D-WAB4 Unified Web Evaluation Methodology (UWEM 1.2 Tests)

<table>
<thead>
<tr>
<th>Contractual Date of Delivery to the EC:</th>
<th>30 September 2007 + 45 days</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual Date of Delivery to the EC:</td>
<td>03 December 2007</td>
</tr>
<tr>
<td>Revision Date:</td>
<td>04 September 2007</td>
</tr>
<tr>
<td>Editors:</td>
<td>Eric Velleman (Accessibility Foundation), Johannes Koch, Carlos A Velasco (Fraunhofer Institute for Applied Information Technology FIT), Christophe Strobbe (K.U. Leuven), Annika Nietzio (Forschungsinstitut Technologie und Behinderung), Mikael Snaprud (Agder University College)</td>
</tr>
<tr>
<td>Contributors:</td>
<td>See UWEM version 1.2 Core document for contributors list</td>
</tr>
<tr>
<td>Workpackage:</td>
<td>WAB1b</td>
</tr>
<tr>
<td>Security:</td>
<td>Public</td>
</tr>
<tr>
<td>Nature:</td>
<td>Report</td>
</tr>
<tr>
<td>Version:</td>
<td>E</td>
</tr>
<tr>
<td>Total number of pages:</td>
<td>100</td>
</tr>
</tbody>
</table>


Please see Appendix A for the document license.
## DOCUMENT HISTORY

<table>
<thead>
<tr>
<th>Version</th>
<th>Version date</th>
<th>Responsible</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2007-08-30</td>
<td>Eric Velleman</td>
<td>TEST document to be combined with UWEM CORE 1.2</td>
</tr>
<tr>
<td>B</td>
<td>2007-08-31</td>
<td>Colin Meerveld</td>
<td>Updates to section 5 in response to comments</td>
</tr>
<tr>
<td>C</td>
<td>2007-09-04</td>
<td>Carlos A Velasco</td>
<td>Overall copy-editing.</td>
</tr>
<tr>
<td>D</td>
<td>2007-11-24</td>
<td>Christophe Strobbe, Johannes Koch, Annika Nietzio, Carlos A Velasco</td>
<td>Integration of issues. Overall editing</td>
</tr>
<tr>
<td>E</td>
<td>2008-03-12</td>
<td>Eric Velleman, Christophe Strobbe</td>
<td>Editorial changes after final review of documents by EC.</td>
</tr>
</tbody>
</table>
# Table of Contents

1 Executive summary ........................................................................................................... 9
2 Tests for conformance evaluation ...................................................................................... 10
  2.1 Introduction .................................................................................................................. 10
  2.2 Guideline 1 .................................................................................................................. 13
    2.2.1 Checkpoint 1.1 ..................................................................................................... 13
      2.2.1.1 (X)HTML tests .............................................................................................. 13
        2.2.1.1.1 Test 1.1_HTML_01 .................................................................................. 13
        2.2.1.1.2 Test 1.1_HTML_02 .................................................................................. 14
        2.2.1.1.3 Test 1.1_HTML_03 .................................................................................. 14
        2.2.1.1.4 Test 1.1_HTML_04 .................................................................................. 15
        2.2.1.1.5 Test 1.1_HTML_05 .................................................................................. 15
        2.2.1.1.6 Test 1.1_HTML_06 .................................................................................. 15
        2.2.1.1.7 Test 1.1_HTML_07 .................................................................................. 16
        2.2.1.1.8 Test 1.1_HTML_08 .................................................................................. 16
        2.2.1.1.9 Test 1.1_HTML_09 .................................................................................. 16
        2.2.1.1.10 Test 1.1_HTML_10 ............................................................................... 17
    2.2.2 Checkpoint 1.2 ...................................................................................................... 17
      2.2.2.1 (X)HTML tests .............................................................................................. 18
        2.2.2.1.1 Test 1.2_HTML_01 .................................................................................. 18
    2.2.3 Checkpoint 1.3 ...................................................................................................... 19
      2.2.3.1 Tests for external objects .............................................................................. 19
        2.2.3.1.1 Test 1.3_external_01 ............................................................................. 19
        2.2.3.1.2 Test 1.3_external_02 ............................................................................. 19
    2.2.4 Checkpoint 1.4 ...................................................................................................... 20
      2.2.4.1 Tests for external objects .............................................................................. 20
        2.2.4.1.1 Test 1.4_external_01 ............................................................................. 20
  2.3 Guideline 2 .................................................................................................................. 20
    2.3.1 Checkpoint 2.1 ...................................................................................................... 20
      2.3.1.1 (X)HTML tests .............................................................................................. 21
        2.3.1.1.1 Test 2.1_HTML_01 .................................................................................. 21
        2.3.1.1.2 Test 2.1_HTML_02 .................................................................................. 21
        2.3.1.1.3 Test 2.1_HTML_03 .................................................................................. 22
      2.3.1.2 CSS tests ....................................................................................................... 22
        2.3.1.2.1 Test 2.1_CSS_01 ................................................................................... 22
    2.3.2 Checkpoint 2.2 ...................................................................................................... 22
      2.3.2.1 (X)HTML tests .............................................................................................. 23
        2.3.2.1.1 Test 2.2_HTML_01 .................................................................................. 23
      2.3.2.2 CSS tests ....................................................................................................... 23
        2.3.2.2.1 Test 2.2_CSS_01 ................................................................................... 23
  2.4 Guideline 3 .................................................................................................................. 24
    2.4.1 Checkpoint 3.1 ...................................................................................................... 24
2.4.1.1(X)HTML tests.............................................................................................................24
  2.4.1.1.1Test 3.1_HTML_01..............................................................................................................24
  2.4.1.1.2Test 3.1_HTML_02..............................................................................................................24
  2.4.1.1.3Test 3.1_HTML_03..............................................................................................................25
2.4.2Checkpoint 3.2.......................................................................................................................25
  2.4.2.1(X)HTML tests.....................................................................................................................26
  2.4.2.1.1Test 3.2_HTML_01..................................................................................................................26
  2.4.2.1.2Test 3.2_HTML_02..................................................................................................................26
  2.4.2.2CSS tests................................................................................................................................26
  2.4.2.2.1Test 3.2_CSS_01....................................................................................................................26
2.4.3Checkpoint 3.3.........................................................................................................................27
  2.4.3.1(X)HTML tests.....................................................................................................................27
  2.4.3.1.1Test 3.3_HTML_01..................................................................................................................27
  2.4.3.1.2Test 3.3_HTML_02..................................................................................................................28
2.4.4Checkpoint 3.4.........................................................................................................................28
  2.4.4.1(X)HTML tests.....................................................................................................................29
  2.4.4.1.1Test 3.4_HTML_01..................................................................................................................29
  2.4.4.1.2Test 3.4_HTML_02..................................................................................................................29
  2.4.4.1.3Test 3.4_HTML_03..................................................................................................................30
  2.4.4.2CSS tests................................................................................................................................30
  2.4.4.2.1Test 3.4_CSS_01....................................................................................................................30
2.4.5Checkpoint 3.5.........................................................................................................................31
  2.4.5.1(X)HTML tests.....................................................................................................................32
  2.4.5.1.1Test 3.5_HTML_01..................................................................................................................32
  2.4.5.1.3Test 3.5_HTML_03..................................................................................................................32
  2.4.5.1.4Test 3.5_HTML_04..................................................................................................................33
  2.4.5.1.5Test 3.5_HTML_05..................................................................................................................33
2.4.6Checkpoint 3.6.........................................................................................................................33
  2.4.6.1(X)HTML tests.....................................................................................................................34
  2.4.6.1.1Test 3.6_HTML_01..................................................................................................................34
  2.4.6.1.2Test 3.6_HTML_02..................................................................................................................34
  2.4.6.1.3Test 3.6_HTML_03..................................................................................................................34
  2.4.6.1.4Test 3.6_HTML_04..................................................................................................................34
  2.4.6.1.5Test 3.6_HTML_05..................................................................................................................35
  2.4.6.1.6Test 3.6_HTML_06..................................................................................................................36
  2.4.6.1.7Test 3.6_HTML_07..................................................................................................................36
2.4.7Checkpoint 3.7.........................................................................................................................37
  2.4.7.1(X)HTML tests.....................................................................................................................37
  2.4.7.1.1Test 3.7_HTML_01..................................................................................................................37
  2.4.7.1.2Test 3.7_HTML_02..................................................................................................................37
  2.4.7.1.3Test 3.7_HTML_03..................................................................................................................38
  2.4.7.1.4Test 3.7_HTML_04..................................................................................................................38
2.5Guideline 4.................................................................................................................................38
  2.5.1Checkpoint 4.1.........................................................................................................................39
    2.5.1.1(X)HTML tests......................................................................................................................39
    2.5.1.1.1Test 4.1_HTML_01..................................................................................................................39
    2.5.1.1.2Test 4.1_HTML_02..................................................................................................................40
2.5.1.1.3 Test 4.1_HTML_03 .................................................. 40
2.5.1.1.4 Test 4.1_HTML_04 .................................................. 41
2.5.1.2 CSS tests ............................................................... 42
2.5.1.2.1 Test 4.1_CSS_01 .................................................. 42

2.6 Guideline 5 ................................................................. 43
2.6.1 Checkpoint 5.1 .......................................................... 43
2.6.1.1(X) HTML tests ..................................................... 43
2.6.1.1.1 Test 5.1_HTML_01 ................................................. 43
2.6.1.1.2 Test 5.1_HTML_02 ................................................. 43
2.6.1.2 CSS Tests .......................................................... 44
2.6.1.2.1 Test 5.1_CSS_01 .................................................. 44

2.6.2 Checkpoint 5.2 .......................................................... 44
2.6.2.1(X) HTML tests ..................................................... 44
2.6.2.1.1 Test 5.2_HTML_01 ................................................. 44
2.6.2.1.2 Test 5.2_HTML_02 ................................................. 45
2.6.2.1.3 Test 5.2_HTML_03 ................................................. 46

2.6.3 Checkpoint 5.3 .......................................................... 46
2.6.3.1(X) HTML tests ..................................................... 46
2.6.3.1.1 Test 5.3_HTML_01 ................................................. 46

2.6.4 Checkpoint 5.4 .......................................................... 47
2.6.4.1(X) HTML tests ..................................................... 47
2.6.4.1.1 Test 5.4_HTML_01 ................................................. 47
2.6.4.1.2 Test 5.4_HTML_02 ................................................. 47
2.6.4.1.3 Test 5.4_HTML_03 ................................................. 48
2.6.4.1.4 Test 5.4_HTML_04 ................................................. 48
2.6.4.1.5 Test 5.4_HTML_05 ................................................. 48

2.7 Guideline 6 ................................................................. 49
2.7.1 Checkpoint 6.1 .......................................................... 49
2.7.1.1(X) HTML tests ..................................................... 49
2.7.1.1.1 Test 6.1_HTML_01 ................................................. 49

2.7.2 Checkpoint 6.2 .......................................................... 50
2.7.2.1(X) HTML tests ..................................................... 50
2.7.2.1.1 Test 6.2_HTML_01 ................................................. 50
2.7.2.1.2 Test 6.2_HTML_02 ................................................. 50
2.7.2.1.3 Test 6.2_HTML_03 ................................................. 51

2.7.3 Checkpoint 6.3 .......................................................... 52
2.7.3.1(X) HTML tests ..................................................... 52
2.7.3.1.1 Test 6.3_HTML_01 ................................................. 52
2.7.3.1.2 Test 6.3_HTML_02 ................................................. 53

2.7.4 Checkpoint 6.4 .......................................................... 53
2.7.4.1(X) HTML tests ..................................................... 54
2.7.4.1.1 Test 6.4_HTML_01 ................................................. 54
2.7.4.1.2 Test 6.4_HTML_02 ................................................. 55

2.7.4.2 Tests for external objects ....................................... 55
2.7.4.2.16.4_external_01 .................................................. 55

2.7.5 Checkpoint 6.5 .......................................................... 55
2.7.5.1(X) HTML tests ..................................................... 56
2.7.5.1.1 Test 6.5_HTML_01 .............................................. 56
2.7.5.1.3 Test 6.5_HTML_03 .............................................. 56
2.8.1.1 Guideline 7 .......................................................... 57

<table>
<thead>
<tr>
<th>Checkpoint 7.1</th>
<th>2.8.1.1</th>
<th>(X)HTML tests</th>
<th>2.8.1.1.1 Test 7.1_HTML_01</th>
<th>58</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.8.1.1.2 Test 7.1_HTML_02</td>
<td>58</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.8.1.1.3 Test 7.1_HTML_03</td>
<td>59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.8.1.2</td>
<td>CSS tests</td>
<td>2.8.1.2.1 Test 7.1_CSS_01</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>2.8.1.3</td>
<td>Test for external objects</td>
<td>2.8.1.3.1 Test 7.1_external_01</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.8.1.3.2 Test 7.1_external_02</td>
<td>60</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Checkpoint 7.2</th>
<th>2.8.2.1</th>
<th>(X)HTML tests</th>
<th>2.8.2.1.1 Test 7.2_HTML_01</th>
<th>61</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.8.2.1.2 Test 7.2_HTML_02</td>
<td>61</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.8.2.1.3 Test 7.2_HTML_03</td>
<td>62</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.8.2.2</td>
<td>CSS tests</td>
<td>2.8.2.2.1 Test 7.2_CSS_01</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.8.2.2.2 Test 7.2_CSS_02</td>
<td>63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.8.2.3</td>
<td>Tests for external objects</td>
<td>2.8.2.3.1 Test 7.2_external_01</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.8.2.3.2 Test 7.2_external_02</td>
<td>64</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Checkpoint 7.3</th>
<th>2.8.3.1</th>
<th>(X)HTML tests</th>
<th>2.8.3.1.1 Test 7.3_HTML_01</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.8.3.1.2 Test 7.3_HTML_02</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.8.3.2</td>
<td>CSS tests</td>
<td>2.8.3.2.1 Test 7.3_CSS_01</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>2.8.3.3</td>
<td>Tests for external objects</td>
<td>2.8.3.3.1 Test 7.3_external_01</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.8.3.3.2 Test 7.3_external_02</td>
<td>66</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Checkpoint 7.4</th>
<th>2.8.4.1</th>
<th>(X)HTML tests</th>
<th>2.8.4.1.1 Test 7.4_HTML_01</th>
<th>67</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.8.4.1.2 Test 7.4_HTML_02</td>
<td>67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.8.4.2</td>
<td>Tests for external objects</td>
<td>2.8.4.2.1 Test 7.4_external_01</td>
<td>68</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Checkpoint 7.5</th>
<th>2.8.5.1</th>
<th>(X)HTML tests</th>
<th>2.8.5.1.1 Test 7.5_HTML_01</th>
<th>68</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.8.5.1.2 Test 7.5_HTML_02</td>
<td>68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.8.5.2</td>
<td>Tests for external objects</td>
<td>2.8.5.2.1 Test 7.5_external_01</td>
<td>69</td>
<td></td>
</tr>
</tbody>
</table>

| Guideline 8 | 2.9.1 | Checkpoint 8.1 | 70 |

Deliverable D-WAB4 (Public)
2.9.1.1(X)HTML tests

2.9.1.1.1Test 8.1_HTML_01

2.9.1.2Tests for external objects

2.9.1.2.1Test 8.1_external_01

2.10Guideline 9

2.10.1Checkpoint 9.1

2.10.1.1(X)HTML tests

2.10.1.1.1Test 9.1_HTML_01

2.10.2Checkpoint 9.2

2.10.2.1Tests for external objects

2.10.2.1.1Test 9.2_external_01

2.10.3Checkpoint 9.3

2.10.3.1(X)HTML tests

2.10.3.1.1Test 9.3_HTML_01

2.11Guideline 10

2.11.1Checkpoint 10.1

2.11.1.1(X)HTML tests

2.11.1.1.1Test 10.1_HTML_01

2.11.1.2Test 10.1_HTML_02

2.11.1.3Test 10.1_HTML_03

2.11.2Checkpoint 10.2

2.11.2.1(X)HTML tests

2.11.2.1.1Test 10.2_HTML_01

2.12Guideline 11

2.12.1Checkpoint 11.1

2.12.1.1(X)HTML tests

2.12.1.1.1Test 11.1_HTML_01

2.12.2Checkpoint 11.2

2.12.2.1(X)HTML tests

2.12.2.1.1Test 11.2_HTML_01

2.12.2.1.2Test 11.2_HTML_02

2.12.3Checkpoint 11.4

2.12.3.1(X)HTML tests

2.12.3.1.1Test 11.4_HTML_01

2.12.3.1.2Test 11.4_HTML_02

2.12.3.1.3Test 11.4_HTML_03

2.13Guideline 12

2.13.1Checkpoint 12.1

2.13.1.1(X)HTML tests

2.13.1.1.1Test 12.1_HTML_01

2.13.1.1.2Test 12.1.HTML_02

2.13.2Checkpoint 12.2

2.13.2.1(X)HTML tests

2.13.2.1.1Test 12.2_HTML_01

2.13.3Checkpoint 12.3

2.13.3.1(X)HTML tests

2.13.3.1.1Test 12.3_HTML_01
1 Executive summary

This document contains the applicable tests of the Unified Web Evaluation Methodology (UWEM) version 1.2. For the sake of conciseness and document usability, UWEM version 1.2 has been split in two documents: the core documentation, and the tests (this document). UWEM is a methodology for evaluating conformance of web sites to Web Content Accessibility Guidelines 1.0 [WCAG10], a recommendation published by the World Wide Web Consortium (W3C) in 1999. For this purpose, UWEM contains:

