

# Applicazione e verifica dei requisiti per siti INTERNET

Analisi dei ventidue requisiti per l'accessibilità delle tecnologie web-based

*Questo documento contiene le motivazioni, le modalità di implementazione e di verifica dei requisiti per i siti INTERNET di cui all'allegato A del DM 8 luglio 2005. Tale documento è un estratto dal libro "Legge 04/2004 dalla teoria alla realtà" di R. Scano (Edizioni IWA ITALY) ed alla stesura di questo capitolo hanno collaborato i soci IWA/HWG Marco Bertoni, Roberto Ellero, Luca Mascaro, Livio Mondini.*

Questo lavoro viene rilasciato con licenza Creative Commons Attribuzione-Non commerciale-Non opere derivate 2.5 Italia License



# Elenco dei requisiti

<b>Requisito 1</b>	<b>11</b>
<i>Motivazione</i>	11
<i>Implementazione</i>	16
<i>Riferimenti WCAG 1.0</i>	19
<i>Prima applicazione</i>	32
<i>Verifica del requisito</i>	33
<b>Requisito 2</b>	<b>41</b>
<i>Motivazione</i>	41
<i>Implementazione</i>	42
<i>Riferimenti WCAG 1.0</i>	43
<i>Prima applicazione</i>	44
<i>Verifica del requisito</i>	45
<b>Requisito 3</b>	<b>46</b>
<i>Motivazione</i>	46
<i>Implementazione</i>	47
<i>Riferimenti WCAG 1.0</i>	47
<i>Verifica del requisito</i>	56

<b>Requisito 4</b>	<b>58</b>
<i>Motivazione</i>	58
<i>Implementazione</i>	58
<i>Riferimenti WCAG 1.0</i>	58
<i>Verifica del requisito</i>	60
<b>Requisito 5</b>	<b>62</b>
<i>Motivazione</i>	62
<i>Implementazione</i>	62
<i>Riferimenti WCAG 1.0</i>	63
<i>Verifica del requisito</i>	65
<b>Requisito 6</b>	<b>67</b>
<i>Motivazione</i>	67
<i>Implementazione</i>	68
<i>Riferimenti WCAG 1.0</i>	69
<i>Verifica del requisito</i>	71
<b>Requisito 7</b>	<b>73</b>
<i>Motivazione</i>	73
<i>Implementazione</i>	73
<i>Riferimenti WCAG 1.0</i>	74
<i>Verifica del requisito</i>	75

<b>Requisito 8</b>	<b>76</b>
<i>Motivazione</i>	76
<i>Implementazione</i>	76
<i>Riferimenti WCAG 1.0</i>	77
<i>Verifica del requisito</i>	78
<b>Requisito 9</b>	<b>79</b>
<i>Motivazione</i>	79
<i>Implementazione</i>	79
<i>Riferimenti WCAG 1.0</i>	80
<i>Verifica del requisito</i>	84
<b>Requisito 10</b>	<b>85</b>
<i>Motivazione</i>	85
<i>Implementazione</i>	85
<i>Riferimenti WCAG 1.0</i>	85
<i>Verifica del requisito</i>	88
<b>Requisito 11</b>	<b>89</b>
<i>Motivazione</i>	89
<i>Implementazione</i>	89
<i>Riferimenti WCAG 1.0</i>	89
<i>Verifica del requisito</i>	93

<b>Requisito 12</b>	<b>94</b>
<i>Motivazione</i>	94
<i>Implementazione</i>	94
<i>Riferimenti WCAG 1.0</i>	95
<i>Verifica del requisito</i>	97
<b>Requisito 13</b>	<b>99</b>
<i>Motivazione</i>	99
<i>Implementazione</i>	99
<i>Riferimenti WCAG 1.0</i>	99
<i>Verifica del requisito</i>	101
<b>Requisito 14</b>	<b>103</b>
<i>Motivazione</i>	103
<i>Implementazione</i>	103
<i>Riferimenti WCAG 1.0</i>	103
<i>Verifica del requisito</i>	107
<b>Requisito 15</b>	<b>108</b>
<i>Motivazione</i>	108
<i>Implementazione</i>	108
<i>Riferimenti WCAG 1.0</i>	109
<i>Verifica del requisito</i>	110

<b>Requisito 16</b>	<b>111</b>
<i>Motivazione</i>	111
<i>Implementazione</i>	111
<i>Riferimenti WCAG 1.0</i>	112
<i>Verifica del requisito</i>	114
<b>Requisito 17</b>	<b>116</b>
<i>Motivazione</i>	116
<i>Implementazione</i>	116
<i>Riferimenti WCAG 1.0</i>	117
<i>Verifica del requisito</i>	118
<b>Requisito 18</b>	<b>120</b>
<i>Motivazione</i>	120
<i>Implementazione</i>	120
<i>Riferimenti WCAG 1.0</i>	121
<i>Verifica del requisito</i>	138
<b>Requisito 19</b>	<b>141</b>
<i>Motivazione</i>	141
<i>Implementazione</i>	141
<i>Riferimenti WCAG 1.0</i>	141
<i>Verifica del requisito</i>	144

<b>Requisito 20</b>	<b>146</b>
<i>Motivazione</i>	<i>146</i>
<i>Implementazione</i>	<i>146</i>
<i>Riferimenti WCAG 1.0</i>	<i>146</i>
<i>Verifica del requisito</i>	<i>149</i>
<b>Requisito 21</b>	<b>151</b>
<i>Motivazione</i>	<i>151</i>
<i>Implementazione</i>	<i>151</i>
<i>Verifica del requisito</i>	<i>155</i>
<b>Requisito 22</b>	<b>157</b>
<i>Motivazione</i>	<i>157</i>
<i>Implementazione</i>	<i>157</i>
<i>Riferimenti WCAG 1.0</i>	<i>157</i>
<i>Verifica del requisito</i>	<i>160</i>

Nel decreto per sito INTERNET si intende l'insieme di componenti (pagine, link, funzioni) che risiedono su uno o più computer collegati alla rete Internet e che è utilizzato per veicolare informazione e/o erogare servizi.

