obligatory”

3. Do not rely on colour alone to highlight links outside of navigation elements. The best practice is to underline links. Alternatively, links could be bold.

Validation

Using the Firefox Web Developer’s Toolbar (plugin):

Go to **Disable** > **Disable page colours**

Go to **Images** > **Disable all images**

Success Criteria

There is no critical information missing when colours and/or images are disabled. Links in the text are clearly identified and underlined.

Colour Contrast

WAI Definition (Checkpoint 2.2)

Ensure that foreground and background colour combinations provide sufficient contrast when viewed by someone having colour deficits or when viewed on a black and white screen.

Explanation

In text and images, ensure that foreground and background colour combinations have sufficient contrast when viewed by someone having impaired colour vision or when viewed on a black and white screen.

Rationale

Ensuring proper contrast between foreground and background colours is important for users, who are partially sighted or colour-blind, as well as for

people who use a notebook screen or a projector on a sunny day. This particularly affects the elderly and those who have some form of colour-blindness.

Technique

Use the validation tools to ensure that colour combinations offer sufficient contrast.

Validation

A. For text and images with flat colours:

Colour contrast requirements can be tested using “Juicy Studio Color Analyzer” at:

<http://juicystudio.com/services/luminositycontrastratio.php#specify>

There is also a Firefox extension that allows for the testing of all colours on a page (with the exception of colours in images):

<http://juicystudio.com/article/colour-contrast-analyser-firefox-extension.php>

Success Criteria

All the colour combinations in the page pass level AAA or AA for regular text size.

B. For images without flat colours:

Run the image through <http://www.vischeck.com/vischeck/> to simulate how the page will look if viewed by a colour-blind person.

Success Criteria

The information conveyed by the image is still understandable.

Use Markup Rather Than Images

WAI Definition (Checkpoint 3.1)

When an appropriate markup language exists, use markup rather than images to convey information.

Explanation

Avoid using images to display text, for example in a navigation menu. Use HTML styled with CSS instead.

Rationale

Visually-impaired users may have more difficulties to read text in an image because it cannot be altered to fit user preferences: font size cannot be increased; the level of contrast cannot be modified.

Furthermore, an image does not convey structural information. If an image is used to display a title, a blind user will not know that this particular image is to be considered as a title because it will not be enclosed in a heading tag (<h1>, <h2>).

Technique

Use CSS to format text in HTML.

Designers who must use an image to display text to achieve a formatting effect that cannot be achieved with CSS alone should consider using the technique described below. It hides equivalent HTML text behind the image. Note that this technique is not supported under certain configurations. This technique should only be used for title banners, not for navigational elements.

Example

CSS

```
#hiddentitle, #hiddentitle span {
    margin: 0;
    padding: 0;
    background-image: url(images/idptop.jpg);
    background-repeat: no-repeat;
    height: 135px;
}
#hiddentitle span {margin-bottom: -75px;}
```

HTML

```
<div id="banner">
    <h1 id="hiddentitle"><span></span>International Day of Peace, Portal
    2007</h1>
</div>
```

Validation

Using the Firefox Web Developer's Toolbar (plugin):

Go to **Images** > **Disable Images** > **All images**

Success Criteria

Check that all the text in the page is still visible with all images disabled.

No Flickering, Blinking or Moving Content

WAI Definition (Checkpoints 7.1, 7.2 and 7.3)

- Until user agents allow users to control flickering, avoid causing the screen to flicker.

- Until user agents allow users to control blinking, avoid causing content to blink (i.e., change presentation at a regular rate, such as turning on and off).
- Until user agents allow users to freeze moving content, avoid moving elements in pages..

Explanation

Avoid visual elements which can cause the screen to flicker, move or blink.

Rationale

Screen flickering and blinking can trigger epileptic seizures in people suffering from photosensitive epilepsy.

Older screen readers also have problems with blinking or moving content and may crash or be unable to render the content at all.

Furthermore, it is difficult to read some text that is flashing or moving, or to read the content of a page when something else is flashing or moving on that page. Users read at different speeds and moving text does not accommodate those differences. Blinking or moving text will frequently distract visitors from what is important on the site.

Finally, the `<blink>` and `<marquee>` tags are not part of the HTML specification and, as such, should be avoided.

Technique

Do not create elements that flicker or blink.

Validation

Check the page content.

Success Criteria

Ensure that the page does not contain elements that flicker, blink or move. There are no brusque changes in luminosity or flash effects.

II - Page Design: Coding

Organize Documents So They May Be Read Without Style Sheets

WAI Definition (Checkpoint 6.1)

Organize documents so that they may be read without style sheets. For example, when an HTML document is rendered without associated style sheets, it must still be possible to read the document.

Explanation

Document layout should be organized in a comprehensible format in order to be legible without style sheets. When style sheets are turned off, all content and links should be accessible and functional to the user.

Rationale

Support for style sheets is not consistent across all browsers and platforms. Furthermore, users have the ability to override the original style sheet with one of their own which might be better suited for their needs (for example, with larger fonts or a different colour scheme).

It is therefore important that the structure of a document can be communicated in a comprehensible order, even when style sheets are disabled. This will ensure that the structure is clear when an alternative style sheet is used.

Technique

The structure of the document should be reflected in the HTML: titles should be tagged with heading tags (<h1>, <h2>), menus and lists should be in list tags (), quotations should be in blockquote tags (<blockquote>).

Validation

Using the Firefox Web Developer's Toolbar (plugin):

Go to CSS > Disable styles > All styles.

Success Criteria

The document is still readable. Elements such as headings or lists can be recognised as such.

Frames

WAI Definition (Checkpoints 12.1 and 12.2)

Title each frame to facilitate frame identification and navigation. Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone.

Explanation

Using frames is not recommended, but if they must be used, it is very important to use descriptive frame names and define them in the <noframe> tag.

Rationale

Users of older and limited browsers or users using screen readers have difficulty in navigating web pages that use frames. When these users open a page with frames, they are presented with a list of frames that make up the page. The user must then navigate each frame individually, a frustrating task if the frame titles are confusing. If this occurs, the user is forced to open each frame individually, review the content and then try to figure out how it relates to the other frames on the page.

Technique

Use a descriptive frame title:

A frame named “nav” is not very informative; instead, the frame’s name should be something like “navigation”.

Use the <noframe> tags to provide links with access to the content without frames in the event that frames are not supported.

Example

```
<frameset>
  <frame name="banner" src="banner.html">
  <frame name="navigation" target="main" src="navigation.html">
  <frame name="main" src="main.html">
</frameset>
<noframes>
  <p>This page uses frames, but your browser doesn't support them.</p>
  <p>Please select the page you wish to navigate to from the list below:</p>
  <ul>
    <li><a href="navigation.html">Navigation page</a></li>
    <li><a href="main.html">Main Page</a></li>
  </ul>
</noframes>
```

Validation

Open the page source for the entire frame:

In the browser, go to **Menu** > **View** > **Source** to check that there are proper alternatives inside the <noframe> tag.

Success Criteria

There are no frames.

OR

If there are frames, the content in the <noframe> tag provides an accurate description of each frames as well as links to access the content of the frames.

Create Valid Documents

WAI Definition (Checkpoint 3.2)

Create documents that validate to published formal grammars.

Explanation

Create documents with valid HTML or XHTML.

Rationale

HTML and XHTML syntax is standardized to ensure a consistent interpretation of pages by different browsers and devices. A valid page will therefore be more widely supported, whether it is accessed by an old browser, a screen reader or a handheld device.

Technique

Add a “doctype” declaration (see example below) at the top of the HTML page if it is not present in order to specify which version of HTML or XHTML is used.

If unfamiliar with HTML syntax, please make sure a recent version of a robust HTML editor is used. Do not use MS Word to create HTML files.

Tools such as “HTML Tidy” are helpful in cleaning up the code of existing pages.

