## Software Accessibility

In many countries, it is now a legal requirement that all software offered for sale be accessible to people with special needs.

For example, under the terms of the Americans with Disabilities Act (ADA), anyone wishing to sell software to the US government or its agencies must demonstrate that every possible effort has been made to accommodate people with certain special needs.

The EU is introducing similar legislation, and a number of countries already have laws in place.

Most such legislation covers both software applications and web-pages developed for profit.

The W3C maintains a [list](http://web.archive.org/web/20100329062659/http%3A/www.w3.org/WAI/Policy/) of legislation affecting web-design in various countries.

Among the special needs that should be considered are:

* Visual impairment/blindness
* Hearing impairment/Deafness
* Physical disabilities
* Speech impairment
* Cognitive & Neurological Disabilities

## Web Accessibility Initiative

The W3C Web Accessibility Initiative (WAI) has issued a set of guidelines to help developers create accessible web-pages.

These are known as the Web Content Accessibility Guidelines (WCAG).

Version is 1.0 in use. Version 2.0 is currently available as a recommendation.

Version 1.0 comprises 14 guidelines.

Each guideline is accompanied by a number of *checkpoints* which are given priorities from 1 to 3:

* If Priority 1 checkpoints are not satisfied, the page will be *impossible* for some groups of users to access.
* If Priority 1 checkpoints are satisfied, but Priority 2 checkpoints are not, the page will be *difficult* for some groups of users to access.
* If Priority 1 and 2 checkpoints are satisfied, but Priority 3 checkpoints are not, the page will be *somewhat difficult* for some groups of users to access.

Web-pages can be rated for accessibility in terms of the checkpoint priority-levels they satisfy:

* Conformance level 'A' - satisfies all Priority 1 checkpoints.
* Conformance level 'AA' - satisfies all Priority 1 & 2 checkpoints.
* Conformance level 'AAA' - satisfies all Priority 1, 2 & 3 checkpoints.

The 14 guidelines are summarised below, along with some of the Priority 1 checkpoints.

The full text of the guidelines, including all Priority 1, 2 and 3 checkpoints, can be found at [http://www.w3.org/TR/WCAG10/](http://web.archive.org/web/20100329062659/http%3A/www.w3.org/TR/WCAG10/).

## Web Content Accessibility Guidelines

**(1) Provide equivalent alternatives to auditory and visual content.**

* For example, use the ALT and/or LONGDESC tags to identify the purpose and/or describe the content of all non-text elements including:

|  |
| --- |
| images (including images of text and images used as bullets, buttons, etc.) |
| image map regions | charts and diagrams |
| audio samples | video clips |
| animations | frames |
| applets and programmatic objects |

For example:

* Provide ALT text for functional images, but not for decorative images.
* Be concise, and avoid stating the obvious.
	+ For example, don't begin with This is a picture of....
* As far as possible, convey in words what the image conveys visually.
	+ For example: The laboratory can hold 25 students and is equipped with iMac computers rather than just The laboratory.
	+ Where a long description is necessary, use LONGDESC and a separate file.
* If images are placed within text, ensure the ALT text makes sense when read in context.

**(2) Don't rely on colour alone.**

Information conveyed through colour alone may be missed by:

* people who are colour-blind
* users with monochrome or non-visual displays

Therefore:

* Only use colour for aesthetic effect or to reinforce information presented by other means.
* Ensure that foreground and background colour combinations provide good contrast when viewed by someone with colour-blindness or when viewed on a black and white screen.

For example:

* Limit the number of colours to a maximum of six (plus black & white).
* Choose colours that offer good contrast.

There are guidelines showing the *intrinsic brightness* of different colours.
* Ensure that colour is only used to provide redundancy by testing the web-page in black-and-white, e.g.:
	+ Take a black-and-white screen-shot of the page - if there is good contrast in the black-and-white image, the colour-contrast is probably good.
	+ Set all colours using a style-sheet, and test with a style-sheet that simulates a monchrome display.

**(3) Use markup and style sheets.**

For example:

* Use CSS instead of the HTML FONT element to control font styles.
* Don't use tables or the PRE tag for layout, or header tags to change the font size.

Doing so makes it difficult for users with specialized software to understand the organization of the page or to navigate through it.
* Use MathML to mark up mathematical equations
* Use relative units (such as *ems* or percentage lengths) rather than absolute units (such as *pts* or *cms*) in markup language attribute values and style sheet property values.

**(4) Clarify natural language usage.**

For example:

* Specify the language of the document (using the lang attribute in the HTML tag) so that automatic content-retrieval systems can locate documents in the required language.
* Identify sections (e.g., quotes, captions) in different languages by using the lang attribute, so that speech-synthesisers, etc., can adjust accordingly.

**(5) Create tables that transform gracefully**

Some user agents allow users to navigate among table cells and access header and other table cell information. Therefore:

* Use tables only for data, not for layout.
* Mark-up tables using th, header, etc.

**(6) Ensure that pages featuring new technologies transform gracefully**

For example:

* Organize documents so they may be read without style sheets.
* 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.

**(7) Ensure user control of time-sensitive content changes**

For example:

* Text, etc., that moves, blinks, scrolls, or otherwise changes over time may be ignored by screen-readers, so either avoid using such elements or provide alternatives.
* Blinking or other rapid changes may trigger fits in those prone to them (e.g., people with epilepsy) so these should be avoided.

**(8) Ensure direct accessibility of embedded user interfaces**

For example:

* If an embedded object has its own interface, check to see if it is accessible and if not, provide an alternative, accessible interface.

**(9) Design for device-independence**

For example:

* Use features that enable activation of page elements via a variety of input devices, such as mouse, keyboard, voice, head wand, etc.
* Provide keyboard shortcuts to important links, form controls, and groups of form controls.

**(10) Use interim solutions**

For example:

* Do not cause pop-ups or other windows to appear, and do not change the current window without informing the user.

**(11) Use W3C technologies and guidelines**

For example:

* Avoid non-W3C formats that require plug-ins or stand-alone applications.

Many of these formats cannot be viewed or navigated with standard user-agents (including assistive technologies).
* If you cannot create an accessible page, provide a link to another page that:
	+ uses W3C technologies
	+ has equivalent information (or functionality)
	+ is updated as often as the original, inaccessible page.
	+ is accessible

**(12) Provide context and orientation information**

For example:

* Title each frame to facilitate frame identification and navigation.
* Use OPTGROUP to group OPTION elements inside a SELECT
* Group form controls with FIELDSET and LEGEND
* Use nested lists where appropriate
* Use headings to structure documents, etc

**(13) Provide clear navigation mechanisms**

For example:

* Clearly identify the target of each link
* Provide metadata to add semantic information to pages and sites
* Provide information about the general layout of a site (e.g., a site map or table of contents)
* Place distinguishing information at the beginning of headings, paragraphs, lists, etc. ("front-loading")

**(14) Ensure that documents are clear and simple**

For example:

* Use consistent page layout.
* Use concise, straightforward language - this benefits users with cognitive disabilities, those whose first language is not that used on the web-site, and blind users accessing the page through speech.