1. Descriptions of evaluation procedures (see UWEM version 1.2 Core);

2. Recommendations for sampling resources ("web pages") and expressing the scope of a conformance claim (see UWEM version 1.2 Core);

3. Tests for the WCAG checkpoints at priority levels 1 and 2 (checkpoints and levels are defined by WCAG 1.0) (Section 2),

4. Recommendations for reporting evaluation results by means of a conformance claim, a text-based report, a machine-readable report and/or a scorecard report. It also includes recommendations on scoring and aggregating evaluation results (see UWEM version 1.2 Core),

5. Templates and data definitions (in RDF Schema) to support evaluators (see UWEM version 1.2 Core).

2 Tests for conformance evaluation

2.1 Introduction

This section covers the UWEM automatic and expert testing of Priority 1 and Priority 2 checkpoints of WCAG 1.0 [WCAG10]. For this purpose, this section provides tests for expert and/or automatic evaluation.

The structure of the tests in this section is the following:

1. Guideline

Quotation of the corresponding WCAG 1.0 guideline. Pointers to additional clarifications might be added.

• Checkpoint

Quotation of the corresponding WCAG 1.0 checkpoint. Pointers to additional clarifications might be added. For each checkpoint, a set of one or more tests is defined. If no automatic tests for a certain technology are defined, this means that there are no applicable tests for automated testing.

a) (X)HTML-specific tests

Set of tests to be made for conformance claims for (X)HTML resources. Each test consists of:

• Title and ID: short descriptive title (informative) and unique identifier (normative).

• Applicability criteria: elements, attributes and combinations thereof used to determine the applicability of the test. Whenever possible, the criteria are presented as XPath\(^1\) expressions, otherwise a prose description is given. XPath expressions always refer to generated code, including code generated by client-side script.

• Test procedure: description in a tool-independent manner of the test procedure. The procedure may

---

\(^1\) XPath is a language for identifying parts or "nodes" in an XML document (not only elements and attributes, but also comments, processing instructions and text nodes). The XPath version used in this version of UWEM is XPath 1.0 [XPATH10]. The UWEM editors assume that the XPath expressions and functions used in UWEM will generate the same results when interpreted according to [XPATH20] and [XPATH-Functions], respectively.
consist of multiple steps and is written so as to enable possible machine-testing. The test procedure needs to be applied to each individual item (element, attribute, object, etc.) covered by the applicability criteria, unless the test procedure states otherwise.

- **Expected results:** statement defining the fail or pass conditions with regard to one or more steps in the test procedure. The elements or content specified in the accessibility criteria pass the test if the result is not FAIL.

- **Fully automatable:** statement whether the test procedure can be fully automated (yes/no).

b) **CSS-specific tests**

Set of tests to be made for conformance claims for CSS resources. Each test consists of:

- **Title and ID:** short descriptive title (informative) and unique identifier (normative).

- **Applicability criteria:** CSS selectors, properties and combinations thereof used to determine the applicability of the test. This includes CSS code in external files (referenced by link elements or @import rules), style elements (<style type="text/css"...), style attributes, and any such CSS (and references to CSS files) generated by client-side script\(^2\).

- **Test procedure:** description in a tool-independent manner of the test procedure. The procedure may consist of multiple steps and is written so as to enable possible machine-testing. The test procedure needs to be applied to each individual item (CSS selector, property, etcetera) covered by the applicability criteria, unless the test procedure states otherwise.

- **Expected results:** statement defining the fail or pass conditions with regard to one or more steps in the test

---
\(^2\) The CSS tests only apply to style rules that match elements in the document; style rules whose selectors do not match any content in the document must be ignored. A tool implementing the CSS tests first needs to retrieve all the stylesheets associated with the document (including external files, style elements, style attributes and JavaScript-generated styles, as mentioned above). In a second step, the style rules that match elements in the document are retained for inspection.
procedure. The elements or content specified in the accessibility criteria pass the test if the result is not FAIL.

- **Fully automatable:** statement whether the test procedure can be fully automated (yes/no).

### c) Tests for external objects

Set of tests to be made for conformance claims for objects included or embedded in web pages through HTML elements or CSS-generated content. This includes applets, Flash, video and audio. Each test consists of:

- **Title and ID:** short descriptive title (informative) and unique identifier (normative).

- **Applicability criteria:** Elements, attributes and combinations thereof used to determine the applicability of the test.

- **Test procedure:** description in a tool-independent manner of the test procedure. The procedure may consist of multiple steps and is written so as to enable possible machine-testing. The test procedure needs to be applied to each individual item (element, attribute, object, CSS rule, etcetera) covered by the applicability criteria, unless the test procedure states otherwise.

- **Expected results:** statement defining the fail or pass conditions with regard to one or more steps in the test procedure. The elements or content specified in the accessibility criteria pass the test if the result is not FAIL.

- **Fully automatable:** statement whether the test procedure can be fully automated (yes/no).

2. (Optional) Additional clarification issues, such as definition pointers.

This section does not repeat information available in W3C documents. Instead, it provides pointers to the relevant places and extends only information when necessary for the defined tests.

Web content passes a checkpoint if it fails none of the applicable tests for that checkpoint. Web content fails a checkpoint if it fails any of the
applicable tests for that checkpoint. Web content fails an applicable test if any item covered by the applicability criteria causes a FAIL after using the test procedure.

For definitions of terms used in this document, see the glossary section in the UWEM 1.2 core document.

2.2 Guideline 1

"Provide equivalent alternatives to auditory and visual content."

(See http://www.w3.org/TR/WCAG10/#gl-provide-equivalents)

This guideline provides information on how to support with complementary text alternatives auditory and visual content.

2.2.1 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, 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. [Priority 1]

(See http://www.w3.org/TR/WCAG10/#tech-text-equivalent and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-text-equivalent)

2.2.1.1 (X)HTML tests

2.2.1.1.1 Test 1.1_HTML_01

This test is targeted to check that non-text content has a text equivalent.

- Applicability criteria: all non-text elements that support the alt attribute.

```html
//img
//area
//input[@type='image']
//applet
```

- Test procedure:
  Check that the element has an alt attribute.

- Expected results: PASS if true. FAIL if false.
• Fully automatable: yes.

2.2.1.1.2 Test 1.1_HTML_02

This test is targeted to analyse non-text elements with an empty text alternative.

• Applicability criteria: non-text elements with empty text alternative.

```xml
//img[@alt='']
//area[@alt='']
//input[@type='image'][@alt='']
//applet[@alt=''][count(local-name(*)!='param')=0]
//object[count(local-name(*)!='param')=0]
```

• Test procedure:
  1. Check that the image/content is purely decorative.
  2. If #1 is false, check that there is a text alternative adjacent to the non-text content.

• Expected results: PASS if #1 or #2 is true. FAIL if #1 and #2 are false

• Fully automatable: no.

2.2.1.1.3 Test 1.1_HTML_03

This test is targeted to analyse non-text elements with non-empty text alternative.

• Applicability criteria: all non-text elements with non-empty text alternative.

```xml
//img[@alt!='']
//area[@alt!='']
//input[@type='image'][@alt][@alt!='']
//applet[@alt][@alt!='']
//object[count(local-name(*)!='param')>0]
```

• Test procedure:
  Check that the text alternative represents the function of the non-text-element within the context. If there is text content adjacent to the non-text element, the text alternative can consist of this text content combined with the non-text element alt attribute value.

• Expected results: PASS if true. FAIL if false.

• Fully automatable: no.

2.2.1.1.4 Test 1.1_HTML_04

This test is targeted to analyse long descriptions of media elements.
• Applicability criteria: all long descriptions of images and media elements.

```
//img/@longdesc
//object//@href
```

• Test procedure:
  1. Check that the long description referenced by the longdesc or href attribute is available.
  2. Check that it describes the element.

• Expected results: FAIL if #1 or #2 is false.

• Fully automatable: no.

2.2.1.1.5  Test 1.1_HTML_05

This test is targeted to find complex images and non-text content that require a long description.

• Applicability criteria: all img and object elements without long description

```
//img[not(@longdesc)]
//object
```

• Test procedure:
  1a. For each img element check that it does not require a long description.
  1b. For each object element that does not contain or reference a long description, check that it does not require a long description.

• Expected results: PASS if 1a and 1b are true. FAIL if 1a or 1b is false.

• Fully automatable: no.

2.2.1.1.6  Test 1.1_HTML_06

This test is targeted to non-text content embedded with the non-standard embed element.

Since there is no defined method of providing alternatives for embed, embed is inherently inaccessible.

• Applicability criteria: all elements in the body of the page

```
//body/**
```
- **Test procedure:**
  Check that no embed elements are present.
- **Expected results:** PASS if true. FAIL if false.
- **Fully automatable:** yes.

### 2.2.1.1.7 Test 1.1_HTML_07

This test is targeted to check for text alternatives for non-text content loaded into an inline frame.

- **Applicability criteria:** all iframe elements.

```
//iframe
```

- **Test procedure:**
  1. Check if the element loads non-text content.
  2. If #1 is true, check that the element contains a text alternative or a link to a text alternative for the non-text content.
  
  Note that the iframe content cannot be updated when the content loaded by the iframe changes as a result of user interaction or script execution.

- **Expected results:** PASS if #2 is true. FAIL if #2 is false.
- **Fully automatable:** no.

### 2.2.1.1.8 Test 1.1_HTML_08

This test is targeted to check for frames that directly load non-text content.

- **Applicability criteria:** all frame elements.

```
//frame
```

- **Test procedure:**
  Check that the element directly loads non-text content.

- **Expected results:** FAIL if true.
- **Fully automatable:** no.

### 2.2.1.1.9 Test 1.1_HTML_09

This test is targeted to find embedded or linked audio-only components that convey information but have no text transcript.

- **Applicability criteria:** all audio-only components that convey information.
2.2.1.10  Test 1.1_HTML_10

This test is targeted to analyse text transcripts of embedded or linked audio-only components.

- Applicability criteria: All audio-only components with a transcript.

- Test procedure:
  Check that the text transcript fully describes all the important information in the audio track(s) of the audio-only component, including spoken words and non-spoken sounds such as sound effects.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: no.

2.2.1.2  Tests for external objects

2.2.1.2.1  Test 1.1_external_01

This test is targeted to find linked or embedded multimedia presentations that convey information but have no associated captions.

- Applicability criteria: all multimedia presentations that convey information with at least one audio and at least one video track.

- Test procedure:
  Check that all applicable components have associated captions.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: no.
2.2.1.2.2  Test 1.1_external_02

This test is targeted to analyse the associated captions of multimedia presentations.

- Applicability criteria: all multimedia presentations with at least one audio and at least one video track.

- Test procedure:
  Check that the associated captions fully convey all the important information in the audio track(s) of the multimedia presentation, including spoken words and non-spoken sounds such as sound effects.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: no.

2.2.2  Checkpoint 1.2

Provide redundant text links for each active region of a server-side image map. [Priority 1]

(See http://www.w3.org/TR/WCAG10-TECHS/#tech-redundant-server-links and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-redundant-server-links)

2.2.2.1  (X)HTML tests

2.2.2.1.1  Test 1.2_HTML_01

This test is targeted to find active regions of a server-side image map without redundant text links.

- Applicability criteria: all server-side image maps.

- Test procedure:
  1. Identify all active regions of the image map.
  2. Check that there is a redundant text link for each active region.

- Expected results: FAIL if #2 is false (for at least one active region).

- Fully automatable: no.
2.2.3 Checkpoint 1.3

Until user agents can automatically read aloud the text equivalent of a visual track, provide an auditory description of the important information of the visual track of a multimedia presentation. [Priority 1]

(See http://www.w3.org/TR/WCAG10-TECHS/#tech-auditory-descriptions and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-auditory-descriptions)

2.2.3.1 Tests for external objects

2.2.3.1.1 Test 1.3_external_01

This test is targeted to find multimedia presentations that convey information but have no auditory description of the important information of their visual track.

- Applicability criteria: all multimedia presentations, that convey information, with at least one visual track.

```
//object
//applet
//a
```

- Test procedure:
  Check that there is an auditory description of the important information of the visual track.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: no.

2.2.3.1.2 Test 1.3_external_02

This test is targeted to analyse the auditory description of linked and embedded multimedia presentations.

- Applicability criteria: all multimedia presentations with at least one visual track.

```
//object
//applet
//a
```

- Test procedure:
  Check that the auditory description effectively conveys all important visual elements of the presentation including information about actors, actions, body language, graphics, and scene changes.
• Expected results: PASS if true. FAIL if false.
• Fully automatable: no.

2.2.4 Checkpoint 1.4

For any time-based multimedia presentation (e.g., a movie or animation), synchronize equivalent alternatives (e.g., captions or auditory descriptions of the visual track) with the presentation. [Priority 1]

(See http://www.w3.org/TR/WAI-WEBCONTENT/#tech-synchronize-equivalents and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-synchronize-equivalents)

2.2.4.1 Tests for external objects

2.2.4.1.1 Test 1.4_external_01

This test is targeted to check the synchronisation of equivalent alternatives for multimedia presentations.

• Applicability criteria: all multimedia presentations with equivalent alternatives.

```html
//a
//applet
//object
```

• Test procedure:
Check that the equivalent alternative (captions, auditory descriptions or other equivalent alternative, as applicable) is synchronised with the presentation.

• Expected results: PASS if true. FAIL if false.
• Fully automatable: no.

2.3 Guideline 2

"Don't rely on color alone."
(See http://www.w3.org/TR/WCAG10/#gl-color)

This guideline provides information on how to use colour appropriately.

2.3.1 Checkpoint 2.1

Ensure that all information conveyed with color is also available
without color, for example from context or markup. [Priority 1]

(See http://www.w3.org/TR/WCAG10/#tech-color-convey and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-color-convey)

2.3.1.1  (X)HTML tests

2.3.1.1.1  Test 2.1_HTML_01

This test is targeted to find phrases in text that refer to parts of a document only by mentioning their colour.

- Applicability criteria: all text.

```
//body
```

- Test procedure:
  1. Check that the text does not refer to parts of a document only by mentioning their colour.
  2. If #1 is false check that references in text via colour are redundant.

- Expected results: PASS if #1 or #2 is true. FAIL if #1 and #2 are false.

- Fully automatable: no.

2.3.1.1.2  Test 2.1_HTML_02

This test is targeted to find phrases in non-text content that refer to parts of a document only by mentioning their colour.

- Applicability criteria: all text in non-text content.

```
//img
//area
//input[@type='image']
//applet
//object
```

- Test procedure:
  1. Check that the text does not refer to parts of a document only by mentioning their colour.
  2. If #1 is false check that references in text via colour are redundant.