Examples

HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/HTML4/loose.dtd">
```

XHTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Validation

Using the Firefox Web Developer’s Toolbar (plugin):

Validate the HTML: Go to **Tools** > **Validate HTML**

Validate the CSS: Go to **Tools** > **Validate CSS**

Note

The W3C validator can also be accessed directly from the W3C site <
<http://validator.w3.org/>>.

Success Criteria

The code passes validation. No errors are found.

NOTE

The W3C validator only checks if the page is coded according to defined standards. It does not validate other aspects of accessibility.

Controlling Layout and Presentation

WAI Definition (Checkpoints 3.3 and 11.2)

- Use style sheets to control layout and presentation.
- Avoid deprecated features of W3C technologies.

Explanation

Style sheets are used to define the presentation aspects of a web page, such as the fonts, sizes and colours used for text, headings and hyperlinks and the positioning of elements on the page. Style sheets (CSS) must be used instead of hard coding element properties in the HTML, such as , , <i>, <background colour>, which are deprecated.

Rationale

Using HTML to format a web page can lead to bulky, inefficient and repetitive coding. This is why the W3C has deprecated many of the tags and attributes of HTML used for visual presentation.

Style sheets separate the content from the presentation of a page. This has several advantages, including manageability, portability and accessibility.

Separating content from layout facilitates device independence (examples of devices include screen readers, cell phones, blackberry, PDA, etc.), making it is easier to tailor the presentation to different browsers or user agents. Browsers and assistive technologies that do not support style sheets simply ignore them and access the content directly – making style sheets a good method of ensuring browser independence.

Another advantage to style sheets is that the users can impose their own CSS via their browser. This will display the presentation and content in a way that suits their own preferences (for example, they could change the default colours and font sizes to a setting that they are more comfortable with).

Also, style sheets help management and maintenance by allowing rapid changes to the visual appearance of a website. With style sheets, if there is a requirement to alter the presentation of a site, it can be done once, at the style sheet level, rather than on each individual page at the element level.

Technique

It is not possible to fully document the use of CSS in these guidelines. Documentation and tutorials on this subject are available online.

It is recommended to use a recent version of a robust HTML editor that does not use deprecated HTML elements.

Validation

Using the Firefox Web Developer's Toolbar (plugin):

Go to [Outline](#) > [Outline deprecated elements](#).

Success Criteria

There are no deprecated elements.

Note

You can skip this checkpoint if your page is in XHTML strict and you passed checkpoint 3.2.

Layout Tables

Layout and Data Tables

Layout table: Table used to control the display of elements in a page.

Data table: Table used to display tabular data, which is logically organised into rows and columns, as in an Excel spreadsheet.

WAI Definition (Checkpoint 5.3 and 5.4)

- Do not use tables for layout unless the table makes sense when linearized. Otherwise, if the table does not make sense, provide an alternative equivalent (which may be a linearized version).
- If a table is used for layout, do not use any structural markup for the purpose of visual formatting.

Explanation

Avoid using tables for layout purposes.

Instead, use CSS to position elements on your page. If there is a need to use tables for layout, do not use the “summary” attribute, the <caption> tag or the <th> tags in the table.

Rationale

Tables may be helpful to place and separate elements to the right or left of a page, but they should be used sparingly and not relied on fully, since tables are not well-supported by some screen readers. The screen reader linearizes the content, which means that it reads it row by row. Sometimes, content loses its coherence when read in this manner, especially when there are a lot of nested tables on a page.

The “summary” attribute, the <th> tag and the <caption> tag are meant to assist screen-reader users to make sense of data tables. Letting these users know that a table is for layout purposes is just cumbersome information. Adding <th> tags would be misleading as there are no column or row headers in a layout table.

Technique

Use CSS to position elements on your pages.

If tables must be used to layout the page, do not use nested tables.

The following tags and attributes should NOT appear in a layout table:

- “summary”
- <th>
- <caption>

Validation

Check if there are layout tables on the page.

Using the Firefox Web Developer’s Toolbar (plugin):

Go to Miscellaneous > Linearize page.

Success Criteria

The content of the page should still make sense once it is linearized, appearing in the proper order.

Using the Firefox Web Developer’s Toolbar (plugin):

Go to Information > Table information.

Success Criteria

There should not be any summary, column or row headings and captions.

Using Relative Units

WAI Definition (Checkpoint 3.4)

Use relative rather than absolute units in markup language attribute values and style sheet property values.

Explanation

Define the size of elements with relative units such as “em” and “%”. Do not use absolute or fixed units such as “pixels” or “points”.

Rationale

If the font size is fixed, the user can't scale the text in order to facilitate legibility. If the width of the page is designed to display well in a fixed browser window size and the user enlarges the text, it could become too large to display on screen. This affects users who are viewing content on a small screen, such as a mobile device or laptop, or users with poor eyesight. Relative sizing also means that the website can be used on a wider selection of devices.

Technique

Use CSS to format elements with relative units (em).

Examples

CSS, using “em” to set font sizes

```
h1 {font-size: 3em;}
```

rather than

```
h1 {font-size: 20px;}
```

CSS, using percentages to set width and margins

```
div#main {
  width: 75%;
  margin-left: 15%;
  margin-right: 10%;
}
```

Validation

Using the Firefox Web Developer’s Toolbar (plugin):

Go to [CSS](#) > [View CSS](#).

Success Criteria

All sizes are set in “%” or “em”.

Note

Margin and padding values can be set using absolute units (“px”).

This requirement for relative units does not apply to print style sheets.

Identify the Main Language

WAI Definition (checkpoint 4.3)

Identify the primary language of the document text.

Explanation

The primary language of a document is the natural language in which a document is written.

Rationale

Screen readers need to know the natural language of a document in order for the words on the page to be pronounced correctly. Failing to identify the natural language could cause these devices to mispronounce the information making it impossible to understand. Identifying the natural language also improves search engine indexation.

Technique

The natural language should be identified in the <html> tag with :

- HTML 4.0: the "lang" attribute
- XHTML 1.0: "lang" **and** "xml:lang" attributes.
- XHTML 1.1: "xml:lang" attribute

The language codes for the six UN official languages are:

- Arabic: ar
- Chinese: zh
- English: en
- French: fr
- Russian: ru
- Spanish: es

Refer to the ISO list of language codes for additional languages.

Example

XHTML 1.1

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">  
...HTML document written in French... </html>
```

Validation

Check the source code.

Success Criteria

The proper language code is set for the <html> tag (as seen in the example above in bold).

III - Textual Content

Changes in the Natural Language

WAI Definition (Checkpoint 4.1)

Clearly identify changes in the natural language of a document's text and any text equivalents (e.g., captions).

Explanation

The <html> tag in the header of the page indicates the language of the page's content. If a word or phrase of another language appears in the content, it should be tagged accordingly.

Rationale

Screen readers and Braille devices need to identify changes in the natural language so they can present it with the correct pronunciation.

Technique

Use the "lang" attribute to identify text in another language. See bold example below.

Example

```
<ul>
  <li><a href="sg_letter.html">Letter of the Secretary general</a></li>
  <li><a href="letter_fr.html" lang="fr">Lettre du Représentant permanent
de la Belgique</a></li>
  <li><a href="letter_sc.html">Letter of the President of the Security
Council</a></li>
</ul>
```

Validation

Check the page manually. If there is content in a language other than the main language of the page, check the HTML code.

Success Criteria

Text in a language other than the main language of the page should be enclosed in a tag using the "lang" attribute.

Simple and Clear Language

WAI Definition (Checkpoint 14.1)

Use the clearest and simplest language appropriate for a site's content.

Explanation

The information should be organized in a logical format, offering to the user clear, comprehensible content in simple language.

Rationale

With a simple writing style that avoids jargon you can reach a larger audience. Web users are quick readers. They skim contents easily when it's clearly structured and organized. Furthermore, simple language is also beneficial to the numerous visitors of the UN homepage who are not native speakers of one of the six official languages.

