Web Development

Lecture 19 – Accessibility

Web Design in a Nutshell 3ed Jennifer Niederst Robbins
Pub by O'Reilly ISBN 0 596 00987 9

Types of Disabilities

Vision impairment

- Fully blind
- Partially blind
- Colour blind

Mobility impairment

- No use of hands
- Partial use of hands

Auditory impairment

- Fully deaf
- Partially deaf

Cognitive impairment

- Memory loss
- Reading difficulty
- Mathematical reasoning

It is a **legal requirement** for organisations to ensure that their premises are accessible to wheelchair users

They must also ensure that their websites are accessible to people with disabilities.

Disability Discrimination Act (1995)

Equality Act (2010)

Accessibility

There are a number of sets of guidelines which are designed to make websites more accessible to people with visual or hearing impairments.

Web Content Accessiblity Guidelines (W3C)
http://www.w3.org/WAI/intro/wcag

Section 508 (US Government)
http://www.section508.gov/index.cfm

BS 8878: 2010 (BSI & Equality and Human Rights Commission)
http://www.equalityhumanrights.com/footer/accessibility-statement/general-web-accessibility-guidance/

Most countries use a variation on the W3C guidelines.

They are also good rules to follow for general web design.

W3C WCAG

Version 1 published in 1999. Web design has changed a lot since then.

Version 2 became a WCAG recommendation on 11/12/08.

WCAG 2 is now approved as an ISO standard: ISO/IEC 40500:2012.

There are four principles, and each principle is broken down into guidelines.

Each guideline has success criteria specified for it, and also a set of techniques are given which can be used by designers to ensure that the criteria is met.

- Level A – **Must** conform to specification
- Level AA – **Should** conform to specification
- Level AAA – **May** conform to specification

WCAG 2.0 Guidelines

1 Perceivable

- 1.1 Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.
- 1.2 Provide alternatives for time-based media.
- 1.3 Create content that can be presented in different ways (for example simpler layout) without losing information or structure.
- 1.4 Make it easier for users to see and hear content including separating foreground from background.

2 Operable

- 2.1 Make all functionality available from a keyboard.
- 2.2 Provide users enough time to read and use content.
- 2.3 Do not design content in a way that is known to cause seizures.
- 2.4 Provide ways to help users navigate, find content, and determine where they are.

3 Understandable

- 3.1 Make text content readable and understandable.
- 3.2 Make Web pages appear and operate in predictable ways.
- 3.3 Help users avoid and correct mistakes.

4 Robust

- 4.1 Maximize compatibility with current and future user agents, including assistive technologies.

Guideline 1.1 Text Alternatives:

Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.

1.1.1 Non-text Content: All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below. (Level A)

Controls, Input: If non-text content is a control or accepts user input, then it has a name that describes its purpose. (Refer to Guideline 4.1 for additional requirements for controls and content that accepts user input.)

Time-Based Media: If non-text content is time-based media, then text alternatives at least provide descriptive identification of the non-text content. (Refer to Guideline 1.2 for additional requirements for media.)

Test: If non-text content is a test or exercise that would be invalid if presented in text, then text alternatives at least provide descriptive identification of the non-text content.

Sensory: If non-text content is primarily intended to create a specific sensory experience, then text alternatives at least provide descriptive identification of the non-text content.

CAPTCHA: If the purpose of non-text content is to confirm that content is being accessed by a person rather than a computer, then text alternatives that identify and describe the purpose of the non-text content are provided, and alternative forms of CAPTCHA using output modes for different types of sensory perception are provided to accommodate different disabilities.

Decoration, Formatting, Invisible: If non-text content is pure decoration, is used only for visual formatting, or is not presented to users, then it is implemented in a way that it can be ignored by assistive technology.

Guideline 2.3 Seizures:

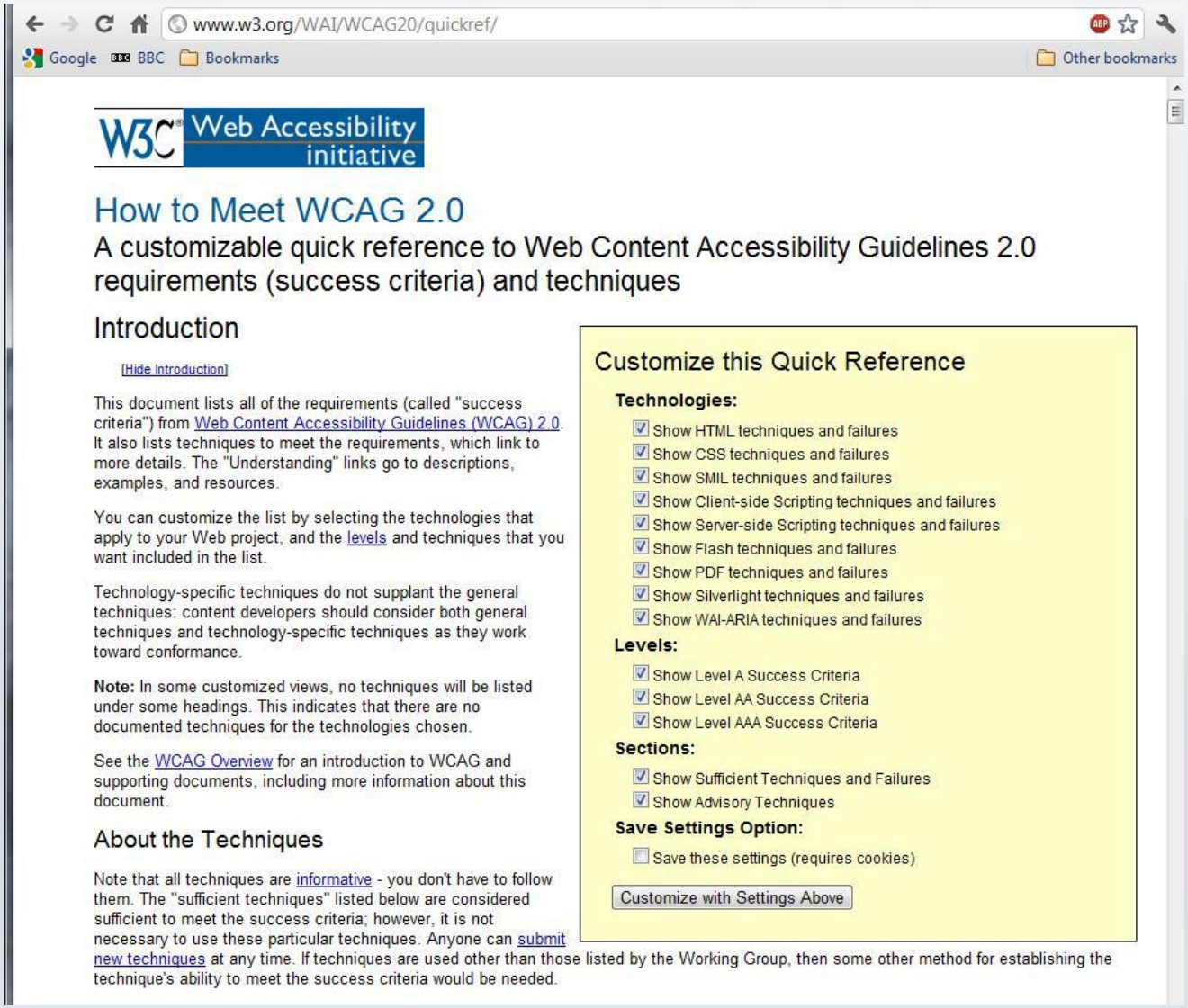
Do not design content in a way that is known to cause seizures.

2.3.1 Three Flashes or Below Threshold: Web pages do not contain anything that flashes more than three times in any one second period, or the flash is below the general flash and red flash thresholds. (Level A)

Note: Since any content that does not meet this success criterion can interfere with a user's ability to use the whole page, all content on the Web page (whether it is used to meet other success criteria or not) must meet this success criterion. See Conformance Requirement 5: Non-Interference.

2.3.2 Three Flashes: Web pages do not contain anything that flashes more than three times in any one second period. (Level AAA)

http://www.w3.org/WAI/WCAG20/quickref/



Quick Reference Checklist

Section 508 (US Government Rules)

A set of rules (a) to (p) that are similar to the W3C guidelines.

(a) A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content).

(b) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.

(c) Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.

(d) Documents shall be organized so they are readable without requiring an associated style sheet.

(e) Redundant text links shall be provided for each active region of a server-side image map.