- Expected results: PASS if #1 or #2 is true. FAIL if #1 and #2 are false.

- Fully automatable: no.
2.3.1.1.3  Test 2.1_HTML_03

This test is targeted to find coloured elements without redundant methods of conveying the information.

- Applicability criteria: all coloured html elements.

```html
/*@color
/*@bgcolor
/*@text
```

- Test procedure:
  Check that the colour information is redundant, in other words, colour is not the only method to provide the information.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: no.

2.3.1.2  CSS tests

2.3.1.2.1  Test 2.1_CSS_01

This test is targeted to find coloured elements without redundant methods of conveying the information.

- Applicability criteria: all coloured content produced by CSS.

```css
color
background-color
background
border-color
border
outline-color
outline
```

- Test procedure:
  Check that the colour information is redundant.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: no.

2.3.2  Checkpoint 2.2

Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen. [Priority 2 for images, Priority 3 for text].

(See http://www.w3.org/TR/WCAG10/#tech-color-contrast and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-
2.3.2.1 (X)HTML tests

2.3.2.1.1 Test 2.2_HTML_01

This test is targeted to find images without sufficient colour contrast.

- Applicability criteria: all images:

```html
//img
//area
//input[@type='image']
//object
```

- Test procedure: Check that the contrast between foreground and background colour is sufficient to convey the information. For both colours, calculate the value \( Y' = ((R\times299) + (G\times587) + (B\times114)) \), where \( R \), \( G \) and \( B \) are the decimal values for the red, green and blue colour components (each in the range \([0,255]\))\(^3\). Check that the difference between the two calculated values \( Y' \) is at least 66.8.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: yes.

2.3.2.2 CSS tests

2.3.2.2.1 Test 2.2_CSS_01

This test is targeted to find images without sufficient colour contrast.

- Applicability criteria: images referenced from CSS styles:

```css
background-image
background
content
cursor
list-style-image
```

- Test procedure: Check that the contrast between foreground and background colour is sufficient to convey the information. For both colours, calculate the value \( Y' = ((R\times299) + (G\times587) + (B\times114)) \), where \( R \), \( G \) and \( B \) are the decimal values for the red, green and blue colour components (each in the range \([0,255]\))\(^3\). Check that the difference between the two calculated values \( Y' \) is at least 66.8.

- Expected results: PASS if true. FAIL if false.

---

3 A thorough analysis in the BenToWeb project led to a recommendation of this algorithm, which is borrowed from Ridpath, C. and Chisholm, W. (Eds), (2000). Techniques For Accessibility Evaluation And Repair Tools. Available at http://www.w3.org/TR/AERT.

4 These CSS properties can specify the URI of an image to use as content or background.
• Fully automatable: yes.

2.4 Guideline 3

"Use markup and style sheets and do so properly."
(See http://www.w3.org/TR/WCAG10/#gl-structure-presentation)

2.4.1 Checkpoint 3.1

When an appropriate markup language exists and is supported, use markup rather than raster-based images to convey information. [Priority 2]
(See http://www.w3.org/TR/WCAG10/wai-pageauth.html#tech-use-markup and the errata at http://www.w3.org/WAI/GL/WAI-WEBCONTENT-ERRATA, the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-use-markup)

2.4.1.1 (X)HTML tests

2.4.1.1.1 Test 3.1_HTML_01

This test is targeted to check that there are no images containing text that can be replaced by markup constructs.

• Applicability criteria: all images.

```html
//img
//input[@type='image']
//object
```

• Test procedure:
  1. Check that the image contains text.
  2. If #1 is true check that the image can be replaced by a markup construct without loss of information\(^5\) conveyed by the image.

• Expected result: FAIL if #2 is true.

• Fully automatable: no.

2.4.1.1.2 Test 3.1_HTML_02

This test is targeted to check that there are no images of mathematical

---

\(^5\) You may use text in images when:
  1. the text does not convey its literal meaning but has a more graphical function, such as a logo and
  2. the effect can not be achieved with CSS and
  3. you have provided a text equivalent for the image
equations that can be replaced by markup constructs.

- Applicability criteria: all images.

```html
//img
//input[@type='image']
//object
```

- Test procedure:
  1. Check that the image contains a mathematical equation.
  2. If yes, check that the image can be replaced by a markup construct.

- Expected result: FAIL if #2 is true.
- Fully automatable: no.

2.4.1.1.3 Test 3.1_HTML_03

This test is targeted to check that there are no bitmap images without text or mathematical equations that can be replaced by markup that is supported.

- Applicability criteria: all bitmap images without text or mathematical equations.

```html
//img
//input[@type='image']
//object
```

- Test procedure:
  Check that there is a supported markup language that is able to convey the information in the image.

- Expected result: PASS if false.
- Fully automatable: no.

2.4.2 Checkpoint 3.2

Create documents that validate to published formal grammars. [Priority 2]

(See [http://www.w3.org/TR/WAI-WEBCONTENT/#tech-identify-grammar](http://www.w3.org/TR/WAI-WEBCONTENT/#tech-identify-grammar) and the techniques in [http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-identify-grammar](http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-identify-grammar))

---

6 You may use text in images when:
   1. the text does not convey its literal meaning but has a more graphical function, such as a logo and
   2. the effect can not be achieved with CSS and
   3. you have provided a text equivalent for the image

7 Since there are no formal grammars against which JavaScript code can be validated, UWEM
2.4.2.1  (X)HTML tests

2.4.2.1.1  Test 3.2_HTML_01

This test is targeted to check that the document contains a valid document type declaration.


- Applicability criteria: content preceding the HTML element of any HTML 4.x or XHTML 1.0 document.

- Test procedure:
  Check that the doctype declaration is valid. In XHTML 1.0 documents, the public identifier in the DOCTYPE declaration must reference one of three DTDs using the respective Formal Public Identifier.\(^8\)

- Expected result: PASS if true. FAIL if false.

- Fully automatable: yes.

2.4.2.1.2  Test 3.2_HTML_02

This test is targeted to find violations against the formal schema for HTML 4.x or XHTML 1.0.

- Applicability criteria: Any HTML 4.x or XHTML 1.0 document.

- Test procedure:
  a) For HTML, check that the document validates against the specified document type using a validating SGML parser.
  b) For XHTML, check that the document is well formed and that it validates against the specified document type using a validating XML parser.

- Expected result: FAIL if false.

- Fully automatable: yes.

---

2.4.2.2 CSS tests

2.4.2.2.1 Test 3.2_CSS_01

This test is targeted to find violations against the formal grammar for CSS 1.0 or CSS 2.x.

- Applicability criteria: Any CSS style rules.

- Test procedure:
  a) For style rules inside the style element or in the style attributes in an (X)HTML file: check that they conform to the formal grammar defined at http://www.w3.org/TR/REC-CSS2/grammar.html with a SAC parser.

  b) For CSS files: check that parsing each CSS file with a SAC parser causes no errors.

  Note that the W3C's “CSS Validator” does more than checking CSS rules against the formal grammar: it also checks for (un-)defined properties and their values, which are not included in the grammar. The grammar does not define the actual “vocabulary” of CSS. When using the W3C CSS Validator for this test, error messages and warnings that are not related to the formal grammar (for example, error messages regarding properties not defined in the CSS specification) must be ignored. Only error messages related to the formal grammar count as failures for this test.

- Expected result: PASS if true. FAIL if false.

- Fully automatable: yes.

2.4.3 Checkpoint 3.3

Use style sheets to control layout and presentation. [Priority 2]

(See http://www.w3.org/TR/WAI-WEBCONTENT/#tech-style-sheets and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-style-sheets)

---

2.4.3.1 **(X)HTML tests**

2.4.3.1.1 **Test 3.3_HTML_01**

This test is targeted to find whitespace that is used to control spacing between characters within words.

Note: There is no language-independent definition of the term “word”, so evaluators need to check that if the term “word” is applicable to the language of the content they are evaluating and, if yes, make sure that they understand what the term “word” means in the language of the content they are evaluating.

- **Applicability criteria:** any “word” containing whitespace.

  ```text()
  text()
  ```

- **Test procedure:**
  Check that the whitespace is not used to convey emphasis or importance.

- **Expected result:** PASS if true. FAIL if false.

- **Fully automatable:** no.

2.4.3.1.2 **Test 3.3_HTML_02**

This test is targeted to determine if layout or presentation of one or more elements has been achieved via means other than CSS.

- **Applicability criteria:** elements and attributes that can be used to position or influence presentation

  ```
  //img
  //font
  //td (in layout table)
  //th (in layout table)
  //center
  //u
  //b
  //i
  //blink
  //strong (unless used semantically)
  //em (unless used semantically)
  /*@align
  /*@border
  /*@hspace
  /*@vspace
  /*@bgcolor
  ```

- **Test procedure:**
  Check that the resulting position and/or presentation could not be achieved using style sheets.
• Expected result: FAIL if false.

• Fully automatable: no.

2.4.4 Checkpoint 3.4

Use relative rather than absolute units in markup language attribute values and style sheet property values. [Priority 2]

(See http://www.w3.org/TR/WCAG10/wai-pageauth.html#tech-relative-units and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-relative-units)

2.4.4.1 (X)HTML tests

2.4.4.1.1 Test 3.4_HTML_01

This test is targeted to check for relative values in (X)HTML attributes of type %Length;

• Applicability criteria: attributes that specify height, width, cell padding, cell spacing or a character offset as a number of pixels or a percentage.

  //table/@cellpadding
  //table/@cellspacing
  //col/@charoff
  //colgroup/@charoff
  //tbody/@charoff
  //td/@charoff
  //tfoot/@charoff
  //th/@charoff
  //thead/@charoff
  //tr/@charoff
  //iframe/@height
  //td/@height
  //th/@height
  //img/@height
  //object/@height
  //applet/@height
  //hr/@width
  // iframe/@width
  //img/@width
  //object/@width
  //table/@width
  //td/@width
  //th/@width
  //applet/@width

• Test procedure: Check that the value of the attribute is a percentage value (positive integer + '%') or that it is an absolute value that does not cause a loss of content or functionality when content is resized up to 200 percent and down to 50 percent without assistive
technology.

- Expected result: PASS if true. FAIL if false.
- Fully automatable: no.

2.4.4.1.2 Test 3.4_HTML_02

This test is targeted to check for relative values in (X)HTML attributes of type multi-length\(^{10}\) ("%MultiLength;" in the HTML 4.01 DTD).

- Applicability criteria: attributes that specify the width of columns or column groups.

  
  ```xml
  //col/@width
  //colgroup/@width
  ```

- Test procedure:
  Check that the value of the attribute is a percentage value (positive integer + ' %') or an * (asterisk) value or that it is an absolute value that does not cause a loss of content or functionality when content is resized up to 200 percent and down to 50 percent without assistive technology.

- Expected result: PASS if true. FAIL if false.
- Fully automatable: no.

2.4.4.1.3 Test 3.4_HTML_03

This test is targeted to check for relative values in (X)HTML attributes of type multi-length-list\(^{11}\) or ("%MultiLengths;" in the HTML 4.01 DTD: a comma-separated list of MultiLength).

- Applicability criteria: attributes that specify a list of lengths in pixels, a percentage or a relative value.

  ```xml
  //frameset/@cols
  //frameset/@rows
  ```

- Test procedure:
  Check that each value listed in the attribute is a percentage value (positive integer + ' %') or an * (asterisk) value or that it is an absolute value that does not cause a loss of content or functionality when content is resized up to 200 percent and down to 50 percent without assistive technology.

- Expected result: PASS if true. FAIL if false.

\(^{10}\) http://www.w3.org/TR/1999/REC-html401-19991224/types.html#type-multi-length

\(^{11}\) http://www.w3.org/TR/1999/REC-html401-19991224/types.html#type-multi-length
• Fully automatable: no.

2.4.4.2 CSS tests

2.4.4.2.1 Test 3.4_CSS_01

This test is targeted to check for relative value units in CSS properties that may contain <length> values.

• Applicability criteria\(^\text{12}\): CSS properties that specify length, width, height, size, spacing or offset.

<table>
<thead>
<tr>
<th>property</th>
</tr>
</thead>
<tbody>
<tr>
<td>background-position</td>
</tr>
<tr>
<td>border-spacing</td>
</tr>
<tr>
<td>bottom</td>
</tr>
<tr>
<td>font-size</td>
</tr>
<tr>
<td>height</td>
</tr>
<tr>
<td>left</td>
</tr>
<tr>
<td>letter-spacing</td>
</tr>
<tr>
<td>line-height</td>
</tr>
<tr>
<td>marker-offset</td>
</tr>
<tr>
<td>max-height</td>
</tr>
<tr>
<td>max-width</td>
</tr>
<tr>
<td>min-height</td>
</tr>
<tr>
<td>min-width</td>
</tr>
<tr>
<td>right</td>
</tr>
<tr>
<td>size</td>
</tr>
<tr>
<td>text-indent</td>
</tr>
<tr>
<td>text-shadow</td>
</tr>
<tr>
<td>top</td>
</tr>
<tr>
<td>vertical-align</td>
</tr>
<tr>
<td>width</td>
</tr>
<tr>
<td>word-spacing</td>
</tr>
</tbody>
</table>

• Test procedure:
  1. Check that the unit of the value is not cm, mm, in, pt, pc or px\(^\text{13}\).
  2. Check that the value is not xx-small, x-small, small, medium, large, x-large or xx-large\(^\text{14}\).
  3. If #1 or #2 are false, check that the absolute value does not interfere with the readability of any text element.

• Expected result: PASS if #1 and #2 are true, or if #3 is true. FAIL if #3 is false.

• Fully automatable: no.

\(^\text{12}\) This includes shorthands derived from the CSS properties list in the applicability criteria.

\(^\text{13}\) The CSS 2.0 specification lists ‘px’ (pixel) as a relative unit. However, the size of a pixel is relative to the computer display, not to any properties defined in web content. The CSS 2.0 specification also defines a "reference pixel" with an absolute size.

\(^\text{14}\) These are defined as absolute values at http://www.w3.org/TR/1998/REC-CSS2-19980512/fonts.html#value-def-absolute-size.
2.4.5 Checkpoint 3.5

Use header elements to convey document structure and use them according to specification. [Priority 2]

(See http://www.w3.org/TR/WCAG10/wai-pageauth.html#tech-logical-headings and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-logical-headings)

2.4.5.1 (X)HTML tests

2.4.5.1.1 Test 3.5_HTML_01

This test is targeted to find markup constructs that conceptually represent headings, but are not marked up with hx elements.

- Applicability criteria: the body of a web page.

```html
//body/*
```

- Test procedure:
  1. Select markup constructs that conceptually represent headings.
  2. Check whether the headings are marked up with hx elements.

  Hint: candidates for insufficient markup include, but are not limited to, combinations of font-weight/font-style changes (HTML b, i elements; CSS font-weight, font-style properties) and font-size enlargements (HTML big, font elements; CSS font-size property). This list is not exhaustive.

- Expected result: PASS if true.

- Fully automatable: no.

2.4.5.1.2

2.4.5.1.3 Test 3.5_HTML_03

This test is targeted to check that no levels are skipped in the heading hierarchy.

- Applicability criteria: all heading elements except h1 and h2\(^{15}\).

```html
//h3
//h4
//h5
//h6
```

\(^{15}\) h1 and h2 don't have to be checked because there is no header level which is two levels higher.
• Test procedure:
  Check that the inspected heading element does not skip one or more
  levels in the structure (for example, check that for h5 the preceding
  heading element is either h4, h5 or h6).

• Expected result: PASS if true.

• Fully automatable: yes.

2.4.5.1.4  Test 3.5_HTML_04

This test is targeted to check if heading elements have been used
(improperly) for font formatting.

• Applicability criteria: all heading elements (h1, ..., h6)

```html
//h1
//h2
//h3
//h4
//h5
//h6
```

• Test procedure: Check that headings are not used to create font
  formatting effects.

• Expected result: PASS if true. FAIL if false.

• Fully automatable: no.