Technique

The Internet encourages the use of less formal styles of presentation and language than are normally used in United Nations materials. Author departments and offices should use concise, direct language in their unofficial materials on the Internet, especially on homepages, aimed at meeting outside user needs for clarity and directness, including those whose primary language is not one of the UN official languages. Slang and idiomatic phrases should be avoided, and abbreviations, acronyms and technical terminology should be explained, used with discretion, and only where widely recognizable.

Validation

There is no recommended tool to check this point, so check the page manually. However, a useful strategy would be to collect feedback among users representative of the site intended audience.

Success Criteria

Sentences are short.

The text is broken down into short paragraphs.

Favour the active voice.

Do not use jargon, slang or idiomatic phrases.

There are links to definitions of technical terms.

Data Tables

Layout and data tables

Layout table: Table used to control the display of elements in a page.

Data table: Table used to display tabular data, which is logically organised into rows and columns, as in an Excel spreadsheet.

WAI Definition (Checkpoints 5.1, 5.2 and 5.5)

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

Explanation

When using tables to display data, it is necessary to identify row and column headers in order to offer a logical flow of navigation. The labels enable a screen reader to read the information in a coherent order.

For data tables that have multiple levels of row or column headers, markup must be used to associate data cells and header cells.

The “summary” attribute describes the table’s purpose structure and purpose to screen-reader users. It is not visible on screen.

Rationale

Tables constitute a complex environment for blind or visually-impaired users because they cannot visualize it globally, which would enable them to understand its structure more easily.

It is therefore necessary to add markup that will guide screen-reader users in the navigation of the table.

Technique

The “summary” attribute is used within the <table> tag to describe the elements and structure of the table. “Summary” attributes belong inside the <table> tag itself.

1. **If the table has one logical level:**
Cells containing columns and/or row headers should be tagged as <th> instead of <td>. The “scope” attribute specifies whether the cell is a column or a row header.
2. **If the table has more than one logical level:**
Consider splitting the table into several simpler tables. If this is not possible, headers and cells should be associated using the “id” and “header” attributes. See example below.

Examples

1. If the table has one logical level

Name	Department	Extension	Room
Peter Johnson	DPI	7.1234	S-1001
Wei Zhang	DPKO	3.9876	DC2-534

This first table would be coded as follows:

```

<table summary="List of staff members with Department, Extension and
room number">
  <tr>
    <th scope="col">Name</th>
    <th scope="col">Department</th>
    <th scope="col">Extension</th>
    <th scope="col">Room</th>
  </tr>
  <tr>
    <td>Peter Johnson</td>
    <td>DPI </td>
    <td>7.1234</td>
    <td>S-1001</td>
  </tr>
  <tr>
    <td>Wei Zhang </td>
    <td>DPKO </td>
    <td>3.9876</td>
    <td>DC2-534</td>
  </tr>
</table>

```

2. If the table has more than one logical level

	Meals	Hotels	Transport	Trip Totals
San Jose				
25-Aug-07	37.74	112.00	45.00	
26-Aug-07	27.28	112.00	45.00	
Subtotals	65.02	224.00	90.00	379.02
Seattle				
27-Aug-07	96.25	109.00	36.00	
28-Aug-07	35.00	109.00	36.00	
subtotals	131.25	218.00	72.00	421.25
Totals	196.27	442.00	162.00	800.27

This second table would be coded as follows:

```

<table summary="This table presents the costs of meals, hotels and
transports fro two trips in San Jose and Seattle">
  <tr>
    <td>&nbsp;</td>
    <th id="meals">Meals</th>
    <th id="hotels">Hotels</th>

```

```

<th id="transport">Transport</th>
<th id="triptotals">Trip Totals</th>
</tr>
<tr>
<th colspan="5" id="san_jose">San Jose</th>
</tr>
<tr>
<td id="aug-25-07" headers="san_jose">25-Aug-07</td>
<td headers="san_jose aug-25-07 meals">37.74</td>
<td headers="san_jose aug-25-07 hotels">112.00</td>
<td headers="san_jose aug-25-07 transport">45.00</td>
<td>&nbsp;</td>
</tr>
<tr>
<td id="aug-26-07" headers="san_jose">26-Aug-07</td>
<td headers="san_jose aug-26-07 meals">27.28</td>
<td headers="san_jose aug-26-07 hotels">112.00</td>
<td headers="san_jose aug-26-07 transport">45.00 </td>
<td>&nbsp;</td>
</tr>
<tr>
<th id="sub" headers="san_jose">Subtotals </th>
<td headers="san_jose sub meals">65.02</td>
<td headers="san_jose sub hotels">224.00</td>
<td headers="san_jose sub transport">90.00</td>
<td headers="san_jose triptotals">379.02</td>
</tr>
<tr>
<th colspan="5" id="Seattle">Seattle</th>
</tr>
<tr>
<td id="aug-27-07" headers="Seattle">27-Aug-07</td>
<td headers="Seattle aug-27-07 meals">96.25</td>
<td headers="Seattle aug-27-07 hotels">109.00</td>
<td headers="Seattle aug-27-07 transport">36.00</td>
<td>&nbsp;</td>
</tr>
<tr>
<td id="aug-28-07" headers="Seattle">28-Aug-07</td>
<td headers="Seattle aug-28-07 meals">35.00</td>
<td headers="Seattle aug-28-07 hotels">109.00</td>
<td headers="Seattle aug-28-07 transport">36.00 </td>
<td>&nbsp;</td>
</tr>
<tr>
<th id="sub2" headers="Seattle">subtotals</th>

```

```

        <td headers="Seattle sub meals">131.25</td>
        <td headers="Seattle sub hotels">218.00</td>
        <td headers="Seattle sub transport">72.00</td>
        <td headers="Seattle triptotals">421.25</td>
    </tr>
    <tr>
        <th id="grandtotals">Totals</th>
        <td headers="grandtotals meals">196.27</td>
        <td headers="grandtotals hotels">442.00</td>
        <td headers="grandtotals transport">162.00</td>
        <td headers="grandtotals triptotals">800.27</td>
    </tr>
</table>

```

Validation

1. If the table has one logical level:

Using the Firefox Web Developer's Toolbar (plugin):

Go to **Information** > **Display table information**

Success Criteria

Data tables have column and/or row titles (tagged as <th> instead of <td>), and a "summary" attribute, describing the structure of the table.

2. If the table has more than one logical level

Check the code of the page

Success Criteria

Headers and cells are associated using the "id" and "header" attributes.

Titles

WAI Definition (Checkpoint 3.5 and 12.3)

- Use header elements to convey document structure and use them according to specification.
- Divide large blocks of information into more manageable groups where natural and appropriate.

Explanation

Documents are often divided into a variety of sections; those sections should be introduced with the HTML heading elements (<h1> through <h6>).

It is important to use headings appropriately to convey document structure. For example, an <h2> is the subtitle of an <h1>; a <h3> is the subtitle of an <h2>, etc.

Content developers should not skip levels (e.g., <h1> directly to <h3>).

They should not use headings to create formatting effects (use CSS style sheets to change font styles instead).

Rationale

Categorizing the information makes it easier for the average user to visually scan the content. Large blocks of information are not very reader-friendly and therefore all attempts should be made to break the information down into smaller sections, where possible.

Furthermore, breaking up the text into smaller sections and sub-sections allows screen-reader users to scan the page. Screen readers can bounce from title to title, enabling the user to skim through the information rather than having to go through all of the content word by word.

Some screen readers will read content marked as a header in a tone of voice different from the one used for other content on the page, providing users with important clues about the document structure. It is therefore important to use heading tags according to specification to give visually-impaired users a better understanding of the structure of a document or page. Misusing heading attributes would misinform and confuse screen-reader users.

Technique

The following HTML markup shows how heading tags may be used to properly render the structure of a document:

Example

```
<body>
  <div class="section2">
    <h1>Welcome to the United Nations!</h1>
    ... some text here ...
  </div>
  <div class="section2">
    <h2>sub topic 1</h2>
    ... text of the section ...
  </div>
  <div class="section3">
    <h2>sub topic 2</h2>
    ... text of the section ...
  </div>
</body>
```

Validation

Using the Firefox Web Developer's Toolbar (plugin):

Go to **Outline** > **Headings**.