I requisiti tecnici si applicano ai contenuti della finestra del programma che ne consente la visualizzazione (browser) sul personal computer utilizzato dall'utente, sia nella loro parte tecnologica, come ad esempio il linguaggio a marcatori utilizzato, gli oggetti presenti in una pagina scritti in un linguaggio di programmazione, sia nella loro parte redazionale, come ad esempio le modalità grafiche di presentazione del contenuto informativo (immagini, colori, oggetti multimediali).

La rispondenza a questi requisiti costituisce il livello minimo obbligatorio di accessibilità per i siti INTERNET, sia quando con questi si forniscono informazioni sia quando tramite questi si erogano servizi in qualsiasi area pubblica o privata come vedremo di seguito nel capitolo.

I requisiti tecnici si applicano anche a tutti i casi in cui i soggetti di cui all'Art. 3 comma 1 della Legge 4/2004 veicolano informazioni o erogano servizi mediante tecnologie Internet sia per mezzo di sistemi informatici (i normali personal computer e server) che fanno parte di reti Intranet o per mezzo di supporti utilizzabili anche quando il personal computer dell'utente non è collegato ad una rete, quali, ad esempio CD-ROM, DVD.

La conformità ai 22 requisiti da sottoporre a verifica tecnica è elemento preliminare per poter procedere alla verifica soggettiva.

Quelli che seguono sono i 22 requisiti da sottoporre a verifica tecnica. Per ciascun requisito viene indicato:

- il numero d'ordine;
- l'enunciato;
- il riferimento ai punti di controllo delle Web Content Accessibility Guidelines - versione 1.0 (WCAG 1.0)  
<<http://www.w3.org/TR/WCAG10/>> del W3C-WAI;
- il riferimento agli standard definiti nel paragrafo 1194.22 della Section 508 del Rehabilitation Act.

Sono stati indicati, quando esistenti, i riferimenti ai punti di controllo delle WCAG 1.0 e agli standard del paragrafo 1194.22 della Section 508. Secondo il decreto tali riferimenti non vanno intesi come perfette corrispondenze, ma solo come analogie o vicinanze per consentire un più facile riscontro con gli standard esistenti a coloro che hanno già applicato tali standard e per facilitare l'utilizzo



degli strumenti informatici di valutazione della accessibilità oggi disponibili sul mercato. Questo significa, come vedremo nell'analisi dei singoli requisiti, che l'enunciato di fatto è completato dai riferimenti ma che non sempre tali riferimenti sono da considerare totalmente aderenti (vedasi, ad esempio, il requisito 1 con riferimento all'utilizzo della DTD Strict rispetto al punto di controllo 3.2 delle WCAG 1.0). Nell'enunciato di alcuni requisiti appare la dizione: In sede di prima applicazione: con questa formula il Gruppo di lavoro ha inteso considerare l'impatto che una applicazione immediata dei requisiti può avere sul panorama attuale dei siti Web pubblici con oggettive difficoltà operative di applicazione del requisito nell'enunciato generale. Per taluni requisiti è stato perciò indicato un possibile percorso di adeguamento. Nel documento per ogni singolo requisito vengono fornite le motivazioni per la sua applicazione, come implementarlo all'interno dei contenuti e come controllarne la corretta applicazione.

# Strumenti di supporto alla verifica

Il tecnico che effettuerà la verifica del sito INTERNET dovrà applicare quanto previsto dall'art. 5 comma 2 tra cui la verifica di conformità del codice per ogni pagina del sito. Si consiglia pertanto di dotarsi di strumenti che consentano l'esecuzione di validazione in modalità "batch", come il prodotto CSE HTML Validator <<http://www.htmlvalidator.com>> che consente di identificare gli errori rispetto al linguaggio di marcatura dichiarato.

Come strumento base per le valutazioni delle singole pagine consigliamo l'uso della Barra dell'accessibilità <<http://www.webaccessibile.org/wat>>, sviluppata con la partecipazione attiva di alcuni associati di IWA/HWG allo scopo di uno strumento di valutazione adatto alla legge 4/2004. A partire dalla versione 1.2 la barra prevede anche un utile strumento per la valutazione della luminosità e della differenza di colore integrato nella stessa.

Infine, per verificare il risultato della pagina su diversi browser e su diversi sistemi operativi, è necessario che l'utente si doti di almeno due sistemi operativi differenti con diversi browser, oppure che utilizzi dei servizi di supporto alla valutazione quali, ad esempio <<http://www.browsercam.com>>.

## Requisito I

Enunciato: Realizzare le pagine e gli oggetti al loro interno utilizzando tecnologie definite da grammatiche formali pubblicate nelle versioni più recenti disponibili quando sono supportate dai programmi utente. Utilizzare elementi ed attributi in modo conforme alle specifiche, rispettandone l'aspetto semantico. In particolare, per i linguaggi a marcatori HTML (Hypertext Markup Language) e XHTML (eXtensible Hypertext Markup Language):

- a) per tutti i siti di nuova realizzazione utilizzare almeno la versione 4.01 dell'HTML o preferibilmente la versione 1.0 dell'XHTML, in ogni caso con DTD (Document Type Definition - Definizione del Tipo di Documento) di tipo Strict;
- b) per i siti esistenti, in sede di prima applicazione, nel caso in cui non sia possibile ottemperare al punto a) è consentito utilizzare la versione dei linguaggi sopra indicati con DTD Transitional, ma con le seguenti avvertenze:

- evitare di utilizzare, all'interno del linguaggio a marcatori con il quale la pagina è realizzata, elementi ed attributi per definirne le caratteristiche di presentazione della pagina (per esempio, caratteristiche dei caratteri del testo, colori del testo stesso e dello sfondo, ecc.), ricorrendo invece ai Fogli di Stile CSS (Cascading Style Sheets) per ottenere lo stesso effetto grafico;
- evitare la generazione di nuove finestre; ove ciò non fosse possibile, avvisare esplicitamente l'utente del cambiamento del focus;
- pianificare la transizione dell'intero sito alla versione con DTD Strict del linguaggio utilizzato, dandone comunicazione alla Presidenza del Consiglio dei Ministri - Dipartimento per l'innovazione e le tecnologie e al Centro nazionale per l'informatica nella pubblica amministrazione.