2.4.5.1.5  Test 3.5_HTML_05

This test is targeted to check for the correct heading level hierarchy.

• Applicability criteria: the whole document.

```html
//body
```

• Test procedure:
  Check that the heading elements convey the logical structure of the
  document.

• Expected result: PASS if true.

• Fully automatable: no.

2.4.6  Checkpoint 3.6

Mark up lists and list items properly. [Priority 2]

(See http://www.w3.org/TR/WCAG10/#tech-list-structure and
Encode list structure and list items (UL, OL, DL, LI) properly. The HTML list elements DL, UL, and OL (available in HTML 3.2 and HTML 4.0) should only be used to create lists, not for formatting effects such as indentation. When possible and appropriate, use ordered (numbered) lists to help navigation.

2.4.6.1 (X)HTML tests

2.4.6.1.1 Test 3.6_HTML_01

Authors can disable the default list style of ordered and unordered list and manually create multi-level numbering (for example, 1, 1.1, 1.2, 1.2.1). This test is targeted to check that manually added list numbering conveys the depth of the list to users.

- Applicability criteria: all nested ordered and unordered lists with manually inserted multi-level numbers.

  //li/ol
  //li/ul

- Test procedure:
  Check that the numbering does not skip levels or numbers.

- Expected result: PASS if true.

- Fully automatable: no.

2.4.6.1.2 Test 3.6_HTML_02

This test is targeted to find out whether the List elements (li) are appropriate for the context of the document, in other words, to find out whether List elements are used to create lists, not for formatting such as indentation.

- Applicability criteria: all list item elements, including definitions in definition lists.

  //ul/li
  //ol/li
  //dl/dd

- Test procedure:
  Check each that li or dd elements are used to mark up list items and not for formatting effects.

- Expected result: PASS if true. FAIL if false.
• Fully automatable: no.

2.4.6.1.3 \hspace{1em} \textit{Test 3.6\_HTML\_03}

This test is targeted to find paragraphs, line breaks and numbers that are used to simulate numbered lists and which can be replaced with the ol element.

• Applicability criteria: all paragraphs starting with a counter (number or character that indicates an order or sequence).

\[
//p \\
//p//br
\]

• Test procedure:
1. Check that the document does not contain sequences of paragraphs that start with counters to simulate numbered lists.
2. Check that the document does not contain paragraphs with line breaks followed by counters to simulate numbered lists.

• Expected result: PASS if #1 and #2 are true.

• Fully automatable: yes.

2.4.6.1.4 \hspace{1em} \textit{Test 3.6\_HTML\_04}

This test is targeted to find paragraphs, line breaks and certain characters such as asterisk and hyphens that are used to simulate unordered lists and which can be replaced with the ul element.

• Applicability criteria: all sequences of lines that simulate list items.

\[
//p \\
//p//br
\]

• Test procedure:
1. Check that the document does not contain sequences of paragraphs that simulate unordered lists\textsuperscript{16}.
2. Check that the document does not contain paragraphs with line breaks to simulate unordered lists.

• Expected result: PASS if #1 and #2 are true. FAIL if #1 or #2 are false.

• Fully automatable: no.

2.4.6.1.5 \hspace{1em} \textit{Test 3.6\_HTML\_05}

This test is targeted to find paragraphs, line breaks and images displaying

\textsuperscript{16} Lists are often simulated by sequences of lines that start with an asterisk or a hyphen.
numbers that are used to simulate ordered lists and which can be replaced with the ol element and CSS.

- Applicability criteria: all paragraphs starting with images displaying a number or other types of counters.

```xml
//p//img
//p//br/following-sibling::img
```

- Test procedure:
  1. Check that the document does not contain sequences of paragraphs that start with images displaying numbers or other types of counters to simulate ordered lists.
  2. Check that the document does not contain paragraphs with line breaks followed by images of consecutive numbers or other types of counters to simulate numbered lists.

- Expected result: PASS if #1 and #2 are true. FAIL if #1 or #2 are false.

- Fully automatable: no.

2.4.6.1.6  Test 3.6_HTML_06

This test is targeted to find paragraphs, line breaks and images (especially bullet images) that are used to simulate unordered lists and which can be replaced with the ul element and CSS.

- Applicability criteria: all paragraphs starting with bullet images.

```xml
//p//img
//p//br/following-sibling::img
```

- Test procedure:
  1. Check that the document does not contain sequences of paragraphs that start with images of bullets to simulate unordered lists.
  2. Check that the document does not contain paragraphs with line breaks followed by images of bullets to simulate unordered lists.

- Expected result: PASS if #1 and #2 are true. FAIL if #1 or #2 are false.

- Fully automatable: no.

2.4.6.1.7  Test 3.6_HTML_07

This test is targeted to find paragraphs, line breaks and formatting effects that are used to simulate definition lists and which can be replaced with the dt and dd elements.
• Applicability criteria: all paragraphs starting with a term followed by a definition.

\[//p
//p//br\]

• Test procedure:
  Check that the document does not contain paragraphs that should be replaced by a definition list.

• Expected result: PASS if true. FAIL if false.

• Fully automatable: no.

2.4.7  Checkpoint 3.7

Mark up quotations. Do not use quotation markup for formatting effects such as indentation. [Priority 2]

(See http://www.w3.org/TR/WCAG10/#tech-quotes and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-quotes

2.4.7.1  (X)HTML tests

2.4.7.1.1  Test 3.7_HTML_01

This test is targeted to check that quotation elements are used properly to mark up quotations and not for formatting or indentation effects.

• Applicability criteria: all blockquote elements.

\[/blockquote\]

• Test procedure:
  Check that the blockquote is used to mark up a quotation.

• Expected result: PASS if true. FAIL if false.

• Fully automatable: no.

2.4.7.1.2  Test 3.7_HTML_02

This test is targeted to check that short quotations (q element) are used properly for quotations and not for layout purposes.

• Applicability criteria: all q elements.

\[/q\]

• Test procedure:
  Check that the q element is used to mark up a quotation.
• Expected result: PASS if true. FAIL if false.
• Fully automatable: no.

2.4.7.1.3 Test 3.7_HTML_03

This test is targeted to find quotations that have not been marked up with q or blockquote.

• Applicability criteria: all text.

```html
//p
```

• Test procedure:
  1. Are there any quotations in the selected paragraphs, for example, passages that contain quotation marks in the markup or have CSS-generated quotation marks?
  2. If yes, check that the quotations are marked up with q or blockquote.

• Expected result: PASS if #2 is true. FAIL if #2 is false.
• Fully automatable: no.

2.4.7.1.4 Test 3.7_HTML_04

This test is targeted to find any cite and address elements that are used to italicise text.

• Applicability criteria: all cite and address elements.

```html
//cite
//address
```

• Test procedure:
  1. Select any cite and address elements.
  2. Determine if they are used to italicise text instead of marking up a citation or providing information on the author of the document, respectively.

• Expected result: FAIL if #2 is true.
• Fully automatable: no.

2.5 Guideline 4

“Clarify natural language usage.”
(See http://www.w3.org/TR/WAI-WEBCONTENT/#gl-abbreviated-and-foreign)
This guideline provides information on how to facilitate pronunciation or interpretation of abbreviated or foreign text.

2.5.1 Checkpoint 4.1

Clearly identify changes in the natural language of a document's text and any text equivalents (e.g., captions). [Priority 1]

(See http://www.w3.org/TR/WAI-WEBCONTENT/#tech-identify-changes and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-identify-changes)

2.5.1.1 (X)HTML tests

2.5.1.1.1 Test 4.1_HTML_01

This test is targeted to find changes in human language that are not marked up.

- Applicability criteria: all elements containing text.

```javascript
//*[true(text())]
```

- Test procedure:
  1. Select elements starting at the lowest level (the "leaves" in the tree structure), and move upwards to parent elements, checking every element in the document.
  2. For each text segment\textsuperscript{17} in the current element, determine the human language. Note that proper nouns, technical terms, and words or phrases that have become part of the language of the rest of the element are not considered as changes in human language for the purpose of this test.
  3. Check that the human language found in \#2 is the same for the rest of the content in the current element.
  4. Determine the human language of the text in the parent element. (If the parent element is the html element, its language is defined either by the lang attribute or from page context.)
  5. If two different human languages are found in \#5, check that the (current) element has a lang attribute that identifies the human language by means of the corresponding two-letter code defined in ISO639\textsuperscript{18}.

\textsuperscript{17} In many languages, a text segment corresponds to single words or groups of words; in some languages, especially those with a logographic (sometimes called "ideographic") writing system, a text segment can even correspond to a single character.

\textsuperscript{18} The HTTP 'Content-Language' header is not taken into account here because this header should not be used for text processing. See http://www.w3.org/TR/i18n-html-tech-lang/#ri20040808.110827800 for further information.
2.5.1.1.2 Test 4.1_HTML_02

This test is targeted to find changes in human language that are not marked up.

- Applicability criteria: all attributes that specify alternative text, advisory information (the title attribute), a table summary, a label in a hierarchical menu (the label attribute), a standby text (the standby attribute) or other textual content.

```
//img/@alt
//applet/@alt
//area/@alt
//input/@alt
//meta/@content
//option/@label
//optgroup/@label
//object/@standby
//table/@summary
/**/@title")
//input[@type='text']/@value
//input[@type='submit']/@value
//frame/@name
//iframe/@name
```

- Test procedure:
  1. Determine the human language of the text in the attribute.
  2. Determine the human language of the text in the nearest ancestor of the attribute's parent element.
  3. If the languages found in steps 1 and 2 are different, check that the attribute's parent element has a lang attribute that identifies the human language of the attribute value by means of the corresponding two-letter code defined in ISO639.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: no.

2.5.1.1.3 Test 4.1_HTML_03

This test is targeted to find block-level elements with changes in text direction (for human languages) that are not marked up.

- Applicability criteria: all block-level elements containing text.\(^{20}\)

---

19 All elements except base, basefont, head, html, meta, param, script and title.
20 Note the following information from the section "Inheritance of text direction information" in HTML 4.01 (http://www.w3.org/TR/html401/struct/dirlang.html#blocklevel-bidi):
Test procedure:
1. Determine the direction of the text in the (current) element.
2. If the current element is the html or body element and the text-direction if right to left, check that the element has a dir attribute with the value "rtl".
3. If the current element is not the html element, determine the direction of the text in the parent element.
4. If two different text directions are found in step 3, check that the (current) element has a dir attribute that identifies the text direction by means of the corresponding value (rtl or ltr).

Expected results: PASS if #2 and #4 are true. FAIL if #2 or #4 is false.

Fully automatable: no.

2.5.1.1.4 Test 4.1_HTML_04

This test is targeted to find changes in text direction (for human languages) that are not marked up.

Applicability criteria: all attributes that specify alternative text, advisory information (the title attribute), a table summary, a label in a hierarchical menu (the label attribute), a standby text (the standby attribute) or other textual content.

"The Unicode bidirectional algorithm requires a base text direction for text blocks. To specify the base direction of a block-level element, set the element's dir attribute. The default value of the dir attribute is "ltr" (left-to-right text).

When the dir attribute is set for a block-level element, it remains in effect for the duration of the element and any nested block-level elements. Setting the dir attribute on a nested element overrides the inherited value.

(...)

Inline elements, on the other hand, do not inherit the dir attribute. This means that an inline element without a dir attribute does not open an additional level of embedding with respect to the bidirectional algorithm. (Here, an element is considered to be block-level or inline based on its default presentation. Note that the INS and DEL elements can be block-level or inline depending on their context.)"

Also note the following information from the section "The effect of style sheets on bidirectionality" in HTML 4.01 (http://www.w3.org/TR/1999/REC-html401-19991224/struct/dirlang.html#style-bidi):

"In general, using style sheets to change an element's visual rendering from block-level to inline or vice-versa is straightforward. However, because the bidirectional algorithm relies on the inline/block-level distinction, special care must be taken during the transformation. When an inline element that does not have a dir attribute is transformed to the style of a block-level element by a style sheet, it inherits the dir attribute from its closest parent block element to define the base direction of the block. When a block element that does not have a dir attribute is transformed to the style of an inline element by a style sheet, the resulting presentation should be equivalent, in terms of bidirectional formatting, to the formatting obtained by explicitly adding a dir attribute (assigned the inherited value) to the transformed element."
• Test procedure:
  1. Determine the direction of the text in the attribute.
  2. Check that the parent element has a dir attribute that identifies the text direction by means of the corresponding value (rtl or ltr).

• Expected results: PASS if #2 is true. FAIL if #2 is false.

• Fully automatable: no.

2.5.1.2  CSS tests

2.5.1.2.1  Test 4.1_CSS_01

This test is targeted to find each CSS style that generates text in a different human language than the language of the parent element of the element or elements for which the style is defined.

Note the following information from the CSS 2.0 specification: "As their names indicate, the :before and :after pseudo-elements specify the location of content before and after an element's document tree content" (emphasis added)\(^\text{22}\). For this reason, the language of the generated content should be the same as the language of the parent of the element for which the CSS style is defined.

• Applicability criteria: each text string generated by CSS styles.

```css
*:after {content: "...";}
*:before {content: "...";}
```

• Test procedure:
  1. Determine the language of the text string.
  2. Determine the language defined for or inherited by the parent element of the element for which the CSS style is defined.
  3. Check that the languages found in #1 and #2 are the same.

---

\(^\text{21}\) All elements except base, basefont, head, html, meta, param, script and title.

\(^\text{22}\) http://www.w3.org/TR/1998/REC-CSS2-19980512/generate.html#before-after-content
• Expected results: PASS if #3 true. FAIL if #3 is false.
• Fully automatable: no.

2.6 Guideline 5

“Create tables that transform gracefully.”
(See http://www.w3.org/TR/WCAG10/#gl-table-markup)

This guideline provides information on how to identify properly marked up tables.

2.6.1 Checkpoint 5.1

For data tables, identify row and column headers. [Priority 1]
(See http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-table-headers)

2.6.1.1 (X)HTML tests

2.6.1.1.1 Test 5.1_HTML_01

This test is targeted to find data tables that do not have row and column headers.

• Applicability criteria: all data tables.

```html
//table
```

• Test procedure:
1. Select the data cells in the data table.
2. For each data cell, check that there is a row header cell and a column header cell that can be identified by the sections "Algorithm to find heading information" or "Associating header information with data cells" in HTML 4.01.23

• Expected results: PASS if #2 is true.
• Fully automatable: No.

2.6.1.1.2 Test 5.1_HTML_02

This test is targeted to identify preformatted text used to display tabular information. Preformatted text does not have mechanisms to specify row and column headings.

• Applicability criteria: preformatted text.

23 http://www.w3.org/TR/1999/REC-html401-19991224/struct/tables.html#h-11.4
• Test procedure:
  Determine if the preformatted text is visually rendered as a table.

• Expected results: FAIL if true.

• Fully automatable: no.

2.6.1.2  CSS Tests

2.6.1.2.1  Test 5.1_CSS_01

This test is targeted to identify tabular information that is not created by means of table markup.

• Applicability criteria: any tabular information that is not created by means of table markup.

  /*[not(self::table)]

• Test procedure:
  Check that the visual grid-like presentation is achieved by means of CSS (absolute positioning or table-related values for the display property).

• Expected results: FAIL if true.

• Fully automatable: no.

2.6.2  Checkpoint 5.2

For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells. [Priority 1]

(See http://www.w3.org/TR/WCAG10-TECHS/#tech-table-structure and the techniques in http://www.w3.org/TR/WCAG10-HTML-TECHS/#identifying-table-rows-columns)

2.6.2.1  (X)HTML tests

2.6.2.1.1  Test 5.2_HTML_01

This test is targeted to identify tables with two or more logical levels of rows or columns that are not marked up properly by using table markup that associates rows and columns.