Make sure the option "Show element name when outlining" is checked (in the Outline menu).

Success Criteria

The page should have titles and subtitles.

All titles should be outlined and the hierarchy between heading tags should be consistent with the hierarchy of the content.

Lists**WAI Definition (Checkpoint 3.6)**

Mark up lists and list items properly.

Explanation

List markup should be used to aggregate related items. List markup should not be used for the sake of adding a single bullet or number item to text or for indentation purposes.

Lists should not be created by using tables with bullet images in one column and text in the other column. To achieve formatting effects, use CSS.

When an element of the list has sub-items, the structure of your list should be reflected in the HTML markup, not only in the styling effect by creating a nested list (a list within a list). See example.

Rationale

List structure and list elements (for example, `<dl>`, `` and ``) should be used only to create lists. These features should not be used to provide formatting effects such as indentation, as it may affect the user's understanding of the page structure.

Although users with graphical browsers can understand the structure of nested lists from their indentation, information can be lost on non-visual user agents. Users accessing the site using these user agents could 'get lost' in lists – particularly in lists that do not indicate the specific nest level for each list item.

Technique

Listed elements should be text based. List tags should not be used for formatting effects, as this can confuse screen-reader users.

Note on nested lists

The closing tag of the main list item `` is placed **after** the nested list. See example.

Example

HTML

```

<ul>
  <li>Our mandate</li>
  <li>Documents </li>
  <li>News
    <ul>
      <li>News Centre </li>
      <li>Press Releases</li>
      <li>Radio</li>
    </ul>
  </li>
  <li>Links </li>
  <li>UN Home</li>
</ul>

```

Text version

- Our mandate
- Documents
- News
 - News Centre
 - Press Releases
 - Radio
- Links
- UN Home

Validation

Using the Firefox Web Developer’s Toolbar (plugin):

Go to **Outline** > **Outline blocks elements**.

Make sure the option “Show element name when outlining” is checked (in the Outline menu).

Success Criteria

Lists (such as menus or lists of links or documents) should be coded as lists. If they are nested, they should be properly coded as nested lists.

Quotations

WAI Definition (Checkpoint 3.7)

Mark up quotations. Do not use quotation markup for formatting effects such as indentation.

Explanation

Use the <q> or <blockquote> tags to mark up quotations.

If you want to achieve indentation effects, use margins or padding in CSS.

Rationale

Some screen readers identify text that has been correctly marked as a quotation differently, which gives people who use these technologies important information about the context of the text.

Technique

Short quotations of 10 words or less should use the `<q>` tag for markup. Long quotations of 11 words or more should use the `<blockquote>` tag.

Example

```
<blockquote>I feel honoured and very humbled when I think of all our founding
fathers wise enough and courageous enough to save this world from the scourge
of war who have negotiated, drafted and finally signed the Charter of the United
Nations, which has shaped the future of the whole international community.
</blockquote>
<p>Secretary-General Ban Ki-moon in remarks at San Francisco War Memorial,
26 July 2007</p>
```

To indent a paragraph without using the `<blockquote>` tag, use the following combination of CSS and HTML code:

CSS

```
p.indent {margin-left:2em;}
```

HTML

```
<p class="indent">This paragraph will have a left indent.</p>
```

Validation

Using the Firefox Web Developer's Toolbar (plugin):

Go to **Outline** > **Outline block elements**.

Make sure the option "Show element name when outlining" is checked (in the Outline menu).

Success Criteria

The `<blockquote>` and `<q>` tags are to be used for quotations only.

Metadata

WAI Definition (Checkpoint 13.2)

Provide metadata to add semantic information to pages and sites.

Explanation

Metadata is information that describes the content of a page or a website, in a standardized format. The recommended metadata format for the website is the Dublin Core (see example below).

Rationale

Good quality metadata improves usability of search facilities by providing useful and accurate keywords and descriptions of content and media types. The descriptions and summaries of page contents displayed in search results are often drawn from metadata which is inserted in web pages.

Example

```
<meta name="dc.title" content="The United Nations Programme on Disabilities" />
<meta name="description" content="The United Nations Programme on Disabilities works to assist in the promotion of effective measures for prevention of disability, rehabilitation, and the realization of the goals of full participation of disabled persons in social life and development, and of equality." />
<meta name="dc.subject" content="disability, accessibility, enable, persons with disabilities, rehabilitation, prevention, handicap, human rights, convention, international day of disabled persons" />
<meta name="dc.publisher" content="United Nations" />
<meta name="dc.format" content="text/html" />
<meta name="dc.language" content="EN" />
<meta name="dc.rights" content=" United Nations" />
```

Validation

Using the Firefox Web Developer's Toolbar (plugin):

Go to **Information > View Metatags information.**

Success Criteria

Relevant metadata should be provided. It is specific and accurately describes the content of the page / site.

Acronyms and Abbreviations

WAI Definition (Checkpoint 4.2)

Specify the expansion of each abbreviation or acronym in a document where it first occurs.

Explanation

The <abbr> (abbreviation) and <acronym> tags should be used with the "title" attribute to provide explanation of the meaning of the abbreviation or acronym text in the document.

Rationale

This aids understanding by novice users and facilitates reading by screen readers which can read out the full title of the text.

Technique

When using acronyms (such as “UN”, “GA”, “SC”) or abbreviations (“Add.”), markup the acronyms and abbreviations respectively with the acronym or the abbreviation tags, and use a title attribute to elaborate on the acronym or abbreviation letters. Remember that the explanation of the acronym or abbreviation will be read as you set it in the “title” attribute; it should therefore be written as it would normally appear in running text.

Example

HTML

```
<p>The Intergovernmental Panel on Climate Change has been established by
<acronym title="the World Meteorological Organization">WMO</acronym>.
</p>
<p><abbr title="Mister">Mr.</abbr> Ban Ki-moon, <acronym title="United
Nations">UN</acronym> Secretary-General</p>
```

Acronym and abbreviation tags can be styled using a style sheet:

CSS

```
acronym {
    cursor: help;
    border-bottom: 1px dotted black;
}
abbr {
    cursor: help;
    border-bottom: 1px dotted black;
}
```

Validation

Using the Firefox Web Developer’s Toolbar (plugin):

Go to **Information** > **Display abbreviation**.

This will display the title attribute of each abbreviation and acronym.

Success Criteria

Each abbreviation or acronym is marked as such, and the “title” attribute describes its meaning, as it would normally appear in running text.

IV - Navigation

Meaningful Link Titles

WAI Definition (Checkpoint 13.1)

Clearly identify the target of each link.

Explanation

The text hyperlinked should make enough sense for users to understand it if read without context. Users should have a relatively clear idea of where the link points to. Links such as “click here”, “read more” or “more information” should not be used.

Rationale

Screen-reader users will often review the list of links on a page before taking the time to listen to the whole content. This is the equivalent of visually scanning a page to get an overview of its content and the available options. Because links are read out of context, it is important to provide link titles that do not require the user to read the surrounding information.

Applying this technique also helps to optimize the content for search engines.

Technique

Use the title of the target page as link text instead of using generic wording such as “read more” or “click here”.

Where possible, do not place the link only on a document symbol, but include the document title too.

Validation

Using the Firefox Web Developer’s Toolbar (plugin):

Go to **Information** > **View Page Information** and click on the link tab.

Success Criteria

All links marked as “anchors” should make sense when read out of context.

There are no two links with similar text, pointing to a different target. Every link text should be unique to its destination.

Sitemap

WAI Definition (Checkpoint 13.3)

Provide information about the general layout of a site (e.g., a site map or table of contents).

Explanation

Provide an overview of the site's structure which summarizes the contents of the site.