Riferimenti WCAG 1.0: 3.1, 3.2, 3.5, 3.6, 3.7, 11.1, 11.2

Riferimenti Section 508: Non presente

## Motivazione

---

L'enunciato numero 1 richiede allo sviluppatore di utilizzare grammatiche formali pubblicate, utilizzando le versioni più recenti se supportate dai programmi utente. E' bene tener presente che nella creazione di contenuti web (pagine HTML,

espressioni matematiche, animazioni, ecc.) si utilizzano dei linguaggi e che tali linguaggi utilizzano delle "parole" e delle "regole" per il loro utilizzo.

Una grammatica formale, in informatica, è una specifica o una raccomandazione tecnica definita in linguaggio naturale (documentazione) oppure in un linguaggio di definizione come una DTD SGML o un XML Schema.

La conformità varia a dipendenza del tipo di definizione: se per esempio prendiamo una normale pagina web che è definita in HTML o XHTML, la sua definizione è un file DTD.

Questo file DTD ci fornisce ufficialmente le seguenti informazioni:

- qual è l'elemento di partenza del documento (HTML);
- quali elementi esistono;
- qual è lo scopo di ogni elemento e quando deve essere usato;
- quali elementi possono essere annidati sotto quali elementi, nonché quante volte possono occorrere;
- quale tipo di contenuto e in che forma può essere inserito all'interno degli elementi;
- quali attributi esistono;
- qual è lo scopo di ogni attributo e quando deve essere usato;
- quali attributi possono essere inseriti in quali elementi;
- quali valori e che forma possono prendere i vari attributi.

Alcuni punti, come lo scopo di ogni elemento o attributo, viene riaffermato esplicitamente nella frase conclusiva della regola tecnica.

In caso adottassimo un linguaggio formalizzato in XML Schema, per esempio RSS, avremmo molte informazioni in più, come i limiti di occorrenza oppure eventuali spazi logici dei nomi.

Infine, se adottassimo una tecnologia definita descritta solo in linguaggio naturale dovremmo semplicemente seguire le direttive in essa definite. Questo vale per esempio per le specifiche di accessibilità dei documenti Adobe Acrobat o Microsoft Word.

La creazione di contenuti web che non rispettano le grammatiche formali pubbliche possono causare dei problemi di accesso ai contenuti da parte di utenti che utilizzano browser web o tecnologie assistive che non riescono ad identificare elementi o attributi non standard.

L'autore del contenuto web a questo punto deve dichiarare il tipo di linguaggio utilizzato utilizzando quindi le specifiche grammatiche formali per specifici contenuti. Non rispettare le grammatiche formali può causare problemi di visualizzazione dei contenuti in diversi browser, anche se spesso questi problemi sono per lo più dovuti alla diversa interpretazione dei fogli di stile: i browser moderni non interrompono il caricamento della pagina anche se sono presenti errori di marcatura e cercano di renderizzare il contenuto correggendo gli errori autonomamente: ciò avviene solamente se il contenuto della pagina è fornito come `text/html`, mentre se il testo è fornito come `application/xhtml+xml` i browser come Mozilla Firefox interromperanno l'esecuzione del caricamento della pagina evidenziando l'errore nel codice di marcatura. In futuro comunque tutti i browser passeranno a questa renderizzazione controllata e sarà quindi necessario sviluppare codice in conformità alla specifica.

Le tecnologie definite da grammatiche formali cui si fa riferimento nell'enunciato sono quelle pubblicate da:

- International Organization for Standardization (ISO);
- World Wide Web Consortium (W3C);
- `xml:lang="en" lang="en">`ECMA, ovvero European Computer Manufacturers Association. Dal 1994 il nome ufficiale è Ecma International - European association for standardizing information and communication systems.

Esempi di tali tecnologie consentite dal requisito sono:

**Tabella 1 Tecnologie consentite**

Linguaggio	Ente	Annotazioni
CSS Level 1	W3C	Cascade Style Sheets (CSS). È il risultato dei lavori sui fogli di stile di Håkon Wium Lie e Bert Bos.
CSS Level 2	W3C	Cascade Style Sheets (CSS). Sono un superset del Level 1, al quale aggiungono proprietà e del quale correggono alcuni errori. Attualmente nessun Browser li supporta completamente.
<code>xml:lang="en" lang="en"&gt;</code> ECMA-262	ECMA	European Computer Manufacturers Association (ECMA). È una specifica relativa al linguaggio JavaScript standard.

HTML 4.01	W3C	Hyper Text Markup Language (HTML). Correzioni secondarie a HTML4.0 – In assoluto l'ultima versione pubblicata di HTML.
MathML	W3C	Mathematical Markup Language, (MathML). È una applicazione XML per descrivere la notazione matematica rendendone struttura e contenuto.  È integrabile in XHTML e in SVG.
P3P	W3C	Platform for Privacy Preferences Project (P3P). Sviluppato per consentire agli utenti internet di esercitare in modo automatizzato un maggiore controllo sull'utilizzo delle informazioni personali da parte dei siti che visitano.
PICS	W3C	Platform for Internet Content Selection (PICS). Tecnologia utilizzata per associare a una pagina oa un sito Web etichette che ne descrivono il contenuto. Le etichette PICS possono essere utilizzate da un browser opportunamente configurato per inibire l'accesso a determinate tipologie di documenti in rete (ad esempio, siti pornografici).
RDF	W3C	Resource Description Framework (RDF). Sintassi astratta che consente di associare informazioni descrittive (es. autore, titolo, soggetto ecc.) a risorse e documenti pubblicati sul Web (o metadati), in modo che siano utilizzabili direttamente da applicazioni.
SMIL	W3C	Synchronized Multimedia Integration Language (SMIL). È un linguaggio basato su XML che consente di scrivere presentazioni multimediali interattive. È integrabile in XHTML e in SVG.