• Applicability criteria: data tables where content in each data cell has a relationship with at least two row headers and/or at least two
column headers.

```xml
//table
```

- **Test procedure:**
  1. For each data cell, check that at least one of the following applies:
     1.a. the headers attribute contains a space-separated list of all the values of the id attributes of the header cells which with the data cell has a relationship;
     1.b. each column header cell that provides header information for the rest of the column that contains it, has a scope attribute with the value 'col'; each column header cell that provides header information for the rest of the column group that contains it, has a scope attribute with the value 'colgroup'; each row header cell that provides header information for the rest of the row that contains it, has a scope attribute with the value 'row'; each row header cell that provides header information for the rest of the row group that contains it, has a scope attribute with the value 'rowgroup'.

- **Expected results:** PASS if #1.a or #1.b is true.

- **Fully automatable:** no.

### 2.6.2.1.2 Test 5.2_HTML_02

This test is targeted to determine if header cells in a heading with two or more levels are categorised consistently. This test does not require that axis should always be used, but that the categories identified by the attribute are appropriate or logical.

- **Applicability criteria:** table headers with two or more levels.

```xml
//table[count(descendant::tr[th]) > 1]
//table[count(descendant::tr[td[@scope]]) > 1]
//table[descendant::tr[count(th) > 1]]
//table[descendant::tr[count(td[@scope]) > 1]]
//table[descendant::td[boolean(substring-after(substring-after(normalize-space(@headers), ' '), ' '), ' ')]]
```

- **Test procedure:**
  For each header cell in a table header with two or more levels, check that any axis attribute consistently labels the category to which the header cell belongs.
  Note that the value of the axis attribute is a label that may be presented to a user, instead of a merely machine-readable class or name.

- **Expected results:** FAIL if false.
2.6.2.1.3 Test 5.2_HTML_03

This test is targeted to find inconsistent structuring of tables. This test does not require that colgroup, thead, tfoot or tbody should always be used, but that their use is appropriate or logical.

- Applicability criteria: tables defining column groups, table headings, table footers and table bodies.
  
  ```html
  //table[colgroup]
  //table[thead]
  //table[tfoot]
  //table[tbody]
  ```

- Test procedure:
  Check that each of the selected elements correctly structures the table.

- Expected results: FAIL if false.

- Fully automatable: no.

2.6.3 Checkpoint 5.3

*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).* [Priority 2]

(See [http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-avoid-table-for-layout](http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-avoid-table-for-layout))

2.6.3.1 (X)HTML tests

2.6.3.1.1 Test 5.3_HTML_01

This test is targeted to find layout tables that do not convey the same information when linearised.

- Applicability criteria: layout tables.

  ```html
  //table
  ```

- Test procedure:
  Check that the table conveys the same information when linearised.24

---

24 The W3C has a Table Linearizer Web Service at [http://www.w3.org/WAI/ER/WG/tabletest/tablin](http://www.w3.org/WAI/ER/WG/tabletest/tablin).
• Expected results: PASS if true. FAIL if false.
• Fully automatable: no.

2.6.4 Checkpoint 5.4

If a table is used for layout, do not use any structural markup for the purpose of visual formatting. [Priority 2]

(See http://www.w3.org/TR/WCAG10-TECHS/#tech-table-structure and the techniques in http://www.w3.org/TR/WCAG10-HTML-TECHS/#tech-table-layout)

2.6.4.1 (X)HTML tests

2.6.4.1.1 Test 5.4_HTML_01

This test is targeted to check that table headers are only used in data tables.

• Applicability criteria: tables with header cells.

```xml
//table[descendant::th]
//table[descendant::td[@scope]]
//table[descendant::td[@axis]]
```

• Test procedure:
  Check that the table is a data table.

• Expected results: PASS if true. FAIL if false.
• Fully automatable: no.

2.6.4.1.2 Test 5.4_HTML_02

This test is targeted to check that table headers and table footers are only used in data tables.

• Applicability criteria: tables with headers and/or footers.

```xml
//table[thead]
//table[tbody]
```

• Test procedure:
  Check that the table is a data table.

• Expected result: PASS if true. FAIL if false.
• Fully automatable: no.
2.6.4.1.3  Test 5.4_HTML_03

This test is targeted to check that id and headers attributes are only used in data tables.

- Applicability criteria: tables with one or more data cells with a headers attribute and one or more header cells with the id attribute.

```xml
//table[descendant::th[@id]]
//table[descendant::td[@id]]
//table[descendant::td[@headers]]
```

- Test procedure:
  Check that the table is a data table.

- Expected result: PASS if true. FAIL if false.

- Fully automatable: no.

2.6.4.1.4  Test 5.4_HTML_04

This test is targeted to check that captions are only used for data tables.

- Applicability criteria: tables with a caption.

```xml
//table[descendant::caption]
```

- Test procedure:
  Check that the table is a data table.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: no.

2.6.4.1.5  Test 5.4_HTML_05

This test is targeted to check that cells are only categorised in data tables.

- Applicability criteria: Tables in which cells are categorised by means of the axis attribute.

```xml
//table[descendant::th[@axis]]
//table[descendant::td[@axis]]
```

- Test procedure:
  Check that the table is a data table.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: no.
2.7 Guideline 6

“Ensure that pages featuring new technologies transform gracefully.”
(See http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/#gl-new-technologies)

This guideline provides information on ensuring that pages are accessible even when newer technologies are not supported or are turned off.

2.7.1 Checkpoint 6.1

Organize documents so 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. [Priority 1]
(See http://www.w3.org/TR/WCAG10-TECHS/#tech-order-style-sheets and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-order-style-sheets)

2.7.1.1 (X)HTML tests

2.7.1.1.1 Test 6.1_HTML_01

This test analyses the effect on the readability of the document of CSS applied in standalone style sheets, embedded style sheets and style attributes of document elements.

Note that the CSS 2.0 specification defines a style sheet as "A set of statements that specify presentation of a document"\(^{25}\). This includes the statements in style attributes.

- Applicability criteria: HTML 4.01 and XHTML 1.0 documents with one or more associated style sheets. This includes style sheets attached to documents by means of HTTP headers (see the section "Linking to style sheets with HTTP headers"\(^{26}\) in HTML 4.01).

```
//link[@rel='stylesheet']
//link[@rel='alternate stylesheet']
/style
/*@style
```

- Test procedure:
  1. Switch off, remove or deactivate all associated style sheets.
  2. Check that content does not become invisible.

\(^{25}\) http://www.w3.org/TR/1998/REC-CSS2-19980512/conform.html#x10
\(^{26}\) http://www.w3.org/TR/1999/REC-html401-19991224/present/styles.html#h-14.6
3. Check that content is not obscured by other content.
4. Check that the meaning is not changed by changes in reading order caused by step 1.

- Expected results: PASS if #2-4 are true. FAIL if #2, #3 or #4 is false.
- Fully automatable: no.

2.7.2 Checkpoint 6.2

Ensure that equivalents for dynamic content are updated when the dynamic content changes. [Priority 1]

(See http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-dynamic-source and the techniques in http://www.w3.org/TR/WCAG10-TECHS/#tech-dynamic-source)

2.7.2.1 (X)HTML tests

2.7.2.1.1 Test 6.2_HTML_01

This test analyses the text equivalent of any non-text content loaded into a frame.

- Applicability criteria: any non-text content referenced by the src attribute of a frame element.

```xml
document(//frame/@src)
document(//iframe/@src)
```

- Test procedure:
  Check that there is a text equivalent that describes the current version of the non-text content.

- Expected results: PASS if true. FAIL if false.
- Fully automatable: no.

2.7.2.1.2 Test 6.2_HTML_02

This test analyses the text equivalent of any non-text content loaded into the frame by the browser as a result of link activation or script execution.

- Applicability criteria: any non-text content that is loaded into a frame as a result of the activation of a link or the execution of a script.

```xml
//script
```
Test procedure:
Check that there is a text equivalent that describes the current version of the non-text content.

Expected results: PASS if true. FAIL if false.

Fully automatable: no.

2.7.2.1.3 Test 6.2_HTML_03

This text is targeted to check that there are appropriate equivalents for non-text content that is added to the markup or the Document Object Model (DOM) by means of scripts.

Applicability criteria: any non-text content that is added to the markup or the DOM by means of scripts.
• Test procedure:
  Check that there is a text equivalent that describes the current version of
  the non-text content.

• Expected results: PASS if true. FAIL if false.

• Fully automatable: no.

2.7.3  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. [Priority 1]

(See http://www.w3.org/TR/WCAG10-TECHS/#tech-scripts and
the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-
TECHS/#tech-scripts)

2.7.3.1 (X)HTML tests

2.7.3.1.1 Test 6.3_HTML_01

This test determines whether information and functionality provided by
embedded content is also available without said content.

• Applicability criteria: applets (in a wide sense, not limited to Java;
  this includes Flash).

  //applet
  //object

• Test procedure:
  1. Disable support for applets.
  2. Check that the page is usable, in other words, that all
     functionality is still available.
  3. If the page is not usable , check that there is an alternative
     accessible page with equivalent information.

• Expected results: PASS if #2 or #3 are true. FAIL if #2 and #3 are
  false.

• Fully automatable: no.

2.7.3.1.2 Test 6.3_HTML_02

This test determines whether essential information and functionality
provided by script is also available when script is not executed.
• Applicability criteria: (functionality provided by) scripts.

```xml
//script
//a[@href='javascript:']
/*/onfocus
/*/onblur
/*/onkeypress
/*/onkeydown
/*/onkeyup
/*/onsubmit
/*/onreset
/*/onselect
/*/onchange
/*/onload
/*/onunload
/*/onclick
/*/onmousedown
/*/onmouseup
/*/onmouseover
/*/onmouseout
/*/onmousemove
/*/onmouseout
```

• Test procedure:
  1. Turn off or disable (the browser's support for) scripts.
  2. Check that the page is usable, in other words, that all essential functionality is still available.
  3. If the page is not usable, check that there is an alternative accessible page with equivalent information.

• Expected results: PASS if #2 or #3 are true. FAIL if #2 and #3 are false.

• Fully automatable: no.

### 2.7.4 Checkpoint 6.4

*For scripts and applets, ensure that event handlers are input device-independent. [Priority 2]*

(See [http://www.w3.org/TR/WAI-WEBCONTENT/#tech-keyboard-operable-scripts](http://www.w3.org/TR/WAI-WEBCONTENT/#tech-keyboard-operable-scripts) and the techniques in [http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-keyboard-operable-scripts](http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-keyboard-operable-scripts))

#### 2.7.4.1 (X)HTML tests

2.7.4.1.1 Test 6.4_HTML_01

This test is targeted to check that mouse-specific event handlers have a keyboard-specific (or device-independent) version.
- Applicability criteria: elements with attributes for mouse-specific events, including event attributes added by scripts, except links and form controls with the onclick attribute.

```html
/*@onclick
/*@onmousedown
/*@onmouseup
/*@onmouseout
/*@onmouseover
/*@onmouseover
```

- Test procedure:
  1. Check that the element is a link or a form control with an onclick attribute.
  2. If step 1 is true, check that the link or form control has no onkeypress attribute or the script code for onkeypress excludes the TAB key (numeric code: 09).
  3. If step 1 is false, check that each of these elements has a keyboard-specific event handler attribute that triggers exactly the same function or functions as the mouse-specific event handler attribute.

The following table maps device-specific event handlers:

<table>
<thead>
<tr>
<th>Mouse-specific</th>
<th>Keyboard-specific or device-independent</th>
</tr>
</thead>
<tbody>
<tr>
<td>onmousedown</td>
<td>onkeydown</td>
</tr>
<tr>
<td>onmouseup</td>
<td>onkeyup</td>
</tr>
<tr>
<td>onclick²⁷</td>
<td>onkeypress</td>
</tr>
<tr>
<td>onmouseover</td>
<td>onfocus</td>
</tr>
<tr>
<td>onmouseout</td>
<td>onblur</td>
</tr>
</tbody>
</table>

- Expected results: PASS if #1, #2 and #3 are true. FAIL if #2 or #3 are false.

- Fully automatable: yes.

2.7.4.1.2 Test 6.4_HTML_02

This test is targeted to check for mouse-specific event handlers for which no device-independent or keyboard-specific equivalent handlers are defined in the HTML 4 specification.

²⁷ The onclick event handler is keyboard accessible when it is attached to a link or a form control. However, when attached to other elements, onclick is device dependent. When both onclick and onkeypress are attached to a link or a form control, tabbing causes the onkeypress event to fire in Opera and Firefox. Therefore, onkeypress must not be added to links or form controls that already have onclick, or the TAB key (numeric code: 09) must be explicitly excluded in the script.
• Applicability criteria:

```javascript
//*[@ondblclick]
//*[@onmousemove]
```

• Test procedure:
  1. Select any elements with a mouse-specific event handler attribute that does not have a keyboard-specific event handler attribute that executes exactly the same function.
  2. Check that the functions performed by the event handlers can also implemented in a mouse-independent way.

• Expected results: PASS if #2 is true. FAIL if #2 is false.

• Fully automatable: no.

2.7.4.2 Tests for external objects

2.7.4.2.1 6.4_external_01

This test is targeted to check event handlers in applets are device-independent.

• Applicability criteria: applets (in a wide sense, not limited to Java; this includes Flash).

- Test procedure:
  1. Select any applets loaded in a page.
  2. Check that each function can be triggered through a keyboard interface.

• Expected results: PASS if #2 is true. FAIL if #2 is false.

• Fully automatable: no.

2.7.5 Checkpoint 6.5

Ensure that dynamic content is accessible or provide an alternative presentation or page. [Priority 2]

(See http://www.w3.org/TR/WCAG10-TECHS/#tech-fallback-page and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-fallback-page)

This checkpoint handles the accessibility of content that is dynamic, for example programmatic objects, scripts, audio, video and multimedia content, frames (including inline frames). There are two ways of creating dynamic content: server-side dynamic content and client-side dynamic
content. In terms of (semi-)automatic testing, we are only interested in the client-side dynamic content, because we cannot really identify the presence of server-side dynamic content.

2.7.5.1 (X)HTML tests

2.7.5.1.1 Test 6.5_HTML_01

This test is targeted to find framesets with inaccessible dynamic content.

Note: checkpoint 6.3 (Priority 1) already requires that essential functionality is still available when scripts, applets etcetera are disabled.

- **Applicability criteria:** framesets.

```xml
//frameset
```

- **Test procedure:**
  1. Check that the dynamic content is accessible.
  2. If #1 is false, check that the frameset contains a noframes element.
  3. If #2 is true, check that the noframes element provides an alternative presentation or a link to an alternative presentation.

- **Expected results:** PASS if #1 or #3 is true. FAIL if #2 or #3 are false.

- **Fully automatable:** no.

2.7.5.1.2

2.7.5.1.3 Test 6.5_HTML_03

This test is targeted to find links that use javascript.

- **Applicability criteria:** links with the 'javascript:' pseudo-protocol.

```xml
//a[starts-with(@href, 'javascript:')]`
```

- **Test procedure:**
  Check that there is an alternative control or an alternative link without the 'javascript:' pseudo-protocol that provides the same functionality.

- **Expected results:** PASS if true. FAIL if false.

- **Fully automatable:** no.
2.8  Guideline 7

“Ensure user control of time-sensitive content changes.”
(See http://www.w3.org/TR/WAI-WEBCONTENT/#gl-movement)

This guideline provides information on moving, blinking, scrolling, or auto-updating objects or pages, which make it difficult, sometimes even impossible, to read or access content.

2.8.1  Checkpoint 7.1

Until user agents allow users to control flickering, avoid causing the screen to flicker. [Priority 1]

Note. People with photosensitive epilepsy can have seizures triggered by flickering or flashing in the 4 to 59 flashes per second (Hertz) range with a peak sensitivity at 20 flashes per second as well as quick changes from dark to light (like strobe lights).

(See http://www.w3.org/TR/WAI-WEBCONTENT/#tech-avoid-flicker and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-avoid-flicker)

Note: for checkpoints with an "until user agents" clause, WCAG 1.0 refers to the document "User Agent Support for Accessibility"28 for information about user agent support for accessibility features. The current version of this document (last updated in on 11 August 2005) states: "Netscape Navigator (versions, platform), Microsoft Internet Explorer (versions, platform), and Opera (versions, platform) allow the user to turn off loading of images, scripts, and applets. Turning these off will allow the user to avoid flicker caused by images, scripts, and applets. For other plug-ins the user can choose not to load the plug-in. However, it would be ideal if users could stop, pause, or step through animations, scripts, or other dynamic content that may cause flicker as discussed in the UAGL checkpoint 3.7 and UAGL checkpoint 3.10."