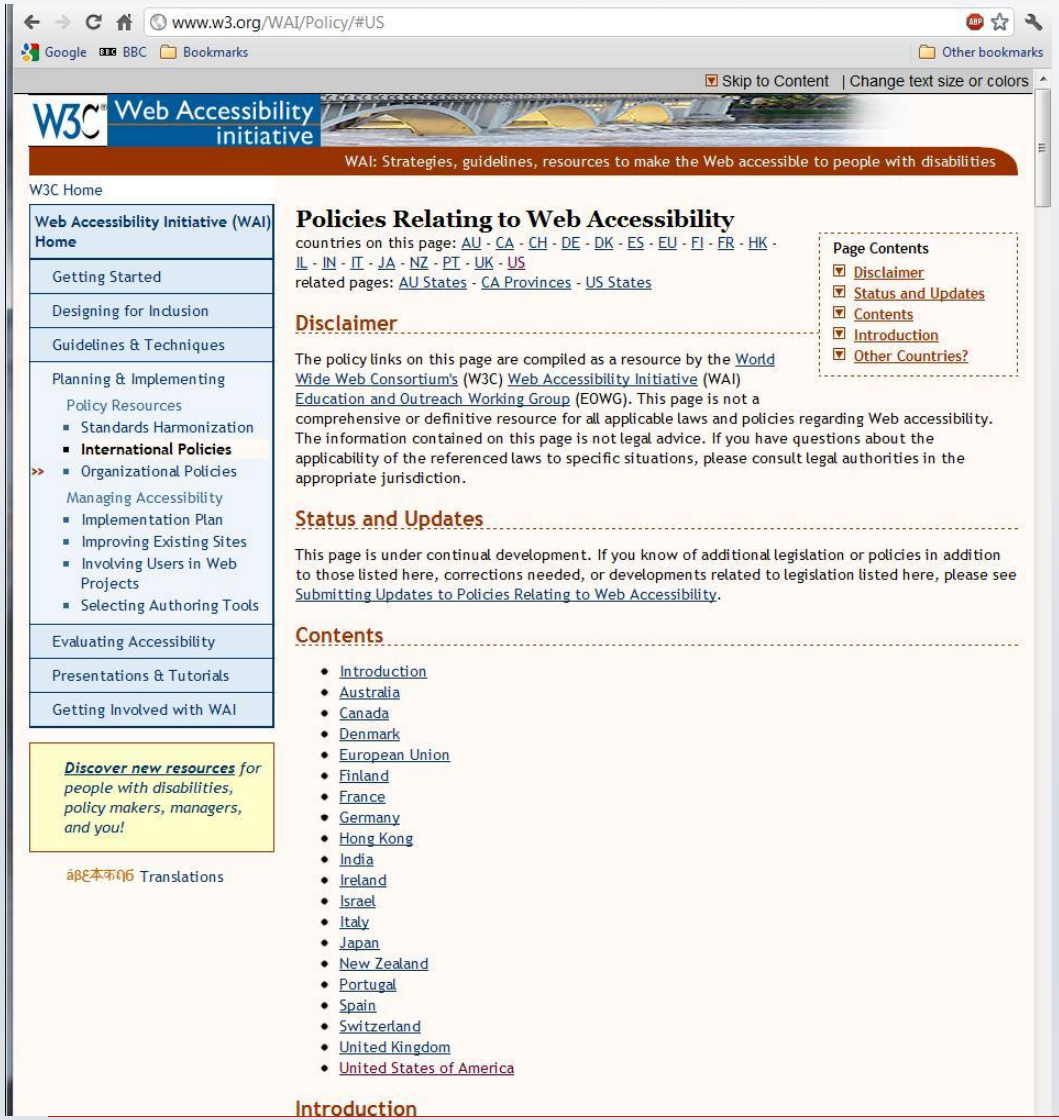
BS 8878:2010 (British Standards Institute)

This is a more general document - targeted more at the manager who is employing someone else to design the website.

Contents include:

1. Terms and definitions
2. General principals
3. How disabled people use websites
4. Defining the accessibility policy for the website
5. Web technologies
6. Accessibility testing and maintenance
7. Contracting web design and accessibility auditing services.