Rationale

An overview of the site's structure and content makes navigation easier. For example, providing a site map or table of contents allows the user to browse through key sections of the site without having to navigate through each individual page. This is also very helpful for people using assistive technologies, such as a screen reader. For example, the screen reader can scan the site map, giving the user an overall idea of the contents so that he/she can rapidly determine the location of the information wanted and go directly to that page.

A table of contents or site map can assist people who have become confused, disoriented or have gotten lost during the navigation process.

Technique

There should be a sitemap on any site with pages that cannot be accessed from the navigation bar's default state or the homepage.

Validation

Check the site.

Success Criteria

A site map or table of contents exists if the site has pages that cannot be accessed from the navigation bar's default state or the homepage.

Avoiding Pop-ups and Links Opening in New Windows**WAI Definition (Checkpoint 10.1)**

Do not generate pop-ups or new windows and do not change the current window without informing the user.

Explanation

Do not create pop-ups (windows that automatically open).

Do not open links in new windows without informing the users.

Rationale

Pop-ups can be very frustrating for users and many people find them annoying. Pop-ups can also erode a user's trust in a site as it affects their control over the navigation and interaction.

Furthermore, pop-ups are known to cause problems for screen-reader users because they change the focus without warning. This means that the screen reader

begins reading the content of the new window, usually without warning the user that a new window has opened. If the new window displays content from another site, the user can become very disoriented and confused.

Pop-up windows used to display an error message in a form can cause problems too. The user has to read the content of the pop-up window, navigate back to the main page, and then find the relevant position on the form again before they can fix the error.

Technique

Avoid opening links in new windows.

If you open a link a new window, add a warning in the link title, including a message like "(this will open in a new window)". Including the warning as part of the hyperlink makes the link title more meaningful when read out of context, e.g. it will be read by screen-reader users when they call up a list of links on the page.

Example

HTML

```
<a href="un.org" target="_blank">United Nations homepage (this link will open in a new window)</a>
```

Validation

Using the Firefox Web Developer's Toolbar (plugin):

Go to **Information** > **View Link Information**.

Success Criteria

Links should not have the attribute 'target="_blank"'. If they do, the users should be warned that the link will open in a new window.

Consistent Navigation

WAI Definition (Checkpoints 13.4 and 13.5)

- Use navigation mechanisms in a consistent manner.
- Provide navigation bars to highlight and give access to the navigation mechanism.

Explanation

Navigation mechanisms direct the user to key sections and/or pages of the site. These mechanisms should be designed to clearly state their destinations and to indicate the current position of the user. They should also remain consistent visually and structurally from page to page.

Rationale

The navigation mechanisms should be presented in a uniform, well-structured and logical order so as to avoid confusion or disorientation.

Technique

In order to ensure a uniform site, you should:

- establish a consistent visual presentation from page to page;
- maintain a constant sequential order;
- maintain language and terminology consistency;
- maintain consistency in the location of navigation controls.

This provides a smooth transition from page to page, enabling the user to navigate easily and quickly.

The navigation menu should always include a link to the site's homepage.

Breadcrumbs

Sites with complex architecture can also benefit from breadcrumbs. They typically appear horizontally across the top of a web page, usually below any title bars or headers. They provide links back to each previous page that the user navigated through in order to get to the current page. In hierarchical structures, it is often the parent of the current page. Breadcrumbs provide a trail for the user to follow back to the starting/entry point of a website and an indication of where they are on the site. They may look something like this:

Home page → Section page → Subsection page

Validation

Look at the page.

Success Criteria

As you navigate to each page, the following elements should be consistent:

- visual presentation;
- sequential order;
- language and terminology;
- location of navigation controls.

Logical Tab Order

WAI Definition (Checkpoint 9.4)

Create a logical tab order through links, form controls, and objects.

For example, in HTML, specify tab order via the “tabindex” attribute or ensure a logical page design.

Explanation

In order to enable the user to use the “tab” key for navigation, you must form a logical order for the tabbing. Normally, the “tab” key will start at the top and works its way down the page. However, there are instances where the “tab” key would require guidance as to which location to jump to next, such as in form fields (see the example below).

Rationale

Many users are unable to use a mouse for navigation. A logical tab order offers the user ease of navigation throughout the site using the “tab” key.

Technique

In the next example, a logical tab order is specified among elements (in order, “field2”, “field1”, “submit”) with “tabindex”.

Example

```
<form action="submit" method="post">
  <label for="field1">Field1:</label>
  <input tabindex="2" type="text" name="field1" id="field1">
  <label for="field2">Field2:</label2>
  <input tabindex="1" type="text" name="field2">
  <input tabindex="3" type="submit" name="submit">
</form>
```

Validation

Use the “tab” key to navigate the site.

Success Criteria

Ensure that the movement to each location follows a logical order.

Provide Information About Documents

WAI Definition (Checkpoint 11.3)

Provide information to enable users to receive documents according to their preferences.

Explanation

When linking to non HTML documents, provide users with information on the document file type, size, language and available alternative formats.

Rationale

Users who want to download files or view pages (such as documents or multimedia files) will want to select versions compatible with their systems and/or human language. Users are very reluctant to commit to downloading files if they can't tell whether they will be able to open or read them. Users with text-

only browsers or screen readers may prefer to use pages that are optimized for certain user agents.

Technique

1. **Specify file type and size with different file types indicated in the links which activate the downloads**
Providing users with details of the file type, size and language in the link title ensures that users are downloading the files which best suit their platform and language.
2. **Provide available links to alternative presentations**
Whenever alternatives are available, provide clear links to the alternatives (Example provide links to different media types or different language options for the same document, if possible).
3. **Consider using profiles to record user preferences and configure the server to choose document preferences automatically**
For example, a browser could indicate that it would like to see information in Arabic or English based on user's settings; provide users with the options to reset their settings.

Example

```
<!-- clearly specify the file type, size and/or language -->
<a href="event.mov">Download the video of the event (Quick Time format,
3MB, 12mn)</a>
or
<a href="event.mp3">Download the audio recording of the event (MP3 format,
112KB, 12mn)</a>

<p>Choose the language of the document :</p>
<ul>
  <li><a href="English_file.html">English (HTML format)</a></li>
  <li><a lang="ar" href="Arabic_file.pdf">عربي(PDF 3MB)</a></li>
</ul>
```

Validation

Look at the page and check if there are links to downloadable files.

Success Criteria

Links to downloadable files provide users with details about the file (format, size or length, language, etc...)

Skip Links

WAI Definition (Checkpoint 13.6)

Group related links, identify the group (for user agents), and, until user agents do so, provide a way to bypass the group.

Explanation

Group related links, such as the navigation links, in a list. Add “Skip links”. They are anchor links that point to the menu and content of the pages.

They allow users to bypass content recurring on every page to access the menu or the content directly.

Rationale

A navigation bar is often the first element presented on a page, after the title and banner elements. Screen-reader users have to listen to this over and over again, for every page, until they get to the actual content.

Similarly, keyboard users have to tab through all the navigation links before they can access any links in the content.

Providing a way to skip over repetitive links allow users to access content more quickly.

Technique

In the example below, there are two skip links. One skips the banner title to access the navigation directly. The second one skips the banner and the navigation to point directly to the content.

Example

HTML

```
<body>
  <div id="skiplink">
    <ul>
      <li><a href="#navigation">Skip to navigation</a></li>
      <li><a href="#content">Skip to content</a></li>
    </ul>
  </div>
  <div id="banner">
    <h1>Title of this page</h1>
  </div>
  <div id="navigation">
    <ul>
      <li id="home"><a href="index.shtml">Menu item 1</a></li>
      <li id="about"><a href="about.shtml">Menu item 2</li>
    </ul>
  </div>
  <div id="content" class>
    <p>Unique content of the page</p>
  </div>
```

CSS

You can use CSS to hide the skip links by giving them the same colour as the background. Keep in mind however that skip links are useful to sighted users who are unable to use a mouse. Therefore, skip links should become visible upon keyboard focus. Below is the CSS code to achieve this.

```
body {background-colour: #FFFFFF;}  
#skiplink a {color:#FFFFFF;}  
#skiplink a:focus, div#skiplink a:active {color:#0000FF;}
```

Validation

Check the code to ensure that the page provides skip links to bypass recurring content and access the menu(s) and the content directly.