2.8.1.1  (X)HTML tests

2.8.1.1.1  Test 7.1_HTML_01

This test is targeted to find marquee text that causes blinking. Marquee does not normally cause blinking, but certain combinations of scroll amount, scroll delay, font size and colour might cause parts of the screen to blink.

28 http://www.w3.org/WAI/Resources/WAI-UA-Support
• Applicability criteria: marquee elements.

 Test procedure:
  1. Check that the marquee does not cause flickering or flashing in the 4 to 59 flashes per second (Hertz) range.
  2. If false, check that the flicker is an unavoidable aspect of the presentation and that it can be controlled by the user, in other words, that it can be turned on and off.

• Expected results: PASS if #1 or #2 is true. FAIL if #1 and #2 are false.

• Fully automatable: no.

2.8.1.1.2  Test 7.1_HTML_02

This test is targeted to find animated GIF files that cause flicker. (Other image file types for inclusion in HTML pages – JPEG and PNG – do not support animation.)

• Applicability criteria: animated GIF files.

 Test procedure:
  1. Check that the playback speed of frames and the colour contrast between subsequent frames do not cause flickering or flashing in the 4 to 59 flashes per second (Hertz) range.
  2. If false, check that the flicker is an unavoidable aspect of the presentation and that it can be controlled by the user, in other words, that it can be turned on and off.

• Expected results: PASS if #1 or #2 is true. FAIL if #1 and #2 are false.

• Fully automatable: no.

2.8.1.1.3  Test 7.1_HTML_03

This test is targeted to find client-side scripts that cause flicker or flashing.

• Applicability criteria: client-side scripts.

29 The selector //object[@type='image/gif'] will miss certain GIF files if the object element's type attribute is not defined or incorrect.
Test procedure:
1. Check that the script does not cause flicker or flashing at a rate between 4 and 59 Hertz.
2. If false, check that the flicker is an unavoidable aspect of the presentation and that it can be controlled by the user, in other words, that it can be turned on and off.

Expected results: PASS if #1 or #2 is true. FAIL if #1 and #2 are false.

Fully automatable: no.

2.8.1.2 CSS tests
2.8.1.2.1 Test 7.1_CSS_01

This test is targeted to find CSS-generated content that causes flicker or flashing.

Applicability criteria: objects (images, videos, animations) embedded into a web page by means of CSS.

Test procedure:
1. Check that the object does not cause flicker at a rate between 4 and 59 Hertz.
2. If false, check that the flicker is an unavoidable aspect of the presentation and that it can be controlled by the user, in other words, that it can be turned on and off.

Expected results: PASS if #1 or #2 is true. FAIL if #1 and #2 are false.

Fully automatable: no.
2.8.1.3 Test for external objects

2.8.1.3.1 Test 7.1_external_01

This test is targeted to find Java applets that cause flicker or flashing.

- Applicability criteria: Java applets.

```xml
//object[@codetype='application/java']
//object[@codetype='application/java-archive']
//object[starts-with(@codetype, 'application/x-java-applet')]
//applet
  any content sent by HTTP with MIME types 'application/java',
  'application/java-archive', 'application/x-java-applet'
```

- Test procedure:
  1. Check that the applet does not cause flicker or flashing at a rate between 4 and 59 Hertz.
  2. If false, check that the flicker is an unavoidable aspect of the presentation and that it can be controlled by the user, in other words, that it can be turned on and off.

- Expected results: PASS if #1 or #2 is true. FAIL if #1 and #2 are false.

- Fully automatable: no.

2.8.1.3.2 Test 7.1_external_02

This test is targeted to find any video content that cause flicker or flashing.

- Applicability criteria: video content.

```xml
//object[starts-with(@type, 'video/')]
```

- Test procedure:
  1. Check that the video content does not cause flicker or flashing at a rate between 4 and 59 Hertz.
  2. If false, check that the flicker is an unavoidable aspect of the presentation and that it can be controlled by the user, in other words, it can be turned on and off.

- Expected results: PASS if #1 or #2 is true. FAIL if #1 and #2 are false.

- Fully automatable: no.
2.8.2 Checkpoint 7.2

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). [Priority 2]

(See http://www.w3.org/TR/WAI-WEBCONTENT/#tech-avoid-blinking and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-avoid-blinking)

2.8.2.1 (X)HTML tests

2.8.2.1.1 Test 7.2_HTML_01

This test is targeted to find any blink elements.

- Applicability criteria: all elements in the body of the page

  //body/*

- Test procedure:
  Check that no blink elements are present.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: Yes.

2.8.2.1.2 Test 7.2_HTML_02

This test is targeted to find animated GIF files that cause blinking. (Other image file types for inclusion in HTML pages – JPEG and PNG – do not support animation.)

- Applicability criteria: Animated GIF files.

  //img
  //object

- Test procedure:
  Check that the image does not cause blinking that violates the general flash and red flash thresholds. Definition of blink: to change the presentation, for example, turn on and off, at a rate of four times per second or slower.

  For the general flash threshold, a flash is defined as a pair of opposing changes in relative luminance of 10% or more and the relative luminance of the darker image is below 0.80. An "opposing

30 The selector //object[@type='image/gif'] will miss certain GIF files if the object element’s type attribute is not defined or incorrect.
32 Note that this applies to very small changes in relative luminance.
change" is an increase followed by a decrease, or a decrease followed by an increase.
For the red flash threshold, a flash is defined as any transition to or from a saturated red33.

- Expected results: PASS if true. FAIL if false.
- Fully automatable: Yes, if contrast threshold is known.

2.8.2.1.3 Test 7.2_HTML_03
This test is targeted to find scripts that cause blinking.

- Applicability criteria: scripts.

```plaintext
script
  /**/@onfocus
  /**/@onblur
  /**/@onkeypress
  /**/@onkeydown
  /**/@onkeyup
  /**/@onsubmit
  /**/@onreset
  /**/@onselect
  /**/@onchange
  /**/@onload
  /**/@onunload
  /**/@onclick
  /**/@ondblclick
  /**/@onmousedown
  /**/@mouseup
  /**/@onmouseover
  /**/@onmousemove
  /**/@mouseout
```

- Test procedure:
  Check that the script does not cause blinking.

- Expected results: PASS if true. FAIL if false.
- Fully automatable: No.

2.8.2.2 CSS tests
2.8.2.2.1 Test 7.2_CSS_01
This test is targeted to find CSS-generated content that causes blinking.

- Applicability criteria: images, video and animations generated by CSS styles.

```plaintext
*:after {content: url(...);}
```

• Test procedure:
  Check that the content does not cause blinking.

• Expected results: PASS if true. FAIL if false.

• Fully automatable: No.

2.8.2.2.2  Test 7.2_CSS_02

This test is targeted to find CSS rules that cause content to blink.

• Applicability criteria: CSS rules with text-decoration: blink

* { text-decoration: blink; }

• Test procedure:
  Check that there are no CSS rules with text-decoration: blink.

• Expected results: PASS if true. FAIL if false.

• Fully automatable: yes.

2.8.2.3  Tests for external objects

2.8.2.3.1  Test 7.2_external_01

This test is targeted to find Java applets that cause blinking.

• Applicability criteria: applets (Java, Flash or other).

//applet
//object

• Test procedure:
  Check that the applet does not cause blinking.

• Expected results: PASS if true.

• Fully automatable: no.

2.8.2.3.2  Test 7.2_external_02

This test is targeted to find any video content that causes blinking.

• Applicability criteria: video.

---

34 Object elements for Java applets can have a codetype attribute with the values 'application/java', 'application/java-archive' or 'application/x-java-applet'. The test is also applicable to any content sent by HTTP with the MIME types 'application/java', 'application/java-archive' or 'application/x-java-applet'.
2.8.3 Checkpoint 7.3

Until user agents allow users to freeze moving content, avoid movement in pages. [Priority 2]

(See http://www.w3.org/TR/WAI-WEBCONTENT/#gl-movement and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-avoid-movement)

2.8.3.1 (X)HTML tests

2.8.3.1.1 Test 7.3_HTML_01

This test is targeted to find marquee elements.

- Applicability criteria: all elements in the body of the page

```html
//body//*
```

- Test procedure:
  Check that no marque elements are present.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: yes.

2.8.3.1.2 Test 7.3_HTML_02

This test is targeted to check for scripts that cause movement.

- Applicability criteria: Scripts.

```html
/*/@onfocus
/*@onblur
/*@onkeypress
/*@onkeydown
/*@onkeyup
/*@onsubmit
/*@onreset
```

35 Object element for video content can have a type attribute that starts with the value 'video///'. The test is also applicable to any content sent by HTTP with a MIME type that starts with 'video///'.

24 November 2007 Page 64 of 100
• Test procedure:
  1. Check that the script does not cause movement.
  2. If false, check that the movement can be frozen.

• Expected results:
  PASS if #1 is true. PASS if #1 is false and #2 is true. FAIL if #1 and #2 are false.

• Fully automatable: no.

2.8.3.2 CSS tests

2.8.3.2.1 Test 7.3_CSS_01

This test is targeted to find CSS-generated content that causes movement.

• Applicability criteria: CSS-generated content.

```css
*:after {content: url(...);}
*:before {content: url(...);}
```

• Test procedure:
  Check that the content does not cause movement.

• Expected results: PASS if true. FAIL if false.

• Fully automatable: no.

2.8.3.3 Tests for external objects

2.8.3.3.1 Test 7.3_external_01

This test is targeted to check for external objects that cause or contain movement.

• Applicability criteria: applets (Java, Flash or other).

```html
//applet
//object
```

36 Object elements for Java applets can have a codetype attribute with the values
Test procedure:
1. Check that the content of the applet does not contain or cause movement.
2. If false, check that the movement can be frozen.

Expected results:
PASS if #1 is true. PASS if #1 is false and #2 is true. FAIL if #1 and #2 are false.

Fully automatable: no.

2.8.3.3.2 Test 7.3_external_02

This test is targeted to check for video content that causes or contains movement.

Applicability criteria: video.

Test procedure:
1. Check that the video content does not contain or cause movement.
2. If false, check that the movement can be frozen.

Expected results:
PASS if #1 is true.
PASS if #1 is false and #2 is true.
FAIL if #1 and #2 are false.

Fully automatable: no.

2.8.4 Checkpoint 7.4

Until user agents provide the ability to stop the refresh, do not create periodically auto-refreshing pages. [Priority 2]

(See http://www.w3.org/TR/WAI-WEBCONTENT/#gl-movement and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-no-periodic-refresh)

'application/java', 'application/java-archive' or 'application/x-java-applet'. The test is also applicable to any content sent by HTTP with the MIME types 'application/java', 'application/java-archive' or 'application/x-java-applet'.

37 Object element for video content can have a type attribute that starts with the value 'video/'. The test is also applicable to any content sent by HTTP with a MIME type that starts with 'video/'.

24 November 2007
2.8.4.1  (X)HTML tests

2.8.4.1.1  Test 7.4_HTML_01

This test is targeted to find elements that can cause page refreshing.

- Applicability criteria: meta elements with http-equiv="refresh".

```xml
/meta[@http-equiv='refresh']
```

- Test procedure:
  1. Check that the meta element contains a URI in its content attribute.
  2. If true, check that the URI in the meta content attribute is not equal to the URI of the HTML resource containing the meta element.

- Expected results: PASS if #1 and #2 are true. FAIL if #1 is false. FAIL if #1 is true and #2 is false.

- Fully automatable: yes.

2.8.4.1.2  Test 7.4_HTML_02

This test is targeted to find scripts objects that can cause page refreshing.

- Applicability criteria: scripts.

```xml
/script
/*@onfocus
/*@onblur
/*@onkeypress
/*@onkeydown
/*@onkeyup
/*@onsubmit
/*@onreset
/*@onselect
/*@onchange
/*@onload
/*@onunload
/*@onclick
/*@ondblclick
/*@onmousedown
/*@onmouseup
/*@onmouseover
/*@onmousemove
/*@onmouseout
```

- Test procedure:
  1. Check that the script does not cause page refreshing.
  2. If false, check that the refreshing can be stopped.

- Expected results:
  PASS if #1 or #2 is true. FAIL if #1 and #2 are false.
• Fully automatable: no.

2.8.4.2 Tests for external objects

2.8.4.2.1 Test 7.4_external_01

This test is targeted to check for external objects that can cause page refreshing.

• Applicability criteria: external objects.

```xml
//applet
//object
```

• Test procedure:
  1. Check that the object does not cause page refreshing.
  2. If false, check that the refreshing can be stopped.

• Expected results:
  PASS if #1 or #2 is true. FAIL if #1 and #2 are false.

• Fully automatable: no.

2.8.5 Checkpoint 7.5

Until user agents provide the ability to stop auto-redirect, do not use markup to redirect pages automatically. Instead, configure the server to perform redirects. [Priority 2]

(See http://www.w3.org/TR/WAI-WEBCONTENT/#gl-movement and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-no-auto-forward)

2.8.5.1 (X)HTML tests

2.8.5.1.1 Test 7.5_HTML_01

This test is targeted to find elements that can cause page redirecting.

• Applicability criteria: meta elements with http-equiv="refresh".

```xml
//meta[@http-equiv='refresh']
```

• Test procedure:
  Check that the URI in the meta element's content attribute is equal to the URI of the document containing the meta element.

• Expected results: PASS if true. FAIL if false.

• Fully automatable: yes.
2.8.5.1.2  Test 7.5_HTML_02

This test is targeted to find scripts that cause redirecting without providing a mechanism to stop the redirect.

- Applicability criteria: scripts.

```html
//script
/**/@onfocus
/**/@onblur
/**/@onkeypress
/**/@onkeydown
/**/@onkeyup
/**/@onsubmit
/**/@onreset
/**/@onselect
/**/@onchange
/**/@onload
/**/@onunload
/**/@onclick
/**/@ondblclick
/**/@onmousedown
/**/@onmouseup
/**/@onmouseover
/**/@onmousemove
/**/@onmouseout
/**/@onmouseover
```

- Test procedure:
  1. Check that the script does not cause redirecting.
  2. If false, check that the redirecting can be stopped.

- Expected results:
  PASS if #1 is true. PASS if #1 is false and #2 is true. FAIL if #1 and #2 are false.

- Fully automatable: no.

2.8.5.2  Tests for external objects

2.8.5.2.1  Test 7.5_external_01

This test is targeted to check for external objects that can cause redirecting without providing a mechanism to stop this.

- Applicability criteria: external objects.

```html
//applet
//object
```

- Test procedure:
  1. Check that the object does not cause redirecting.
  2. If false, check that the redirecting can be stopped.

- Expected results:
PASS if #1 is true. PASS if #1 is false and #2 is true. FAIL if #1 and #2 are false.

- Fully automatable: no.

## 2.9 Guideline 8

"Ensure direct accessibility of embedded user interfaces."

(See [http://www.w3.org/TR/WCAG10/#gl-own-interface](http://www.w3.org/TR/WCAG10/#gl-own-interface))

This guideline provides information on how to create accessible embedded user interfaces.

### 2.9.1 Checkpoint 8.1

Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies [Priority 1 if functionality is important and not presented elsewhere, otherwise Priority 2.]

(See [http://www.w3.org/TR/WCAG10/#tech-directly-accessible](http://www.w3.org/TR/WCAG10/#tech-directly-accessible) and the techniques in [http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-directly-accessible](http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-directly-accessible))

### 2.9.1.1 (X)HTML tests

#### 2.9.1.1.1 Test 8.1_HTML_01

This test is targeted to find scripts that are not directly accessible or compatible with assistive technologies.