SVG	W3C	Scalable Vector Graphics (SVG). È un linguaggio derivato dall'XML, cioè di un'applicazione del metalinguaggio posto a base degli sviluppi del Web da parte del consorzio W3C, che si pone l'obiettivo di descrivere figure bidimensionali statiche e animate.
W3C DOM (Level 1, 2 e 3)	W3C	Document Object Model (DOM). È una interfaccia API indipendente dal linguaggio e dalla piattaforma che permette a programmi e a script di accedere in modo dinamico e di aggiornare il contenuto, la struttura e lo stile di documenti.
XHTML 1.0	W3C	Extensible Hyper Text Markup Language (XHTML). Riformulazione di HTML 4.01 in XML. Disponibile in tre DTD: Transitional, Strict e Frameset. Consente l'inclusione di gruppi di elementi specifici per la grafica (SVG), la multimedialità (SMIL), la matematica (MathML), l'e-commerce, ecc.
XHTML 1.1	W3C	Extensible Hyper Text Markup Language (XHTML). È il primo vero linguaggio di programmazione a marcatura interamente basato sulla logica dei Moduli. Esiste solo nella DTD Strict, molto simile (ma più rigida) alla DTD Strict di XHTML 1.0.
XML	W3C	Extensible Markup Language (XML). È un linguaggio destinato a facilitare la definizione, la validazione e la condivisione di differenti formati di documenti sul Web. La versione 1.1 una formalizzazione di alcuni elementi della specifica XML 1.0 che altrimenti provocano errori nei parser.
XSL	W3C	Extensible Stylesheet Language (XSL).  È un linguaggio che permette di descrivere come devono apparire i file codificati in formato XML.

## Implementazione

---

Di seguito si riportano alcune delle grammatiche formali create dal W3C, il cui elenco completo è disponibile sul sito del consorzio  
<<http://www.w3.org/QA/2002/04/Valid-DTD-list.html>>.

Le seguenti DTD riguardano le definizioni per le pagine web.

### ***HTML 4.01 Transitional***

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

### ***HTML 4.01 Frameset***

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

### ***HTML 4.01 Strict***

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

### ***XHTML 1.0 Transitional***

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

### ***XHTML 1.0 Frameset***

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

### ***XHTML 1.0 Strict***

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```



## XHTML 1.1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

Come si può notare, HTML e XHTML sino alla versione 1.0 descrivono tre diversi tipi di documento:

- Transitional: mantiene la compatibilità con le precedenti versioni, consentendo l'utilizzo di elementi ed attributi di presentazione all'interno del contenuto.
- Frameset: definisce le grammatiche per l'utilizzo di frame.
- Strict: è la versione "rigorosa", nella quale sono eliminate tutte le impostazioni di formattazione che possono essere gestite tramite i fogli di stile.

La versione XHTML 1.1 prevede una sola DTD, molto simile alla DTD XHTML 1.0 Strict, e presenta alcuni problemi di retrocompatibilità con i vecchi browser e con i dispositivi mobile (PocketPC e palmari).

Se, per esempio, si dichiara che un documento è conforme alla DTD Strict di HTML 4.01, non sarà possibile usare nel corpo del documento elementi di presentazione come `<FONT>` e attributi come `SIZE` e `FACE`, specificamente definiti con finalità di presentazione, perché non sono compresi nella DTD Strict di HTML 4.01 e l'usarli significa produrre un documento formalmente non valido.

Per rispettare il requisito, tutte le pagine web devono rispettare la grammatica formale dichiarata dall'autore: questo significa che ogni pagina deve:

- dichiarare il tipo di documento (tramite l'elemento `<DOCTYPE>`);
- dichiarare la codifica di carattere utilizzata (esempio: `utf-8`);
- utilizzare elementi ed attributi secondo quanto definito dalle specifiche (ad esempio, rispettare i livelli di titolazione, usare correttamente le liste, utilizzare acronimi ed abbreviazioni in modo corretto, ecc.).

Quanto sopra elencato consentirà al programma utente utilizzato per visualizzare la pagina di valutare la semantica da utilizzare.

Poiché il codice più accessibile è quello che tiene separato il contenuto del documento dalla presentazione, le DTD più indicate per ottenere pagine accessibili sono quelle rigorose (strict), che escludono l'uso di elementi e attributi di presentazione: la DTD Strict di HTML 4.01 e la DTD Strict di XHTML 1.0 che di

fatto sono richieste dal requisito 1 per tutti i nuovi siti. Il fatto che XHTML sia una riformulazione di HTML nella sintassi più rigorosa di XML rende preferibile (per possibili vantaggi in proiezione futura) l'uso della DTD Strict di XHTML 1.0 rispetto a quella Strict di HTML 4.01.

Un esempio di corretto utilizzo del linguaggio di marcatura con la DTD XHTML 1.0 Strict fornito come `application/xhtml+xml` è il seguente.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it" lang="it">
<head>
<title>Titolo del documento</title>
<meta http-equiv="Content-Type"
content="application/xhtml+xml; charset=utf-8" />
<meta name="language" content="it" />
</head>
<body>
...
</body>
</html>
```

Lo stesso esempio fornito come `text/html` (ovvero compatibile con Microsoft Internet Explorer e versioni precedenti degli altri browser) è il seguente:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it" lang="it">
<head>
<title>Titolo del documento</title>
<meta http-equiv="Content-Type"
content="text/html; charset=utf-8" />
<meta name="language" content="it" />
</head>
<body>

</body>
</html>
```

Come si noterà, nel primo caso si utilizza la prefazione XML ("XML Prolog"), la definizione di `content-type application/xhtml+xml` e la definizione della lingua predefinita esclusivamente con `xml:lang="it"` mentre nel secondo caso la prefazione XML non è utilizzata, il `content-type` è `text/html` e la definizione del

tipo di lingua è segnalata sia come `xml:lang="it"` sia come `lang="it"`, in quanto nel primo caso il contenuto è basato su XML mentre nel secondo riporta una compatibilità con html, ereditandone gli attributi.

È comunque consigliato utilizzare la prefazione XML per qualsiasi documento XHTML in quanto di fatto è un documento basato su XML.

Come vedremo negli esempi delle WCAG il rispetto del requisito 1 non è solamente legato alla verifica tramite W3C Markup validator del contenuto della pagina e del foglio di stile ma richiede una verifica da parte dell'esperto del corretto utilizzo degli elementi ed attributi nonché la verifica di eventuali "trucchi" utilizzabili dagli sviluppatori per aggirare la valutazione.

## Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

### **Punto di controllo 3.1 (Priorità 2)**

#### **Quando esiste un linguaggio di marcatura adatto, per veicolare informazione usare un marcatore piuttosto che le immagini**

Le equazioni matematiche/scientifiche, i simboli e le note musicali in una pagina Web spesso vengono rappresentate con immagini. Un'equazione rappresentata sotto forma di immagine è di fatto inaccessibile, in quanto anche un'alternativa testuale non consentirà di comprenderne chiaramente il contenuto.