Success Criteria

The page provides skip links to bypass recurring content and access the menu(s) and the content directly.

V - Scripts and Forms

Pages Are Usable When Programmatic Elements Are Not Supported

WAI Definition (checkpoint 6.3)

Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative, accessible page.

Explanation

Scripts (such as JavaScript) are often used to dynamically create or display the content that appears on a web page, for example, to validate form entries or to display menus.

If the scripts are not supported (for instance, JavaScript is deactivated), the content and functionality of the page should still be available.

Scripting as referred to in the section is client-side scripting.

Rationale

Some browser configurations or some organizations that have proxies may turn off the client-side scripting. Old browsers may not support scripting. Dynamic drop-down menus, in particular, are known to cause significant accessibility problems for people with motor or visual impairments, for example when screen magnifiers are used. Therefore, they should not be used.

Applets and/or support for applets (particularly Java) may require downloads before running. Download time with slow connection and security issues can also present a problem for usage of applets.

Techniques

- Do not use JavaScript as a link target. A user agent that does not support JavaScript will not be able to follow the link,
- Use the `<noscript>` tag. User agents that do not support scripting will display the contents of the `<noscript>` tag.
- Use server-side scripts instead of client-side scripts. If you use a client-side script and the client does not support this technology, the user will not be able to access the content or features provided by the script.. But in many cases, the same functionality can be achieved using a server-side script.
- As a last resort, provide an up to date text equivalent of the information on an alternative page.

Validation

Using the Firefox Web Developer's Toolbar (plugin):

Go to **Disable** > **Disable JavaScript** > **All JavaScript**; then select **Disable Java**

Success Criteria

The page is still usable. There is no critical function missing (check especially that the menu remains visible and that forms can still be submitted).

Accessible Dynamic Content

WAI Definition (Checkpoint 6.5)

Ensure that dynamic content is accessible or provide an alternative presentation or page.

Explanation

Client-side dynamic content is information on the page that is updated in real time. A Java applet continually displaying stock exchange news is a good example of dynamic content.

Dynamic content must be accessible using assistive technologies. If this cannot be achieved, provide an alternative page that is accessible.

Rationale

Obviously, not all users will be able to view the appropriate content – either because they do not have the relevant plug-in enabled (through choice or because they are unable to), or because they have a vision impairment and cannot read and follow rapidly changing content.

If a web page contains dynamic information, it should be delivered in an accessible manner. If this cannot be achieved, visitors should be able to navigate to another page where the same information is presented in an accessible format.

Database-driven dynamic pages created with server-side scripting (such as ASP or PHP) are not affected by this checkpoint if the information on the page does not change after it is served.

Technique

Write scripts that transform gracefully: if the script is not supported, the information should still appear in another accessible format.

If you cannot make a dynamic page element accessible, you must create an alternative page which is updated as often as the original.

Validation

Using the Firefox Web Developer's Toolbar (plugin):

Go to **Disable** > **Disable JavaScript** > **All JavaScript**; then select **Disable Java**

Success Criteria

The page is still usable. There is no critical function missing (check especially that the menu remains visible, and that the forms can still be submitted).

No Auto-Refresh

WAI Definition (Checkpoint 7.4)

Until user agents provide the ability to stop the refresh, do not create periodically auto-refreshing pages.

Explanation

You should not create pages that automatically reload in the user's browser every so often, as long as the browser window is open, since not all user agents allow the user to disable auto-refresh.

User agents are not limited to web browsers. They also include graphical browsers, text-only browsers, voice browsers, mobile phones, multimedia players, plug-ins and assistive technologies such as screen readers, screen magnifiers and voice recognition software.

Rationale

Not all users (particularly those who are new to, or not comfortable with, computers) will understand what is happening when a page auto-refreshes.

Auto-refreshing can also make it difficult for the user to navigate around the site. People tend to read at varying speeds and some users may not have finished reading the content before the page automatically refreshes or a screen reader may only be halfway through reading a page before it changes (which may cause the assistive technology to re-read the page from the beginning).

Some older screen readers cannot handle auto-refreshing content and may not display any content at all.

Technique

Do not use the <meta> element for auto-refresh or auto-redirect.

Example

Do not use the following:

```
<meta http-equiv="refresh" content="60">
<body>
<p>...information...</p>
</body>
```

Validation

Check the page code.

Success Criteria

There is no `<meta http-equiv="refresh">`. If there is an automated refresh, there is a mechanism to control it.

No Client-Side Automatic Redirects**WAI Definition (Checkpoint 7.5)**

Until user agents provide the ability to stop auto-redirect, do not use markup to redirect pages automatically.

Explanation

Do not automatically redirect pages with client-side technologies (technologies that should be executed by the browser). Such redirecting techniques include using the `<meta>` element or JavaScript.

Rationale

Automatically redirecting to a new page may be confusing for users who are not very familiar with using the Internet. Many people can feel lost because it takes the control over the navigation away from users. It is particularly disorienting for screen-reader users.

Furthermore, if the client-side technique for the redirect is not supported, the user may not be able to access the new content.

Technique

- Configure the web server to redirect traffic to a new location instead of using `<meta>` header redirects (consult with your server administrator on the best practice for this approach).
- You can also create redirects that are executed on the server side, for instance in ASP or PHP.
- You can replace the redirected page with a static page containing a normal link to the new page.

Validation

Check the page code.

Success Criteria

There is no `<meta http-equiv="refresh">` or JavaScript automatic redirect.

Labels and Forms

WAI Definition (Checkpoint 12.4)

Associate labels explicitly with their controls.

Explanation

In web forms, form fields should be associated with labels to describe their purpose, using structural markup.

Rationale

It is good practice to position labels so that the visual relationship with corresponding form controls is clear.

However, if the relationship between labels and form controls is only visible, but not explicitly defined in the underlying HTML, older browsers and screen-reader users may not understand the form and the purpose of each field.

This does not apply to hidden fields.

Technique

Every descriptive label (description of the purpose of the field) should be tagged as `<label>` and associated with the name of the field. The “for” attribute of the `<label>` tag is used by modern screen readers to identify a field reached by tabbing. The field should have the corresponding “id” attribute.

Without this, tabbing between fields is completely disorienting.

Example

XHTML

```
<label for="fname">Name</label>
<input type="text" name="name" id="fname" tabindex="1" />
```

Validation

Using the Firefox Web Developer’s Toolbar (plugin):

Go to **Form** > **View form information**.

Success Criteria

Each visible field is associated with a descriptive label.

Device Independence

Definition (Checkpoints 6.4, 8.1, 9.2, 9.3)

Ensure that user interfaces are device-independent and use logical device-independent event handlers. Ensure that scripts and applets are accessible.

Explanation

Ensure that it is possible to logically interact with the site using keyboard and/or mouse input/output methods. Scripts and applets are types of programmatic objects that should either be inherently accessible or should include features that are usable with assistive technologies.

Rationale

Interfaces which do not provide flexibility in the type of device which the user relies on to input information are inherently inaccessible. For example, a laptop user may choose to work without a mouse. If this were the case and interactive features on the site relied on "drag and drop" interactivity as the only means of interaction, the site would be unusable. Additionally, if the site were delivered through a kiosk or public access terminal with a touch screen interface, it would be unusable.

Screen-reader users, on the other hand, rely entirely on the keyboard for interacting with websites. Failing to support the keyboard as an input device will make the site unusable to them. Accessibility features which are provided in programming technologies and supported by assistive technologies should be included in programmatic elements.