- Applicability criteria: scripts.
• Test procedure:
  1. If the script provides user agent functionality, check that it conforms to the User Agent Accessibility Guidelines 1.0\textsuperscript{38}.
  2. If the script provides authoring tool functionality, check that it conforms to the Authoring Tool Accessibility Guidelines 1.0 guideline 7\textsuperscript{39}.
  3. In other cases, check that the script is directly accessible or compatible with assistive technologies, either through one of the available Accessibility APIs or by ensuring that all functionality is available when only a keyboard interface is used.

• Expected results: PASS if #1 or #2 or #3 is true. FAIL if #1, #2 and #3 are false.

• Fully automatable: no.

2.9.1.2 Tests for external objects

2.9.1.2.1 Test 8.1\_external\_01

This test is targeted find external objects that are not directly accessible or compatible with assistive technologies.

• Applicability criteria: all external objects.

```xml
  //applet
  //object
```

• Test procedure:
  1. If the external object is a user agent, check that it conforms to the User Agent Accessibility Guidelines 1.0\textsuperscript{40}.
  2. If the external object is an authoring tool, check that it conforms to the Authoring Tool Accessibility Guidelines 1.0 guideline 7\textsuperscript{41}.
  3. In other cases, check that the object is directly accessible or compatible with assistive technologies, either through an Accessibility API if available or by ensuring that all functionality is available when only a keyboard interface is used.

• Expected results: PASS if #1 or #2 or #3 is true. FAIL if #1, #2 and #3 are false.

• Fully automatable: no.

\textsuperscript{38} http://www.w3.org/TR/2002/REC-UAAG10-20021217/
\textsuperscript{39} http://www.w3.org/TR/2000/REC-ATAG10-20000203/
\textsuperscript{40} http://www.w3.org/TR/2002/REC-UAAG10-20021217/
\textsuperscript{41} http://www.w3.org/TR/2000/REC-ATAG10-20000203/
2.10 Guideline 9

“Design for device-independence.”
(See http://www.w3.org/TR/WCAG10/#gl-device-independence)

This guideline provides information on how to create web content that does not rely on one specific input or output device.

2.10.1 Checkpoint 9.1

Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape. [Priority 1]
(See http://www.w3.org/TR/WCAG10/#tech-client-side-maps and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-client-side-maps)

2.10.1.1 (X)HTML tests

2.10.1.1.1 Test 9.1_HTML_01

This test is targeted to find server-side image maps.

- Applicability criteria: server-side image maps.

  //a//img[@ismap]
  //input[@type='image']

- Test procedure:
  Check that the server-side image map can not be replaced with a client-side image map.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: yes.

2.10.2 Checkpoint 9.2

Ensure that any element that has its own interface can be operated in a device-independent manner. [Priority 2]
(See http://www.w3.org/TR/WAI-WEBCONTENT/#tech-keyboard-operable and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-keyboard-operable)

42 This test is automatable as it will always produce a FAIL, because every geometric region can be defined by a polygon and every pixel can be defined as a circle with radius 0.
2.10.2.1 Tests for external objects

2.10.2.1.1 Test 9.2_external_01

This test is targeted to check for the device-independence of the interface of embedded or included objects.

- Applicability criteria: all embedded or included objects.

```
//applet
//object
```

- Test procedure:
  1. Check that the external object does not have its own interface.
  2. If false, check that any operation supported by the interface can be operated in a device-independent manner by using only a keyboard or keyboard alternative interface.

- Expected results:
  PASS if #1 or #2 is true. FAIL if #1 and #2 are false.

- Fully automatable: no.

2.10.3 Checkpoint 9.3

For scripts, specify logical event handlers rather than device-dependent event handlers. [Priority 2]

(See http://www.w3.org/TR/WAI-WEBCONTENT/#tech-device-independent-events and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-device-independent-events)

2.10.3.1 (X)HTML tests

2.10.3.1.1 Test 9.3_HTML_01

This test is targeted to analyse device-dependent event handlers.

- Applicability criteria: elements with device-dependent event handler attributes.

```
/**/@onclick
/**/@ondblclick
/**/@onkeydown
/**/@onkeypress
/**/@onkeyup
/**/@onmousedown
/**/@onmousemove
/**/@onmouseout
/**/@onmouseover
```
Test procedure:
1. Check that invoking the device-dependent event handler causes only purely decorative effects, such as the disappearance of an underline, a change in text colour or an image switch.
2. If #1 is false, check that replacing the device-dependent event-handler attribute with a device-independent one (such as onblur, onchange, onfocus, onload, onreset, onselect, onsubmit and onunload) causes loss of content or functionality.

Expected results: PASS if #1 true. PASS if #1 is false and #2 is true. FAIL if #2 is false.

Fully automatable: no.

2.11 Guideline 10

"Use interim solutions."
(See http://www.w3.org/TR/WCAG10/#gl-interim-accessibility)

2.11.1 Checkpoint 10.1

Until user agents allow users to turn off spawned windows, do not cause pop-ups or other windows to appear and do not change the current window without informing the user. [Priority 2]
(See http://www.w3.org/TR/WAI-WEBCONTENT/#tech-avoid-pop-ups and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-avoid-pop-ups)

2.11.1.1 (X)HTML tests

2.11.1.1.1 Test 10.1_HTML_01

This test is targeted to find any target attributes that cause content to be opened in a new window without informing the user.

Applicability criteria: a, area, form or link elements with a target attribute with a value that is not one of “_self”, “_parent” or “_top”.

//a[@target!='_self' and @target!='_parent' and @target!='_top']
//area[@target!='_self' and @target!='_parent' and @target!='_top']
//form[@target!='_self' and @target!='_parent' and @target!='_top']
//link[@target!='_self' and @target!='_parent' and @target!='_top']
Test procedure:
1. If the element is an a element with link text, check that there is text immediately before, in or immediately after the link text that warns the user that activating the link spawns a new window. If the link is part of a list of links, the warning can be before the list.
2. Else, if the element is an image link (a element containing img element and no text), check that there is text before the link, immediately after the link, or in the alt attribute of the image that warns the user that activating the link spawns a new window. If the link is part of a list of links, the warning can be before the list.
3. Else, if the element is an area in an image map, check that there is text in the area's alt attribute or before the image map that warns the user that activating the link spawns a new window.
4. Else, if the element is a form element, check that there is text immediately before or in the submit button that warns the user that activating the link spawns a new window.
5. Else, if the element is a link element, check that the link element's title attribute contains text that warns the user that activating the link spawns a new window.

Expected results: PASS if #1-#5 are true. FAIL if #1, #2, #3, #4 or #5 is false.

Fully automatable: no.

2.11.1.1.2 Test 10.1_HTML_02

This test is targeted to find base elements with a target attribute that cause content to be opened in a new window without informing the user.

Applicability criteria: base elements with a target attribute with a value that is not one of "_self", "_parent" or "_top".

//base[@target!='_self' and @target!='_parent' and @target!='_top']

Test procedure:
Check that the content warns the user that activating any a, area, form or link element spawns a new window.

Expected results: PASS if true. FAIL if false.

Fully automatable: no.

2.11.1.1.3 Test 10.1_HTML_03

This test is targeted to find scripts that cause content to be opened in a new window without informing the user.
• Applicability criteria: scripts that cause content to be opened in a new window.

```html
//script
/**/@onblur
/**/@onchange
/**/@onclick
/**/@ondblclick
/**/@onfocus
/**/@onkeydown
/**/@onkeypress
/**/@onkeyup
/**/@onload
/**/@onmousedown
/**/@onmousemove
/**/@onmouseout
/**/@onmouseover
/**/@onmouseup
/**/@onreset
/**/@onselect
/**/@onsubmit
/**/@onunload
/a[starts-with(@href, 'javascript:')]
```

• Test procedure:
Check that the content warns the user that the script spawns a new window.

• Expected results: PASS if true. FAIL if false

• Fully automatable: no.

### 2.11.2 Checkpoint 10.2

Until user agents support explicit associations between labels and form controls, for all form controls with implicitly associated labels, ensure that the label is properly positioned. [Priority 2]

(See [http://www.w3.org/TR/WCAG10/#tech-unassociated-labels](http://www.w3.org/TR/WCAG10/#tech-unassociated-labels) and the techniques in [http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-unassociated-labels](http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-unassociated-labels))

### 2.11.2.1 (X)HTML tests

#### 2.11.2.1.1 Test 10.2_HTML_01

This test is targeted to find form controls with improperly positioned, implicitly associated labels.

• Applicability criteria: visible form controls that can have labels.

```html
//input[not(@type='hidden')][not(@type='submit')][not(@type='button')][not(@type='reset')]
```
2.12 Guideline 11

Use W3C technologies and guidelines.

(See http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/#gl-use-w3c)

This guideline recommends using W3C technologies and describes what to do if other technologies are used.

2.12.1 Checkpoint 11.1

Use W3C technologies when they are available and appropriate for a task and use the latest versions when supported. [Priority 2]

(See http://www.w3.org/TR/WCAG10-TECHS/#tech-latest-w3c-specs and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-latest-w3c-specs)

2.12.1.1 (X)HTML tests

2.12.1.1.1 Test 11.1_HTML_01

This test is targeted to find out whether the latest versions of W3C technologies for HTML and XHML have been used.

- Applicability criteria: HTTP content header and document type declaration

- Test procedure:
  1. Check that the HTTP content type is text/html or application/xhtml+xml.
  2. If HTTP content type is text/html, check that the document type declaration matches the content.

43 checking whether the document type matches the content is covered by test 3.2_HTML_01 (validation).
is HTML 4.01 or XHTML 1.0.
3. If HTTP content type is application/xhtml+xml, check that the
document type is XHTML 1.1.

- Expected results: FAIL if any is false.
- Fully automatable: yes.

2.12.2 Checkpoint 11.2

Avoid deprecated features of W3C technologies. [Priority 2]
(See http://www.w3.org/TR/WCAG10-TECHS/#tech-avoid-deprecated and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-avoid-deprecated)

Note that “deprecated” can mean different things in different
specifications. In HTML 4.01 it means the following:

“A deprecated element or attribute is one that has been outdated by
newer constructs. Deprecated elements are defined in the reference
manual in appropriate locations, but are clearly marked as deprecated.
Deprecated elements may become obsolete in future versions of HTML.

User agents should continue to support deprecated elements for reasons
of backward compatibility.”

It is possible that a deprecated feature is deprecated in favour of another
feature that is not well supported (for example, object is intended to
replace applet).

2.12.2.1 (X)HTML tests

2.12.2.1.1 Test 11.2_HTML_01

This test is targeted to find deprecated HTML elements.

- Applicability criteria: deprecated elements in the HTML 4.01
  specification.
• Test procedure:
  Check that any of these elements are present.

• Expected results: PASS if false. FAIL if true.

• Fully automatable: yes.

2.12.2.1.2 Test 11.2_HTML_02

This test is targeted to find deprecated HTML attributes.

• Applicability criteria: deprecated attributes in the HTML 4.01 specification.

```
//body/@bgcolor
//body/@link
//body/@alink
//body/@vlink
//body/@background
//body/@text
//caption/@align
//iframe/@align
//img/@align
//img/@border
//img/@hspace
//img/@vspace
//object/@align
//object/@border
//object/@hspace
//object/@vspace
//input/@align
//legend/@align
//li/@value
//table/@align
//table/@bgcolor
//tr/@bgcolor
//th/@bgcolor
//th/@height
//th/@nowrap
//th/@width
//td/@bgcolor
//td/@height
//td/@nowrap
//td/@width
//hr/@align
//hr/@noshade
//hr/@size
//hr/@width
//p/@align
//div/@align
//h1/@align
//h2/@align
//h3/@align
//h4/@align
//h5/@align
//h6/@align
//br/@clear
```
• Test procedure:
  Check that any of these attributes are present.

• Expected results: PASS if false. FAIL if true.

• Fully automatable: yes.

2.12.3 Checkpoint 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. [Priority 1]

(See http://www.w3.org/TR/WCAG10-TECHS/#tech-alt-pages and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-alt-pages)

2.12.3.1 (X)HTML tests

2.12.3.1.1 Test 11.4_HTML_01

This test looks for the alternative content and checks whether it is equivalent.

• Applicability criteria: linked alternative page.

  document(/a/@href)
  document(/link/@href)

• Test procedure:
  1. Compare functionality and information content of the two delivery units.
  2. Check that they are equivalent.

• Expected results: PASS if #2 is true. FAIL if #2 is false.

---

44 Section 4.10 in the XHTML 1.0 specification: "The elements with 'id' and 'name' attributes": http://www.w3.org/TR/xhtml1/#h-4.10.
2.12.3.1.2 Test 11.4_HTML_02

This test looks for the alternative content and checks whether it is accessible.

- Applicability criteria: linked alternative page.

```html
document(//a[href])
document(//link[@href])
```

- Test procedure:
  1. Identify the alternative content.
  2. Check that it conforms to all tests for Priority 1 checkpoints in this section of UWEM other than this one, 11.4.

- Expected results: PASS if #2 is true. FAIL if #2 is false.

- Fully automatable: no.

2.12.3.1.3 Test 11.4_HTML_03

This test looks for the alternative content and checks whether the original content could have been made accessible.

- Applicability criteria: linked alternative page.

```html
document(//a[href])
document(//link[@href])
```

- Test procedure:
  1. Identify the alternative content.
  2. Check that the original content could not have been made accessible without undue effort.

- Expected results: PASS if #2 is true. FAIL if #2 is false.

- Fully automatable: no.

2.13 Guideline 12

*Provide context and orientation information.*

(See [http://www.w3.org/TR/WAI-WEBCONTENT/#gl-complex-elements](http://www.w3.org/TR/WAI-WEBCONTENT/#gl-complex-elements))

This guideline provides information on how to provide contextual and orientation information to help users understand complex pages or elements.
2.13.1 Checkpoint 12.1

Title each frame to facilitate frame identification and navigation. [Priority 1]

(See http://www.w3.org/TR/WAI-WEBCONTENT/#tech-frame-titles)

2.13.1.1 (X)HTML tests

2.13.1.1.1 Test 12.1_HTML_01

This test is targeted to find frames without description.

• Applicability criteria: frame elements without title attribute.

```
//frame[not(@title)]
//iframe[not(@title)]
```

• Test procedure:
  Check that any such frames are found.

• Expected results: FAIL if true.

• Fully automatable: yes.

2.13.1.1.2 Test 12.1_HTML_02

This test is targeted to check whether the title attribute identifies the frame.

• Applicability criteria: frame elements with title attribute.

```
//frame[@title]
//iframe[@title]
```

• Test procedure:
  1. Select elements.
  2. Check that the title identifies the frame.

• Expected results: PASS if #2 is true. FAIL if #2 is false.

• Fully automatable: no.

2.13.2 Checkpoint 12.2

Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone. [Priority 2]

(See http://www.w3.org/TR/WAI-WEBCONTENT/#tech-frame-titles)
longdesc)

2.13.2.1 (X)HTML tests

2.13.2.1.1 Test 12.2_HTML_01

This test is targeted to check whether the long description represents the context of the frame, if it is not clear by the frame title alone.

- Applicability criteria: documents referenced by a frame element's longdesc attribute or by an a element's href attribute in anoframes element, the purpose of which are to describe the context of a frame.

  document(//frame/@longdesc)
document(//noframes//a/@href)

- Test procedure:
  1. Select long description document referenced by the element.
  2. Check that the frame and its context are described by the text in the document, if not obvious by the frame title alone.

- Expected results: PASS if #2 is true. FAIL if #2 is false.

- Fully automatable: no.

2.13.3 Checkpoint 12.3

Divide large blocks of information into more manageable groups where natural and appropriate. [Priority 2]

(See http://www.w3.org/TR/WAI-WEBCONTENT/#tech-group-information)

2.13.3.1 (X)HTML tests

2.13.3.1.1 Test 12.3_HTML_01

This test is targeted to find fieldsets without legend.

- Applicability criteria: fieldset elements without legend child element.

  //fieldset[not(legend)]

- Test procedure:
  Check that any such fieldsets are found.

- Expected results: PASS if false. FAIL if true.

- Fully automatable: yes.
2.13.3.1.2 Test 12.3_HTML_02

This test is targeted to check whether the legend describes the meaning of the fieldset.