http://www.w3.org/WAI/Policy/



Links to Web Accessibility resources for other countries

Example of good practice

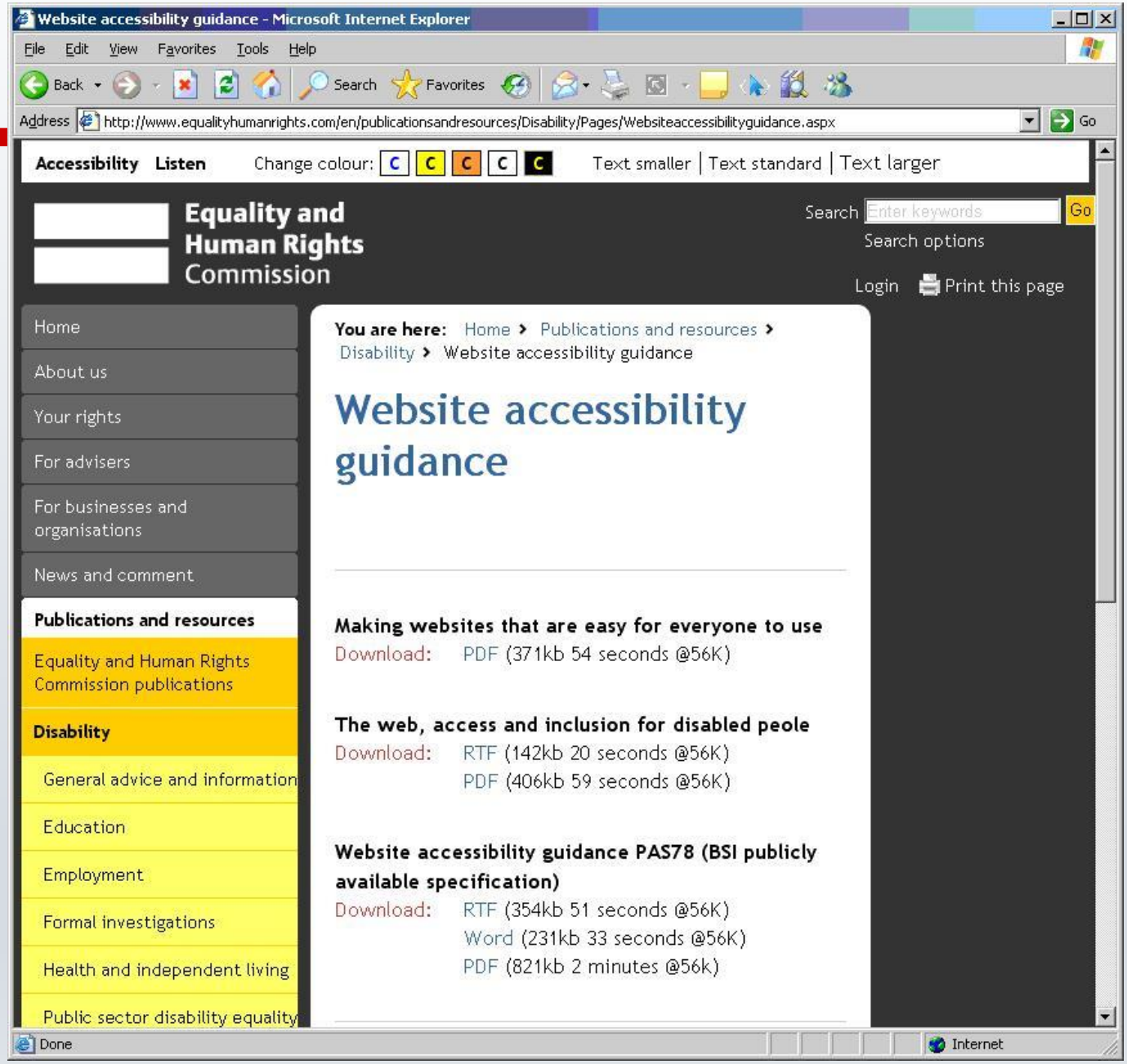
Instructions

Reader

Colour

Text Size

Breadcrumb Trail

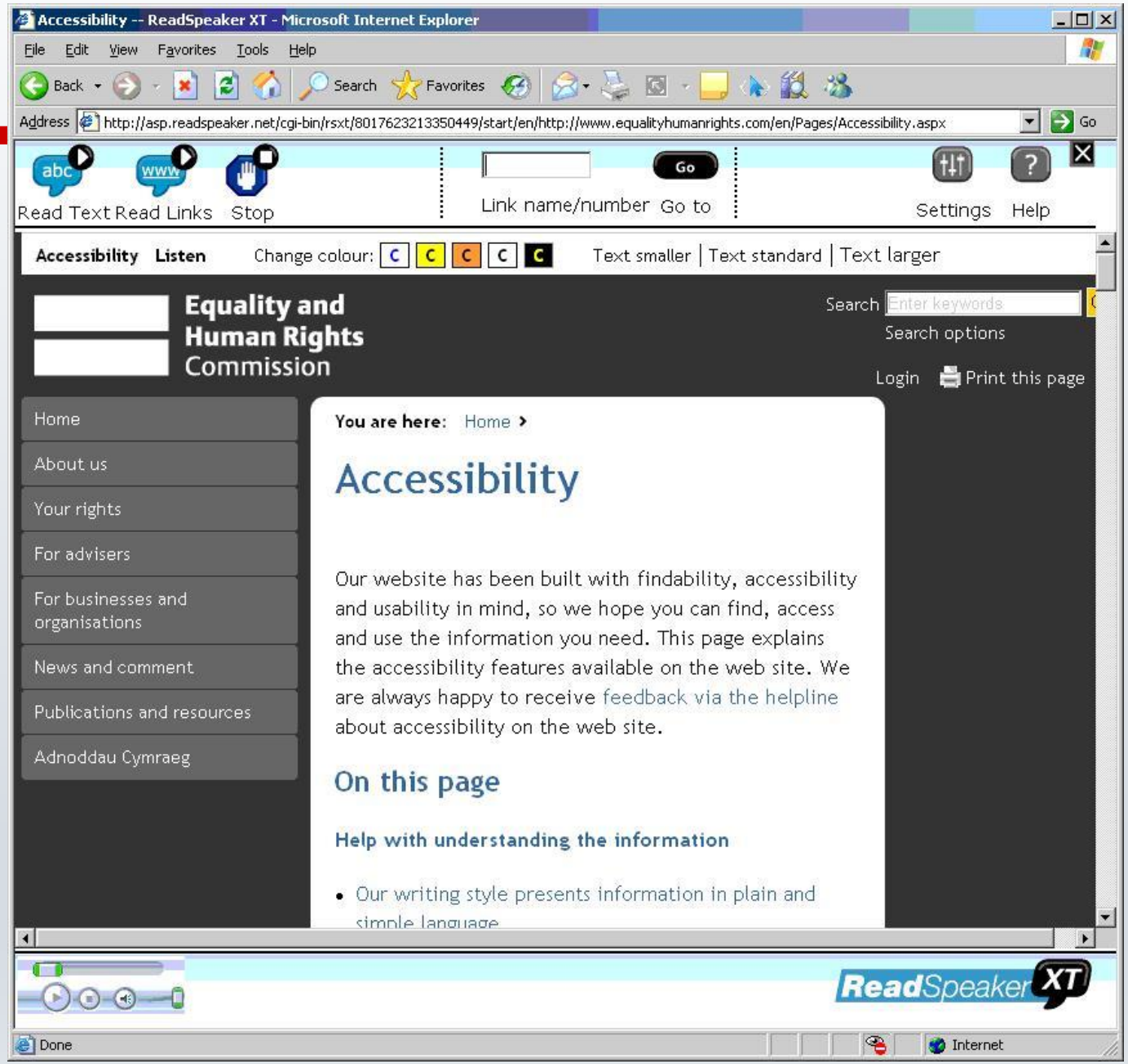


Reader

Read Links

Go to Link

Settings



Official guidelines often tell you what you should do, but not necessarily how you should do it.

The W3C guidelines are the exception - they contain good advice to the web designer about techniques - with examples.

There follows a general set of rules which you try to keep in mind when designing websites.

1. Use Meaningful Mark-up

Use HTML tags to describe meaning, not appearance.

<h1>Introduction</h1>

indicates that the text is a heading

You can associate any fontsize, color etc with <h1> in the stylesheet.

The elements or attributes that have been deprecated are the ones that used to be used for controlling appearance

<bold> <underline> <img align="center">

Example of a tag in use

```
<p>
Call me on <span class="phone">12345</span> now.
</p>
```

I can then set up a style sheet which will give a uniform appearance to all phone numbers – or all numbers that I have marked up as phone numbers.

2. Provide Alternatives

Use of the alt attribute.

```
<img src="Cat.gif" alt="Picture of a cat">
```

will:

- cause a tooltip to pop up when the mouse is over the image
- cause the text to appear if the image cannot load
- enable reader software to describe the image to a blind person

Also

- Provide transcripts for audio files.
- Provide captions for video files.

This will not only benefit people with sight and hearing impairments, but also people who are still using dial-up.

3. Remember that Order Counts

A logical order within the page makes it a lot easier to understand.

Screen readers tend to read things in the order of the source code.

Make sure that it applies to not just the entire page, but to things within the page – groups of links, form fields.

What does your page look like with the styles turned off?

4. Test Your Javascript

Some people often have Javascript switched off.

Test your page with Javascript support turned on and turned off.

Can the user still use your website?