Technique

A. If you must use device-dependent attributes, provide redundant input mechanisms.

In JavaScripts and HTML 4.01, application-level event attributes are:

1. "onfocus" (the event occurs when an object gets focus): this will highlight the form field when the event is tabbed into or clicked on.
2. "onblur" (the opposite of "onfocus" where the event occurs when an object loses focus)
3. "onselect" (the event occurs when text is selected in a "text" or "textarea" field)

Note that the above attributes are designed to be device-independent, but are implemented as keyboard specific events in some browsers. Otherwise, if you must use device-dependent attributes, provide redundant input mechanisms (i.e., specify two handlers for the same element):

1. "onmousedown" (the event occurs when a mouse button is clicked) along with "onkeydown" (the event occurs when a keyboard key is pressed): the event will be triggered when the user tabs or clicks on a specific field
2. "onmouseup" (the event occurs when a mouse button is released) and "onkeyup" (the event occurs when a keyboard key is released)
3. "onclick" (the event occurs when an object gets clicked) along with "onkeypress" (the event occurs when a keyboard key is pressed or held down)

Notes

- On links and form controls only, the “onclick” event can be used without the redundant “onkeypress” event.
- There is no keyboard equivalent to double-clicking (“ondblclick”).

B. Do not write event handlers that rely on mouse coordinates.

This is bad practice because it relies on the use of a mouse and a highly visual style of interaction to invoke an event handler. Event handlers that rely on mouse coordinates don't support keyboard input.

C. If you use programmatic elements such as Java applets and Flash, consult the accessibility guidelines from the application vendor.

Example

The following example illustrates how to provide a redundant input mechanism in JavaScript.

HTML

```
<form>
  <p>You will you go <label>back to the previous page
    <input name="Back to Previous Page" type="button" id="Back to
Previous Page" onClick="JavaScript:history.go(-1)"
onKeyPress="JavaScript:history.go(-1)" value="Back to Previous
Page"></label> if you PRESS (keyboard) or CLICK (mouse) on the button </p>
  <noscript>
    <p><a href="index.html">Go back to previous page</a></p>
  </noscript>
</form>
```

Validation

Use programmatic elements with keyboard-only and mouse-only navigation. Try to operate it using only the keyboard and using only the mouse.

Success Criteria

All functionalities should be reached and properly controlled.

VI - Accessible Alternatives

The principle of inclusion in web accessibility

‘Full and effective participation and inclusion in society’ is one of the general principles that underlie the Convention on the Rights of Persons with Disabilities and each one of its articles. The goal of inclusion signifies that there is an obligation to create environments that **provide access to all aspects for all people on an equal basis**.

The concept of inclusion is linked with “universal design”, which is defined in the Convention as “the design of products, environments, programmes and services to be **usable by all people, to the greatest extent possible, without the need for adaptation or specialized design**”.

Therefore, every effort should be made to create universally accessible pages. The use of alternative, accessible versions should be kept to a minimum, as a last resort.

Up-to-Date Equivalents

WAI Definition (Checkpoint 6.2 and 11.4)

- If, after best efforts, you cannot create an accessible page, provide a link to an alternative page that uses W3C technologies, is accessible, has equivalent information (or functionality), and is updated as often as the inaccessible (original) page.
- Ensure that equivalents for dynamic content are updated when the dynamic content changes.

Explanation

If an accessible page cannot be created, provide a link to an alternative, accessible page which uses W3C technologies, containing the same information (or functionality), and is updated as often as the inaccessible (original) page. The United Nations strives to have an inclusive approach for disabled persons. Therefore, every effort should be made so that the majority of pages are accessible. The use of an alternative, accessible page should be reserved to specific cases where the content cannot be made accessible directly.

As dynamic content is updated, ensure that the “equivalent” for the dynamic content is updated simultaneously.

Rationale

When content is delivered with technologies that may not be accessible to all, it is important to offer an alternative, accessible presentation of this content, which is kept up-to-date so that users do not access outdated information.

Technique

- **Provide a clear link to alternative version(s) at the top of each page.**
Providing the link at the top of the page means that the user does not have to wade through inaccessible content to find the alternative version. You should also provide a link to allow the user to navigate directly back to the original version.
- **Use scripts to automatically detect different browsers and present appropriate versions of web pages.**
A 'browser detect' is a script which can detect the type of browser used by a visitor to a site. If a browser detect is used in combination with alternative versions, it is possible to present the user with a version of the page which works well in their browser. Note that it is not possible to detect if the user is using a screen reader because it is not a browser.
- **Use server-side scripts to generate alternative pages on demand.**
Server-side scripts, such as Java servlets, or PHP, can be used to create alternative presentations of a page if the user requests one through the browser. The benefit of providing alternative pages in this way is that there is no requirement to maintain different 'versions' of a website, which reduces the maintenance effort and ensures that up-to-date content is delivered to the user, regardless of the version selected.

Validation

Check the content of the alternative page.

Success Criteria

The content of the alternative page is equivalent to the content of the dynamic page.

HTML Alternatives for Non-W3C Technologies

WAI Definition (Checkpoint 11.1)

Use W3C technologies when they are available and appropriate for a task and use the latest versions when supported.

Explanation

Publish web content in HTML format. If you publish web content using another format, then you should provide an alternative HTML version. This applies in particular to Flash and PDF formats.

The official UN document hosted on the ODS are an exception to this rule. It is allowed to link to those documents without re-creating an HTML version, even though the official document will not be in an accessible format.

Rationale

Creating accessible documents in PDF or Flash formats raises a number of different accessibility issues. At this point, the W3C has not yet issued accessibility guidelines for these types of formats.

Validation

Check the site for non-HTML elements.

Success Criteria

If content is published in a non-HTML format, such as Flash or PDF, an accessible, up-to-date HTML format version should be provided.

Note

Does not apply to ODS documents.

Annex

The following validation procedures use the Firefox add-ons:

- Web Developer Toolbar
- Juicy Studio Colour Contrast Analyser

Other tools with similar functionalities could be used successfully to complete the evaluation.

Validation Procedure for Designers and Developers

More on page...	WAI Checkpoint	Test	Success Criteria
5	1.1	Under Images, select <u>Outline images without “alt” attribute</u> Under Images, select <u>Replace images with alt</u> attributes	No image exists without an “alt” attribute. The information conveyed through the use of images is replaced by descriptive “alt” attribute. No information is missing. There are no “alt” attribute with meaningless information (such as “spacer” or “blue line”). “Meaningless information” is considered information that is not fully descriptive of the element.
6 and 8	2.1 and 3.1	Under Disable, select <u>Disable page colours</u> Under Images, select <u>Disable all images</u>	There is no critical information missing when colours and/or images are disabled. Links in the text are clearly identified and underlined.
7	2.2	Colour contrast requirements can be tested using “Juicy Studio Color Analyzer” at: http://juicystudio.com/services/luminositycontrastratio.php#specify For text and images with flat colours: In Firefox, under Tool, select	All the colour combinations in the page pass level AAA or AA for regular text size.

		<u>Colour Contrast Analyzer</u> > <u>Luminosity Contrast Ratio</u> For images without flat colours: Run the image through http://www.vischeck.com/vischeck/ to simulate how your document will look if viewed by a colour-blind person	The information conveyed by the image is still understandable.
9	7.1, 7.2 and 7.3	Check the page	There are no brusque changes in luminosity or flash effects. Ensure that the page does not contain elements that flicker, blink or move.
11	6.1	Under CSS, select <u>Disable styles</u> / <u>All styles</u>	The document is still readable. Elements such as headings or lists can be recognised as such.
11	12.1 and 12.2	Check the code	There are no frames. Or, if there are, the content in the <noframe> tag provides an accurate description of each frames as well as links to access the content of the frames.
13	3.2	In Firefox Development Toolbar, validate the HTML (Tools > <u>Validate HTML</u>) and the CSS (Tools > <u>Validate CSS</u>) ² .	The HTML and CSS codes pass validation. No errors are found.
14	3.3 and 11.2	In Firefox Developer Toolbar, Under Outline, <u>Outline deprecated elements</u> . ³	There are no deprecated elements.
16	3.4	In Firefox Developer Toolbar, under CSS, select <u>View CSS</u> . ⁴	All sizes are set in “em” or “%” Note Margin and padding values can be set using absolute units (“px”)
17	4.3	Check the code manually	The proper language code is set for the <html> tag.
24	3.5 and 12.3	In Firefox Development toolbal, select <u>Outline</u> > <u>Block Level Elements</u> ⁵ .	The page should have titles and subtitles. All titles should be outlined and the

² If your server cannot be accessed by the W3C validator in a test or intranet environment, in Tools, select Display Page Validation and check the icons in the toolbar, or in Tools, select Validate Local CSS and Validate Local HTML.