- Applicability criteria: legend elements.

```html
//legend
```

- Test procedure:
  Check that the legend represents the context of the fieldset parent element.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: no.

2.13.3.1.3 Test 12.3_HTML_03

This test is targeted to check whether the elements are grouped in a practical way.

- Applicability criteria: fieldset elements.

```html
//fieldset
```

- Test procedure:
  Check that form control elements in the fieldset are grouped in a practical way.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: no.

2.13.3.1.4 Test 12.3_HTML_04

This test is targeted to find optgroup elements without label.

- Applicability criteria: optgroup elements without a label attribute.

```html
//optgroup[not(@label)]
```

- Test procedure:
  Check that any such optgroup elements are found.

- Expected results: FAIL if true.

- Fully automatable: yes.
2.13.3.1.5  Test 12.3_HTML_05

This test is targeted to check whether the label describes the meaning of the optgroup.

- Applicability criteria: label attribute of optgroup elements.
  
  ```xml
  //optgroup/@label
  ```

- Test procedure:
  Check that the label represents the context of the optgroup.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: no.

2.13.3.1.6  Test 12.3_HTML_06

This test is targeted to check whether the option elements are grouped in a practical way.

- Applicability criteria: optgroup elements.

  ```xml
  //optgroup
  ```

- Test procedure:
  Check that the elements are grouped in a practical way.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: no.

2.13.3.1.7

2.13.3.1.8  Test 12.3_HTML_08

This test is targeted to check whether the caption describes the meaning of the table.

- Applicability criteria: caption element in a data table.

  ```xml
  //caption
  ```

- Test procedure:
  Check that caption describes the nature of the table.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: no.
2.13.3.1.9  Test 12.3_HTML_09

This test is targeted to check whether the elements are grouped in a practical way.

- Applicability criteria: thead, tbody, tfoot, and colgroup elements.

```html
//thead
//tbody
//tfoot
//colgroup
```

- Test procedure:
  1. Check that the thead element groups header information.
  2. Check that the tfoot element groups footer information.
  3. Check that the tbody element(s) group(s) rows of table data.
  4. Check that the colgroup elements group columns that belong together.

- Expected results: PASS if #1-4 are true. FAIL if #1, #2, #3 or #4 is false.

- Fully automatable: no.

2.13.3.1.10  Test 12.3_HTML_10

This test is targeted to check whether the elements are grouped in a practical way.

- Applicability criteria: ul, ol, and dl elements.

```html
//ul
//ol
//dl
```

- Test procedure:
  Check that ul, ol and dl elements group list items that belong together.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: no.

2.13.3.1.11  Test 12.3_HTML_11

This test is targeted to check whether the elements are structured in a practical way.

- Applicability criteria: paragraphs (p elements).

```html
//p
```

- Test procedure:
Check that the paragraph is used to structure text in a practical way.

- Expected results: PASS if true. FAIL if false.
- Fully automatable: no.

2.13.3.1.12  Test 12.3_HTML_12

This test is targeted to check whether form controls need grouping.

- Applicability criteria: form element without a fieldset descendant element.

```xml
//form[not(.//fieldset)]
```

- Test procedure:
  Check that the form controls in the form do not need grouping.

- Expected results: PASS if true. FAIL if false.
- Fully automatable: no.

2.13.3.1.13  Test 12.3_HTML_13

This test is targeted to check whether the options need grouping.

- Applicability criteria: select element without optgroup child element.

```xml
//select[not(optgroup)]
```

- Test procedure:
  Check that options in the select element do not need grouping.

- Expected results: PASS if true.
- Fully automatable: no.

2.13.3.1.14  Test 12.3_HTML_14

This test is targeted to check whether the table rows need grouping.

- Applicability criteria: tables without headers or footers.

```xml
//table[not(thead) or not(tfoot) or count(tbody)<2]
```

- Test procedure:
  Check that the table rows do not need grouping.

- Expected results: PASS if true.
- Fully automatable: no.
2.13.3.1.15  Test 12.3_HTML_15

This test is targeted to check whether text needs grouping with headings and paragraphs.

- Applicability criteria: body text.

```
//body//text()
```

- Test procedure:
  1. Check that chapters, subtopics and sections are introduced by headings.
  2. Check that the headings introduce or describe the chapters, subtopics or sections that they belong to.

- Expected results: PASS if #1 and #2 are true. FAIL if #1 or #2 are false.

- Fully automatable: no.

2.13.4  Checkpoint 12.4

Associate labels explicitly with their controls. [Priority 2]

(See [http://www.w3.org/TR/WAI-WEBCONTENT/#tech-associate-labels](http://www.w3.org/TR/WAI-WEBCONTENT/#tech-associate-labels))

2.13.4.1  (X)HTML tests

2.13.4.1.1  Test 12.4_HTML_01

This test is targeted to find form control elements without id.

- Applicability criteria: input or selection form control elements without id attribute.

```
//input[not(@type='hidden')][not(@type='submit')][not(@type='reset')][not(@type='button')][not(@type='image')][not(@id)]
//select[not(@id)]
//textarea[not(@id)]
```

- Test procedure:
  Check that no such elements are found.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: yes.

2.13.4.1.2  Test 12.4_HTML_02

This test is targeted to find form control elements without label element.
Unified Web Evaluation Methodology (UWEM) Deliverable D-WAB4 (Public)

- Applicability criteria: input or selection form control elements without associated label element.

```html
//input[@type[not(='hidden')]][not(='submit')][not(='reset')][not(='button')][not(='image')][@id]
//select[@id]
//textarea[@id]
```

- Test procedure:
  Check that there is a label element in the document, the for attribute of which equals the form control element's id attribute.

- Expected results: PASS if true. FAIL if false.

- Fully automatable: yes.

2.14 Guideline 13

*Provide clear navigation mechanisms.*

(See [http://www.w3.org/TR/WAI-WEBCONTENT/#gl-facilitate-navigation](http://www.w3.org/TR/WAI-WEBCONTENT/#gl-facilitate-navigation))

This guideline provides information on how to provide contextual and orientation information to help users understand complex pages or elements.

2.14.1 Checkpoint 13.1

*Clearly identify the target of each link.* [Priority 2]

(See [http://www.w3.org/TR/WAI-WEBCONTENT/#tech-meaningful-links](http://www.w3.org/TR/WAI-WEBCONTENT/#tech-meaningful-links) and the techniques in [http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-meaningful-links](http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-meaningful-links))

2.14.1.1 (X)HTML tests

2.14.1.1.1 Test 13.1_HTML_01

This test is targeted to find a and area elements with the same title and text with different different link target (href). If no title attribute is provided, only the element text is checked.

- Applicability criteria: a and area elements with same link text and title attributes, but different link targets.

```html
//a
//area
```

- Test procedure:
1. Select elements pairwise.
2. Check that two links with same link text (including alternative text for non-text elements) and same title attribute (if provided) point to the same resource.

- Expected results: PASS if #2 is true. FAIL if #2 is false.
- Fully automatable: yes

2.14.1.1.2  Test 13.1_HTML_02

This test is targeted to find a and area elements with link text that does not clearly identify the target of the link.

- Applicability criteria: a and area elements with same link text and title attributes, but different link targets.

```
//a
//area
```

- Test procedure:
  1. Check that the link text (including alternative text for non-text elements) clearly describes the effect of activating the element.

- Expected results: PASS if true. FAIL if false.
- Fully automatable: no

2.14.2  Checkpoint 13.2

*Provide metadata to add semantic information to pages and sites. [Priority 2]*

*(See [http://www.w3.org/TR/WAI-WEBCONTENT/#tech-use-metadata](http://www.w3.org/TR/WAI-WEBCONTENT/#tech-use-metadata) and the techniques in [http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-use-metadata](http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-use-metadata))*

2.14.2.1  (X)HTML tests

2.14.2.1.1  Test 13.2.HTML_01

This test is targeted to find a document title element in a web resource.

- Applicability criteria: documents.

```
```

- Test procedure:
  1. Check that there is a title element.
  2. Check that the title is not empty.
3. Check that the title element contains text that identifies the purpose of the resource.

   • Expected results: PASS if #1 and #2 and #3 are true. FAIL if #1 or #2 or #3 are false.
   • Fully automatable: no

2.14.3 Checkpoint 13.3

*Provide information about the general layout of a site (e.g., a site map or table of contents). [Priority 2]*

*In describing site layout, highlight and explain available accessibility features.*

*(See [http://www.w3.org/TR/WAI-WEBCONTENT/#tech-site-description](http://www.w3.org/TR/WAI-WEBCONTENT/#tech-site-description) and the techniques in [http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-site-description](http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-site-description))*

2.14.3.1 (X)HTML tests

2.14.3.1.1 Test 13.3_HTML_01

This test is targeted to find a web site without information about the general layout of the site.

   • Applicability criteria: the whole web site.
   • Test procedure:
     1. Check that the web site contains a document containing a description of the general layout of a site. A site map or a table of contents would meet this requirement.
     2. Check that this document can be reached from the index page.
   • Expected results: PASS if #1 and #2 true. FAIL if #1 or #2 is false.
   • Fully automatable: no.

2.14.4 Checkpoint 13.4

*Use navigation mechanisms in a consistent manner. [Priority 2]*

*(See [http://www.w3.org/TR/WAI-WEBCONTENT/#tech-clear-nav-mechanism](http://www.w3.org/TR/WAI-WEBCONTENT/#tech-clear-nav-mechanism) and the techniques in [http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-clear-nav-mechanism](http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-clear-nav-mechanism))*
2.14.4.1 (X)HTML tests

2.14.4.1.1 Test 13.4.HTML_01

This test is targeted to check whether navigation mechanisms are used in a consistent manner.

- Applicability criteria: navigation menus and navigation bars.

- Test procedure:
  1. Select navigation elements on several pages.
  2. Check that the navigation facilities are similar with respect to presentation (location in rendered document, location in content rendered without style sheets, colours, font) and behaviour.

- Expected results: PASS if #2 is true. FAIL if #2 is false.

- Fully automatable: no.

2.15 Guideline 14

Ensure that documents are clear and simple.

(See http://www.w3.org/TR/WAI-WEBCONTENT/#gl-facilitate-comprehension)

This guideline provides information on how to create clear and simple documents.

2.15.1 Checkpoint 14.1

Use the clearest and simplest language appropriate for a site's content. [Priority 1]

(See http://www.w3.org/TR/WAI-WEBCONTENT/#tech-simple-and-straightforward and the techniques in http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#tech-icons)

2.15.1.1 (X)HTML tests

2.15.1.1.1 Test 14.1.HTML_01

This test is targeted to analyse the readability of the content\(^\text{45}\).

- Applicability criteria: whole document.

\(^{45}\text{This test has only been investigated for English, French, German, Italian, Hungarian, and Dutch. Applicability for other languages requires further investigation.}\)
Test procedure:
1. For each instance of jargon, check that it is part of the language of the context in which it is used. Note: this also applies to abbreviated forms used as jargon.
2. If the resource is intended for the general public, check for each word that it is in common usage. Note: this does not require that only words with a native origin are used; frequency of usage is more important than word origin. Check dictionaries for basic vocabulary in case of doubt.
3. If the resource is intended for the general public, check that no slang, jargon or specialized meanings of familiar words are used, unless they are defined within the same page or link to a glossary entry where the meaning is explained.
4. If the resource is intended for people who work in a particular technical field, check that each instance of jargon is used according to its accepted dictionary meaning.
5. Graphical metaphors and symbols are explained.
6. For each sentence, check that it does not use more than one level of subordination and that it does not contain more than two subordinate clauses.
7. Check that each pronoun reference and each reference to an earlier point in the document is unambiguous.
8. Check that each paragraph covers only one single topic or subtopic.

46 One can check "common usage" by consulting basic dictionaries such as the following:
For British English:
* Oxford Children's Dictionary, ed Robert Allen, 2003 (with 20,000 headwords);
* Cambridge Essential English Dictionary (with definitions for 12,000 words and phrases).
For American English:
* Oxford Elementary Learner's Dictionary (15,000 references);
* Oxford Essential Dictionary (19,000 words and phrases).
For French:
* Larousse Junior - 7/11 ans - CE/CM (20,000 words);
* Le Robert Junior - 8-11 ans (20,000 words).
For German:
* Duden-Hueber - Wörterbuch Deutsch als Fremdsprache (11,000 entries).
For Italian:
* DIB - Dizionario Italiano di base, ed. Tullio De Mauro, Paravia 2000.
* DIB - Dizionario Italiano di base della lingua italiana con Dizionario visuale, ed. Tullio De Mauro & Gian Giuseppe Moroni, Paravia.
* PRIMO Dizionario Italiano. Garzanti, 2004 (over 15,000 entries).
* Il mio primo dizionario - Nuovo MIOT. Giunti, 2001 (20,000 entries).
For Hungarian:
* Helyesírási diákszótár (Kugler Nóra, Lengyel Klára, Markó Alexandra, Mártonfi Attila; Korona Kiadó Kft; 16,000 entries).
For Dutch:
All of the dictionaries listed here contain 20,000 entries for words and phrases or less.

47 A subordinate clause is a clause that is embedded in a sentence and that cannot stand alone as a complete sentence and that functions within the sentence as a noun or adjective or adverb. A subordinate clause is introduced by a subordinating conjunction (such as 'because', 'if', 'although', 'unless' and 'after' in English) or by a relative pronoun (such as 'who', 'whose', 'that', 'why' in English). Further details and examples can be found in grammars and at http://en.wikipedia.org/wiki/Subordinate_clause.
9. Check that there are no spelling errors that cause deviating pronunciation in a speech synthesizer, except where deviating spelling is explicitly intended for literary or educational purposes.

• Expected results: PASS if #1 to #9 is true. FAIL if #1 or #2 or #3 or #4 or #5 or #6 or #7 or #8 or #9 is false.

• Fully automatable: no.
3 References


4 Appendix A: Document License

Copyright © 2005—2007 European Commission and WAB Cluster members

[License based upon the W3C document license48. NOTICE: This license applies specifically to UWEM and WAB Cluster materials. None of the documents from the World Wide Web Consortium or its Web Accessibility Initiative that are referenced or quoted in this document are subject to the conditions of this license. Materials from the World Wide Web Consortium or the Web Accessibility Initiative that are referenced or quoted in this document or in documents derived from it, are subject to the W3C© Document License.]

By using and/or copying this document, or the document from which this statement is linked (Unified web Evaluation Methodology, UWEM), you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to copy, and distribute the contents of this document, or the document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

• A link or URL to the original document.
• The pre-existing copyright notice of the original author, or if it doesn't exist, a notice (hypertext is preferred, but a textual representation is permitted) of the form: "Copyright © 2005 European Commission and WAB Cluster members. All Rights Reserved."
• The STATUS of the document.

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

This license allows the use, modification and extension of this document to

48 http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231. Our license allows extensions and modifications of UWEM, as far as references to the original document are given, and copies of this license are provided.
any organisation royalty free, with the conditions expressed above. In case of modifications outside the selected standardisation body or equivalent entity by the copyright holders, neither the term “Unified Web (Site) Evaluation Methodology” nor the acronym “UWEM” could be used to denominate the resulting work.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.
5 Appendix B: W3C® DOCUMENT LICENSE

http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231

Public documents on the W3C site are provided by the copyright holders under the following license. By using and/or copying this document, or the W3C document from which this statement is linked, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to copy, and distribute the contents of this document, or the W3C document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original W3C document.

2. The pre-existing copyright notice of the original author, or if it doesn't exist, a notice (hypertext is preferred, but a textual representation is permitted) of the form: "Copyright © [date-of-document] World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved. http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231"

3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of W3C documents is granted pursuant to this license. However, if additional requirements (documented in the Copyright FAQ) are satisfied, the right to create modifications or derivatives is sometimes granted by the W3C to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE
NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

This formulation of W3C’s notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, moves information on style sheets, DTDs, and schemas to the Copyright FAQ, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, such as the translating or annotating specifications. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

Last revised $Id: copyright-documents-20021231.html,v 1.6 2004/07/06 16:02:49 slesch Exp $