Alcuni gruppi di lavoro del W3C stanno definendo una serie di linguaggi di marcatura (tra cui il MathML <<http://www.w3.org/Math/>>) che consentono di creare le formule in modo semplice. Esistono diversi linguaggi di marcatura e diverse tecnologie che consentono di leggere questi valori. Utilizzando XML le formule sono di facile interpretazione sia per le tecnologie assistive che per le applicazioni matematiche. Un editor che consente di creare funzioni matematiche è Amaya del W3C. Per esempio, la seguente formula matematica potrà essere inserita in una pagina Web in MathML con il codice seguente:

$$\int_0^t \frac{dx}{x}$$

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
<mrow>
<msubsup>
<mo>&int;</mo>
<mn>0</mn>
<mi>t</mi>
</msubsup>
<mfrac>
<mrow>
<mo>d</mo>
<mi>x</mi>
</mrow>
<mi>x</mi>
</mfrac>
</mrow>
</math>
```

Per visualizzare contenuti in MathML è comunque necessaria la presenza di plug-in, disponibili per i browser più diffusi.

Per quanto riguarda la validazione della grammatica del linguaggio di marcatura MathML, è possibile utilizzare il W3C Markup Validator, che dal 2002 integra la possibilità di controllare anche questa grammatica formale. Maggiori informazioni sono disponibili nel sito del W3C dedicato alla matematica.

## Punto di controllo 3.2 (Priorità 2)

### Creare documenti che rispettino le grammatiche formali

Abbiamo già visto nelle motivazioni dell'applicazione del requisito come sia necessario rispettare la DTD dichiarata. Michele Diodati

<http://www.diodati.org/> in un suo scritto ha spiegato chiaramente com'è composta una DTD e cosa significando quindi quelle sigle e quei riferimenti contenuti nell'elemento DocType. Prendiamo ad esempio il DocType di XHTML 1.0 Strict:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

la DTD è composta dai seguenti valori:

- TopElement: html. Il tipo di documento SGML.

- Availability: PUBLIC. Dichiarata se il tipo di documento referenziato è pubblico o meno (il valore alternativo è SYSTEM).
- Formal public identifier: "-//W3C//DTD XHTML 1.0 Strict//EN".
- Registration: -. Indica se l'organizzazione che ha definito la DTD è registrata all'ISO o meno (il W3C non lo è). Il valore alternativo è +.
- Organization: W3C. L'organizzazione che ha creato e mantiene la DTD.
- Type: DTD. Specifica a cosa si riferisce la dichiarazione (una DTD, nel nostro caso).
- Label: XHTML 1.0. Il nome descrittivo della DTD referenziata, che può contenere la versione del documento.
- Definition: Strict. Il tipo di DTD, se ne esiste più di un tipo per il linguaggio referenziato.
- Language: EN. La lingua in cui è scritta la DTD.
- URI: "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd". L'indirizzo dove fisicamente risiede il documento referenziato.

L'URI indica un documento che contiene i riferimenti al corretto utilizzo degli elementi e degli attributi spiegati nella specifica: un uso non corretto di questi attributi, come ad esempio utilizzare l'elemento <b> per effettuare decorazioni grafiche tramite fogli di stile anziché utilizzarlo per la definizione del grassetto per un testo, significa violazione del primo requisito. Inoltre il requisito richiede il rispetto anche della conformità dei fogli di stile che, come vedremo, è verificabile tramite l'applicazione CSS Validator del W3C.

Il primo requisito richiede di utilizzare, per i nuovi siti, la DTD Strict per HTML 4.01 oppure XHTML 1.0: questo comporta l'indisponibilità, per esempio, dell'attributo target per i collegamenti esterni. È possibile quindi utilizzare codice javascript conforme ai requisiti in modo da garantire l'apertura della nuova finestra informando allo stesso tempo l'utente di tale funzionalità (Requisito 19).

Una soluzione ottimale ottenibile tramite DOM Injection è la seguente:

```
<a href="destinazione.html" class="targetblank"
title="Descrizione">Prova</a>
```

Utilizzando la funzione seguente e richiamandola all'interno della pagina web con un collegamento ad un documento Javascript, si otterrà a livello di DOM un risultato come il seguente:

```
<a href="destinazione.html" class="targetblank"
title="Descrizione (collegamento in nuova finestra)"
```

```
onclick="window.open(this.href); return false;"
onkeypress="window.open(this.href); return false;">Prova</a>
```

In questo modo si potrà quindi aprire la nuova finestra (e l'utente ne sarà informato) esclusivamente se Javascript è attivo, altrimenti l'utente potrà comunque accedere al contenuto nella stessa finestra. Lo script che consente tale funzionalità è riportato qui di seguito.

```
function TargetBlank() {
if(document.getElementsByTagName){
var msg = " (collegamento in nuova finestra)";
var links = document.getElementsByTagName("a");
for (i = 0; i < links.length; i++) {
var link = links[i];
if (link.className.indexOf("targetblank") != -1)
{
link.title += msg;
var fn = function () {
window.open(this.href); return false; }
link.onclick = link.onkeypress = fn;
}
}
}
}
```

Si tiene comunque a precisare che l'apertura di nuove finestre andrebbe evitata in quanto non consente di evitarne l'apertura ad un utente dotato di tecnologia assistiva. Inoltre oggi la maggior parte dei browser contiene delle funzionalità per impedire l'apertura di nuove finestre. Se un utente desidera aprire una pagina in una nuova finestra, esistono delle funzionalità idonee all'interno del browser: evitiamo quindi di limitare la libertà dell'utente obbligandolo ad azioni non gradite o non consentite dal browser utilizzato.

## **Punto di controllo 3.5 (Priorità 2)**

### **Usare elementi di intestazione per veicolare la struttura del documento e usarli in modo conforme alle specifiche**

Una pagina Web può essere formata da brevi contenuti, ma può contenere anche testi molto articolati (per esempio, un bando di gara d'appalto, un comunicato stampa, ecc.) che per poter essere letti devono essere organizzati in modo da garantirne la leggibilità: come per i libri, anche sul Web servono titoli, sottotitoli, capitoli, sottocapitoli e così via. Gli elementi di titolazione (<h1> - <h6>) devono essere utilizzati in modo corretto per garantire una corretta visualizzazione e il

rispetto della semantica dei contenuti. Alcuni programmi di navigazione utilizzano gli elementi di titolazione per ottimizzare la navigazione della pagina da parte degli utenti. Utilizzando gli elementi <h> si garantirà maggiore accessibilità ai contenuti della pagina: tutte le pagine dovrebbero prevedere almeno un elemento <h1> e via via, a scalare di importanza, gli elementi <h2>, <h3> che forniscono all'utente la possibilità di individuare i paragrafi considerati importanti dall'autore dei contenuti. Da ciò si può chiaramente comprendere che è importante rispettare l'ordine di importanza dei titoli, in modo che un elemento <h2> segua un elemento <h1>, mentre un elemento <h3> deve seguire un elemento di tipo <h2>. È errato quindi saltare livelli, passando ad esempio da <h1> a <h3> senza evidenziare l'elemento <h2> e nella verifica del requisito vedremo come è possibile controllare il corretto utilizzo di tali elementi.