If your menu or navigation system relies on Javascript, have you provided an alternative?

5. Make it Easy to Move Around the Page

Make use of anchor points within a page

```
<a id="para3">Course List</a>
<h1 id="para5">Student List</h1>
```

and then you can provide links to them from elsewhere in the page:

```
<a href="#para3">Go to Course List</a>

<a href="#para5">Go to Student List</a>
```

6. Make It Clear Which Page Element Has Focus

Use the focus state for links.

This enables the link to change colour when it is selected or has focus. Put the following in the stylesheet.

```
a:link
{
color: #000000;
background-color: #ffffff;
}

a:hover
{
color: #ffffff;
background-color: #000000;
}
```

To provide the same feedback for people who are using the keyboard / tab key:

```
a:focus
{
color: #ffffff;
background-color: #000000;
}
```

Internet Explorer doesn't recognise focus – use the following instead.

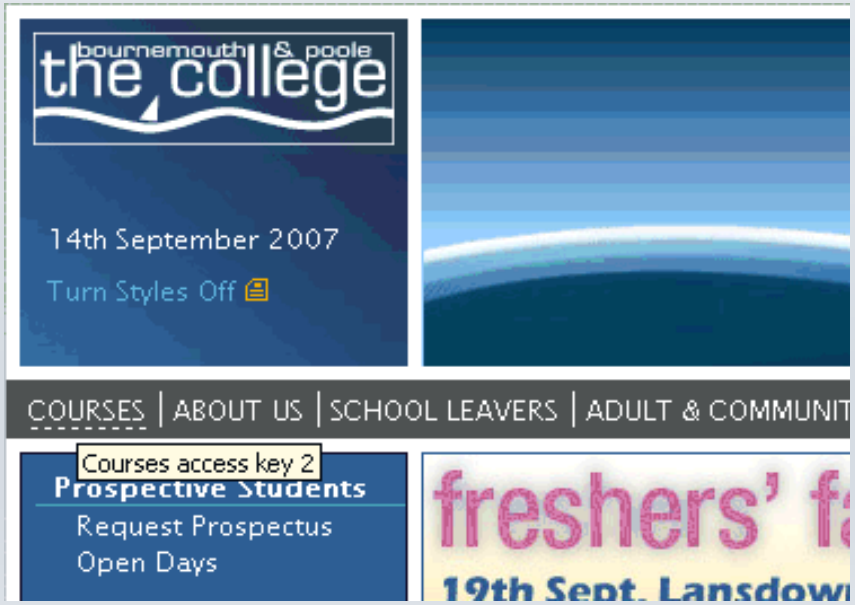
```
a:active
{
color: #ffffff;
background-color: #000000;
}
```

7. Provide Keyboard Alternatives to Mouse Use

Use the accesskey attribute

```
<input type="text" name="address" accesskey="2">
```

will enable you to select that text box by typing Alt-2. You can also use it on other elements.



tabindex

Another way of changing focus without using a mouse is to use the tab key.

If you want to change the default tab order, you can use the tabindex attribute.

```
User ID: <input type="text" name="login" tabindex="3"/>
Password: <input type="text" name="Pass" tabindex="2" />
Postcode: <input type="text" name="Postcode" tabindex="1"/>
```

8. Handle Colours Carefully

Declare your colours in pairs (foreground and background).

Do so only in CSS.

Don't declare the foreground colour in CSS and the background colour in the HTML.

The stylesheet may be switched off, or not supported, or overridden by the user.

Provide alternative stylesheets for the same page to help people with colour blindness.

Colour contrast analysers are available at:
http://juicystudio.com/services/colourcontrast.php

9. Use Well-Formed XHTML

- All lower case element and attribute names.
- Quoted Attribute Values (even single words and numbers).
- Always terminate elements.
  - Even empty elements. (
 or
</br>)
- Explicit attribute values (checked="checked").
- Make sure that all elements are nested correctly.
- Use the id attribute in preference the name attribute.
- Place CSS and Scripts in their own files.

Testing for Accessibility

Web Developer Toolbar for Firefox

- Provides a set of tools which include some that can be used to test for accessibility.

Accessibility Toolbar for Internet Explorer

- See WAT-C website

Opera Browser

- Includes quick access to various browser modes. Also includes voice recognition and speech capabilities built into the browser.

WAT-C Online tools

- Web Accessibility Tool Consortium - set of free tools that can be used for testing websites (http://www.wat-c.org/)

http://www.w3.org/WAI/ER/tools