³ You can skip this checkpoint is your page is in XHTML strict and you passed checkpoint 3.2

⁴ Does not apply to print style sheets. Optional for padding and margin values.

26	3.6		hierarchy between heading tags should be consistent with the hierarchy of the content.
27	3.7		Lists (such as menus or lists of links or documents) should be coded as lists. If there are nested, they should be coded as nested lists. The <blockquote> and <q> tags are used for quotations only.
31	13.1	In Firefox Developer Toolbar, under Information, select <u>View Page Information</u> and click on the link tab.	All links marked as “anchors” should make sense when read out of context. There are no two links with similar text, pointing to a different target. Every link text should be unique to its destination.
31	13.3	Check the site	A site map or table of contents exists if the site has pages that cannot be accessed from the navigation bar’s default state or the homepage. ⁶
32	10.1	In Firefox Development Toolbar, under Information, select View Link Information.	Links should not have the attribute ‘target=”_blank”’. If they do, the users should be warned that the link will open in a new window.
33	13.4 and 13.5	Manual	As you navigate to each page, the following elements should be consistent: visual presentation; sequential order; language and terminology; location of navigation controls.
34	6.4, 8.1, 9.2, 9.3, and	Manual	All features of the site can be accessed

⁵ Make sure the option “Show element name when outlining” is checked (in the Outline menu)

⁶ The provision of a site map can be the responsibility of the design and development team or of the content provider, depending on the tools used to publish content and their ability to automatically generate a sitemap.

	9.4		through keyboard navigation (tab key). Forms and menus can be navigated in a logical order.
36	13.6	Manual	The page provides skip links to bypass recurring content and access the menu(s) and the content directly.

If the site uses layout tables:

More on page...	WAI Checkpoint	Test	Success Criteria
15	5.3 and 5.4	Under Information, select <u>Display table information</u> In Firefox Developer Toolbar, select <u>Miscellaneous</u> > <u>Linearize page</u> .	There should not be any summary, column or row headings and captions.. The content of the page should still make sense once it is linearized, appearing in the proper order

If the site uses JavaScript or Java applets

More on page...	WAI Checkpoint	Test	Success Criteria
39 and 40	6.3 6.5	Under Disable, select <u>Disable JavaScript</u> / <u>All JavaScript</u> Select <u>Disable Java</u>	The page is still usable. There is no critical function missing (check especially that the menu remains visible and that forms can be submitted).

If the site redirects or refreshes the content

More on page...	WAI Checkpoint	Test	Success Criteria
41 and 42	7.4 and 7.5	Check the code	There are no <meta http-equiv="refresh"> or JavaScript automatic redirect. Users can control when the page is refreshed or redirected.

If the site uses forms:

More on page...	WAI Checkpoint	Test	Success Criteria
43	12.4	Under Firefox Developer Toolbar, select <u>Form</u> > <u>View Form Information</u> .	Each visible field is associated with a descriptive label.

If the site provides alternative page(s) for dynamic pages

More on page...	WAI Checkpoint	Test	Success Criteria
41 and 42	6.2 11.4	Check the content of the alternative page.	The content of the alternative page is equivalent to the content of the dynamic page.

If the content is published in a non-html format, such as Flash or PDF:

More on page...	WAI Checkpoint	Test	Success Criteria
46 and 47	6.2, 11.1 and 11.4	Check the site for non-HTML elements.	If content is published in a non-HTML format, such as Flash or PDF, an accessible, up-to-date HTML format version should be provided. ⁷

⁷ Does not apply to ODS documents

Validation Procedure for Content Providers

More on page...	WAI Checkpoint	Test	Success Criteria
11	6.1	Under CSS, select <u>Disable Styles / All Styles</u>	The document is still readable. Elements such as headings or lists can be recognised as such.
13	3.2	In Firefox Development Toolbar, validate the HTML (Tools > <u>Validate HTML</u>) and the CSS (Tools > <u>Validate CSS</u>) ⁸ .	The HTML and CSS codes pass validation. No errors are found..
14	3.3 and 11.2	In Firefox Developer Toolbar, Under Outline, <u>Outline Deprecated Elements</u> . ⁹	There are no deprecated elements.
17	4.3	Check the code manually	The proper language code is set for the <html> tag.
19	4.1	Check the page manually. If there is content in a language other than the main language of the page, check the HTML code.	Text in a language other than the main language of the page should be enclosed in a tag using the "lang" attribute.
19	14.1	Manual test	Sentences are short. The text is broken down into short paragraphs. Favour the active voice. Do not use jargon, slang or idiomatic phrases. There are links to definitions of technical terms.
24	3.5 and 12.3	In Firefox Development toolbal, select <u>Outline > Block Level Elements</u> ¹⁰ .	The page should have titles and subtitles. All titles should be outlined and the hierarchy between heading tags should be consistent with the hierarchy of the content.
26	3.6		Lists (such as menus or lists of links or documents)

⁸ If your server cannot be accessed by the W3C validator in a test or intranet environment, in Tools, select Display Page Validation and check the icons in the toolbar.

⁹ You can skip this checkpoint is your page is in XHTML strict and you passed checkpoint 3.2

27	3.7		should be coded as lists. If there are nested, they should be coded as nested lists. The <blockquote> and <q> tags are used for quotations only.
28	13.2	In Firefox Development Toolbar, select <u>Information</u> > <u>View Metatags Information</u> .	Relevant metadata should be provided. It is specific and accurately describes the content of the page / site.
29	4.2	In Firefox Developer Toolbar, under Information, select <u>Display Abbreviation</u> .	Each abbreviation or acronym is marked as such, and the “title” attribute describes its meaning, as it would normally appear in running text.
31	13.1	In Firefox Developer Toolbar, under Information, select <u>View Page Information</u> and click on the link tab.	All links marked as “anchors” should make sense when read out of context. There are no two links with similar text, pointing to a different target. Every link text should be unique to its destination.
32	10.1	In Firefox Development Toolbar, under Information, select View Link Information.	Links should not have the attribute ‘target=’_blank’’. If they do, the users should be warned that the link will open in a new window.

If the page content has image(s):

More on page...	WAI Checkpoint	Test	Success Criteria
5	1.1	Under Images, select <u>Outline images without “alt” attribute</u> Under Images, select <u>Replace images with alt attributes</u>	No image exists without an “alt” attribute. The information conveyed through the use of images is replaced by descriptive “alt” attribute. No information is missing. There are no “alt” attribute with meaningless information.

¹⁰ Make sure the option “Show element name when outlining” is checked (in the Outline menu)

If the page has table(s):

More on page...	WAI Checkpoint	Test	Success Criteria
20	5.1 (1) 5.2 and 5.4	If the table has one logical level: Under Information, select <u>Display table information</u> If the table has more than one logical level: Check the code of the page manually	Data tables have columns and / or rows titles (tagged as <th> instead of <td>) and a summary attribute, describing the structure of the table. Headers and cells are associated using the “id” and “header” attributes.

If content is published in a non-html format (such as Flash or PDF)

More on page...	WAI Checkpoint	Test	Success Criteria
35	11.3	Manual	Links to downloadable files provide users with details about the files (format, size or length, language, etc.).
46	11.1	Check the page.	If content is published in a non-HTML format, such as Flash or PDF, an accessible, up-to-date HTML format version should be provided. ¹¹

¹¹ Does not apply to ODS documents