```
<h1>Titolo livello 1</h1>
<p>... testo di esempio ..... </p>
<h2>Titolo livello 2</h2>
<h3>Titolo livello 3</h3>
<p>... testo di esempio ..... </p>
<h2>Titolo livello 2</h2>
<p>... testo di esempio ..... </p>
<h3>Titolo livello 3</h3>
<p>... testo di esempio ..... </p>
<p>... testo di esempio ..... </p>
<h3>Titolo livello 3</h3>
<p>... testo di esempio ..... </p>
<h1>Titolo livello 1</h1>
<p>... testo di esempio ..... </p>
```

Va precisato inoltre che la titolazione deve avvenire in tutti i contesti (struttura pagina, contenuti, ecc.) ma soprattutto che può ripartire dall'elemento di titolazione principale <h1> per ogni contesto che sia racchiuso dentro un elemento di blocco con attributo "id" univoco.

Non bisogna mai utilizzare gli elementi di titolazione a puro scopo decorativo, come per ottenere un incremento nella dimensione del testo, poiché tale comportamento inibisce la comprensione della semantica del documento: l'utilizzo dei fogli di stile, anche in questo caso, è la miglior soluzione per definire le modifiche degli attributi grafici dei testi.

Per quanto riguarda invece le tecnologie assistive, i lettori dello schermo utilizzano gli elementi di titolazione per determinare la struttura del documento, e quindi la sua lettura corretta e la possibilità di muoversi agevolmente tra i contenuti: è chiaro quindi che i testi dei titoli di capitolo, settore o paragrafo dovranno essere chiari e comprensibili.

## Punto di controllo 3.6 (Priorità 2)

### Marcare le liste ed elencare le voci della lista in modo appropriato

Gli elementi di lista `<dl>`, `<ul>` e `<ol>` devono essere utilizzati unicamente per la definizione delle liste e non devono essere utilizzati a scopo di formattazione, ad esempio per rientrare dei contenuti. Vediamo quindi di approfondire la conoscenza di tali elementi.

Un corretto utilizzo delle liste senza ordinamento (unordered list, con l'elemento `<ul>`) è rappresentato nell'esempio seguente:

```
<ul>
<li>Prima voce di lista</li>
<li>Seconda voce di lista</li>
<li>Terza voce di lista</li>
</ul>
```

Per modificare la formattazione delle liste, anche in questo caso ci si affiderà ai fogli di stile. Ad esempio tramite i fogli di stile è possibile modificare il classico pallino delle liste non ordinate per l'elemento `<li>` con una immagine personalizzata.

```
ul {
color: #036;
list-style: url(/immagini/pallino.gif) disc;
}
```

Come si noterà, nell'esempio proposto viene indicato anche il valore `disc`, che consente di visualizzare il classico pallino vuoto se l'immagine indicata dal parametro `url` non è disponibile.

Se le liste non sono definite in modo corretto oppure se si diramano troppo profondamente gli utenti non vedenti durante la navigazione possono perdere l'orientamento: è necessario pertanto che gli sviluppatori mettano a disposizione di tali utenti indicazioni riepilogative per la navigazione dei vari livelli di lista raggiunti.

Finché i CSS2 non saranno supportati in modo uniforme da tutti i programmi utente e sarà possibile ottimizzare la navigazione delle liste tramite CSS, gli sviluppatori dovrebbero considerare di aiutare gli utenti non vedenti fornendo delle "annotazioni", per ricordare all'utente l'argomento principale dell'elenco che sta navigando. Usando i fogli di stile è possibile definire un attributo che renda invisibile questo testo agli utenti vedenti, mentre sarà letto agli utenti non



vedenti dai lettori vocali. Il foglio di stile conterrà quindi una nuova opzione (suggerita da Gianluca Troiani <<http://www.constile.org>> nella lista CSSdesign):

```
.finelista {
width:0;
position:absolute;
height:0;
overflow:hidden;
top:-200em;
}
```

È inoltre possibile creare collegamenti che consentano di saltare gli elenchi contenenti collegamenti ipertestuali, tecnica analizzata nel Requisito 19.

Applicando la classe sopra riportata a un esempio di lista di tipo descrittivo, il codice di una pagina accessibile sarà simile al seguente.

```
<ul>
<li>Veneto:
<ul>
<li>Belluno</li>
<li>Padova</li>
<li>Rovigo</li>
<li>Venezia</li>
<li>Verona</li>
<li>Vicenza</li>
</ul>
<span class="finelista">(Fine Veneto)</span></li>
<li>Liguria:
<ul>
<li>Genova</li>
<li>Imperia</li>
<li>La Spezia</li>

<li>Savona</li>
</ul>
<span class="finelista">(Fine Liguria)</span></li>
</ul>
```

Se invece è necessario dare una certa importanza all'ordine degli elementi, sono disponibili le liste ordinate (ordered list, elemento <ol>):

```
<ol>
<li>Scaricare il modulo</li>
```

```
<li>Compilare il modulo in ogni sua parte</li>
<li>Inviare il modulo</li>
</ol>
```

Considerando che per le liste ordinate l'utilizzo degli attributi `compact`, `start` e `type` è stato deprecato, è possibile aggirare il problema usando CSS2, che permette di definire il formato di auto numerazione delle liste (1, 1.1, 1.1.1, ecc.).

```
ul {
counter-reset: item;
}
li {
display: block;
}
li:before
{
content: counters(item, ".");
counter-increment: item;
}
```

Questa specifica contenuta nella raccomandazione CSS 2.0 attualmente non risulta supportata dai browser, ad eccezione di Opera.

L'ultima tipologia di liste è formata dalle liste di definizione (definition list, elemento `<dl>`). Le liste di definizione sono formate da elementi contenenti il termine (definition term, ovvero l'elemento `<dt>`) e la sua descrizione (definition description, ovvero l'elemento `<dd>`).

```
<dl>
<dt>SEO</dt>
<dd><span xml:lang="en">Search Engine Optimizers</span>,
ovvero gli esperti di posizionamento nei motori di
ricerca.</dd>
</dl>
```

Un'altra possibile applicazione delle liste di definizione, consentita dalla specifica HTML 4.x, è la possibilità di utilizzare tali elementi per riportare dei dialoghi, indicando con l'elemento `<dt>` il nome della persona e con l'elemento `<dd>` il testo del discorso. Un esempio pratico può essere un verbale di una seduta del Consiglio Comunale.

```
<dl>
<dt><abbr title="Consigliere">Cons.</abbr> Rossi</dt>
```

```
<dd>Signor Presidente, chiedo di poter esporre il progetto.<dd>  
<dt>Presidente Consiglio Comunale</dt>  
<dd>Ne ha facoltà.<dd>  
</dl>
```

Utilizzando i fogli di stile sarà quindi possibile rappresentare i contenuti in modo gradevole, affiancando ad esempio l'elemento `<dt>` all'elemento `<dd>`.

Dopo aver analizzato le diverse tipologie di liste è quindi necessario fare delle considerazioni finali che coinvolgono tutte le diverse tipologie analizzate.

Si consiglia innanzitutto di evitare la creazione di lunghi elenchi con le liste, in quanto rendono il sito di difficile utilizzo e ne danneggiano l'usabilità, poiché un utente non riesce a memorizzare contemporaneamente tutti i collegamenti.

Per esempio, un sito istituzionale in cui siano presenti decine di collegamenti ipertestuali ad un gran numero di comunicati stampa o notizie può creare confusione agli utenti, che probabilmente preferirebbero poter consultare singole pagine di riepilogo (per data, per argomento, ecc.) oppure gradirebbero poter saltare tali elenchi utilizzando la navigazione delle intestazioni. Inoltre è necessario precisare che la creazione di menu utilizzando elementi differenti dalle liste (per esempio, utilizzando elementi specifici di tabelle) è un errore che porta alla violazione del primo requisito in quanto i menu sono di natura delle liste di elementi.

## **Punto di controllo 3.7 (Priorità 2)**

### **Marcare le citazioni**

Le citazioni devono essere marcate utilizzando gli elementi `<q>` e `<blockquote>` che non devono essere usati per effetti estetici ma per fornire al browser e alle tecnologie assistive le informazioni necessarie per identificare il blocco di citazione, lasciando ai fogli di stile eventuali effetti di indentazione dei testi.

#### *Elemento <q>*

La rappresentazione delle virgolette tipografiche e dei relativi stili varia da lingua a lingua. L'utilizzo dell'elemento `<q>` al posto del corrispondente carattere ASCII o dell'entità `&quot;`; consente al browser di visualizzare le virgolette secondo la codifica attiva sul computer in uso.

```
<q lang="en">What a wonderful web!</q>
```

Nota: l'elemento `<q>` non risulta completamente supportato dai browser ed in ogni caso per ottenere una corretta interpretazione delle virgolette richiede la presenza dell'attributo `lang` e/o `xml:lang`.

### *Elemento `<blockquote>`*

L'elemento `<blockquote>` è utilizzato per identificare un blocco di testo di citazione che richiede maggior visibilità, per separarlo e distinguerlo dal corpo del testo normale.

```
<blockquote xml:lang="en"
cite="http://www.w3.org/TR/WCAG10-HTML-TECHS/">
<p>
Do not use quotation markup for formatting
effects such as indentation. [Priority 2]
</p>
</blockquote>
```

È da notare che nell'esempio viene utilizzato l'attributo `"cite"`, che riporta l'URI del documento origine della citazione e che è utilizzabile anche per l'elemento `<q>`. L'attributo `cite` può essere utilizzato per identificare anche un URN (Uniform Resource Name) per marcare citazioni che non sono direttamente riferite al web quale per esempio il codice ISBN di un libro.

```
<blockquote cite="urn:isbn:88-7633-000-3">
<p>
Al termine della lettura di questo libro si auspica che
al lettore sia maggiormente comprensibile la necessità
di puntare sull'accessibilità reale, rispettando
le raccomandazioni internazionali senza abusare di loghi
o bollini.
</p>
</blockquote>
```

Un ulteriore esempio è la citazione di RFC (Request for Comments): nell'esempio che segue è citato del testo dalla RFC 2141 emanata dall'IETF.

```
<blockquote xml:lang="en" cite="urn:ietf:rfc:2141">
<p>
Assignment of URNs from this namespace occurs in three ways.
The first is through publication of a new RFC, FYI, STD or BCP
is by the RFC Editor. This new document will have a new series
number and will therefore define a new URN.
```

```
</p>  
</blockquote>
```

Molti sviluppatori però utilizzano l'elemento `<blockquote>` per impaginare i contenuti anziché creare uno stile di paragrafo per l'indentazione del testo. Se si desidera solamente ottenere degli effetti grafici simili a quelli generati dall'elemento `<blockquote>`, in questo caso utilizzando i fogli di stile è possibile definire delle personalizzazioni nella visualizzazione dei paragrafi per l'elemento `<p>`:

```
p.rientro { margin-left: 10%; margin-right: 10%;}
```

mentre nel codice HTML sarà sufficiente richiamare la classe:

```
<p class="rientro">Questo testo è rientrato in quanto...</p>
```

È chiaro quindi che la differenza tra le due soluzioni è semantica: l'elemento `<blockquote>` deve essere utilizzato solamente per la funzionalità assegnatagli (citazioni), mentre per ottenere effetti grafici similari è necessario evitare l'uso di `<blockquote>` ricorrendo a paragrafi correttamente formattati tramite fogli di stile.

### **Punto di controllo 11.1 (Priorità 2). Utilizzare le tecnologie W3C quando sono disponibili e sono appropriate per un certo compito usando le versioni più recenti quando sono supportate dai programmi utente.**

I gruppi di lavoro che si occupano della creazione delle specifiche delle raccomandazioni del W3C tengono in considerazione l'attività del progetto WAI e quindi coinvolgono gli esperti di accessibilità per valutare di volta in volta come le nuove specifiche possano rispettare il principio di universalità del World Wide Web: è anche questo il motivo per cui le raccomandazioni del W3C sono universalmente riconosciute come standard de facto. È pur vero che con il rilascio di nuove specifiche i browser e le tecnologie assistive devono aggiornare i loro motori per rendere fruibili queste nuove caratteristiche, considerando che tutte le nuove specifiche mirano a mantenere una compatibilità con le precedenti. Può quindi succedere che attributi come `longdesc` per le immagini non vengano riconosciuti e ignorati da alcuni browser e/o tecnologie assistive ma questo non significa che per una tecnologia che non lo utilizza non ve ne siano altre che rispettano le raccomandazioni del W3C. Un altro problema che si è riscontrato è relativo al mancato supporto dell'elemento `<object>` da parte di Microsoft Internet Explorer e da Netscape che causa problemi di accessibilità (per esempio per la creazione di mappe immagine lato client). I contenuti delle pagine Web

devono quindi validare secondo le grammatiche formali utilizzando per esempio il W3C Markup Validator o il validatore di fogli di stile. È bene quindi cercare di mantenere le proprie pagine secondo le ultime specifiche del W3C: l'elenco completo e costantemente aggiornato è disponibili nella pagina W3C Technical Reports and Publications <<http://www.w3.org/TR/>>.

Alcune di queste specifiche sono le seguenti:

- MathML per equazioni matematiche;
- HTML, XHTML, XML per la struttura dei documenti;
- RDF per i meta data;
- SMIL per creare presentazioni multimediali;
- CSS e XSL per la definizione dei fogli di stile;
- PNG per la grafica.

Quali sono quindi i programmi utente (user agent) che supportano le raccomandazioni W3C, e in che modo? Non è facile rispondere alla domanda e neppure il W3C ci riesce nella pagina dedicata al supporto dei programmi utente alle linee guida WAI. Sul Web sono disponibili alcune risorse, come [webstandards.org](http://www.webstandards.org), che hanno la finalità di fornire informazioni sull'effettiva applicazione degli standard, mentre sul sito del W3C è presente una sezione di riferimento <<http://www.w3.org/WAI/Resources/WAI-UA-Support>> delle risorse che supportano le User Agent Accessibility Guidelines.

## **Punto di controllo 11.2 (Priorità 2). Evitare l'utilizzo di tecnologie disapprovate dal W3C.**

Con la definizione di nuove raccomandazioni e con l'avvento di nuove tecnologie alcuni elementi e attributi possono essere eliminati a favore di nuove raccomandazioni che ne migliorano l'utilizzo. Un esempio concreto è dato dall'elemento `<font>` che è stato disapprovato (deprecated) dal W3C a favore dei fogli di stile che consentono di separare il contenuto dalla presentazione. Per lo stesso motivo, l'attributo "color" è stato deprecato, sempre a favore dell'utilizzo dei fogli di stile. Gli elementi e gli attributi disapprovati diventano quindi obsoleti e vengono solitamente mantenuti nelle grammatiche dei DOCTYPE di tipo Transitional (ossia grammatiche di transizione) mentre vengono rimossi nelle grammatiche dei DOCTYPE di tipo Strict: un ulteriore esempio è dato dall'elemento `<center>` e dall'attributo `target` per i collegamenti che in XHTML 1.0 Strict e XHTML 1.1 non è previsto dalla DTD e quindi non è utilizzabile, nemmeno utilizzando la cosiddetta DOM Injection che vedremo nel paragrafo dedicato alla verifica del requisito. Un interesse particolare va verso gli elementi `<b>` / `<strong>` e `<i>` / `<em>`. Entrambe le coppie di elementi si occupa

rispettivamente di formattare il testo in grassetto e in corsivo ma il loro significato semantico è nettamente differente: gli elementi `<strong>` ed `<em>` servono per enfatizzare il contenuto (hanno quindi valore "logico"), mentre `<b>` ed `<i>` hanno uno scopo puramente decorativo, tenendo presente che entrambe le coppie di elementi sono disponibili anche nelle versioni più recenti di XHTML.

Elementi come `<applet>` ed `<embed>` sono stati sostituiti dall'elemento `<object>`, che consente di trasformare il contenuto in modo gradevole nel caso non sia presente l'oggetto richiesto ma il cui supporto però con alcuni browser e tecnologie assistive non è ancora ottimale. Per la massima accessibilità si consiglia quindi di utilizzare le grammatiche con DOCTYPE di tipo Strict. L'elenco degli elementi ed attributi utilizzabili <http://www.w3.org/TR/WCAG10-HTML-TECHS/#html-index> è disponibile direttamente nel sito del W3C, dove con asterisco sono indicati gli elementi e gli attributi disapprovati. Ma come possiamo ottenere gli effetti grafici e le impostazioni che utilizzavamo tramite l'elemento `font` tramite i fogli di stile ed allo stesso tempo garantire l'accessibilità della nostra pagina web? Invece di utilizzare elementi ed attributi disapprovati per controllare la presentazione dei documenti è possibile, per esempio, utilizzare le seguenti proprietà CSS:

- `font` (solo CSS 2.x)
- `font-family`
- `font-size`
- `font-size-adjust` (eliminata in CSS 2.1)
- `font-stretch` (eliminata in CSS 2.1)
- `font-style`
- `font-variant`
- `font-weight`
- `color`
- `background-color`

ecc.

È importante controllare sempre il contrasto tra il colore di sfondo ed il colore del testo (Requisito 6), consentire il ridimensionamento dei caratteri utilizzando unità di misura in em o in percentuale (Requisito 12) e definire sempre un tipo di carattere generico per consentire a chiunque di fruire del contenuto anche in mancanza di un particolare carattere prescelto dallo sviluppatore.

## Prima applicazione

---

In caso di prima applicazione nel caso si utilizzino DTD di tipo Transitional l'enunciato richiede di evitare sia l'uso di attributi di formattazione, a favore dell'utilizzo di fogli di stile, sia l'apertura di nuove finestre. L'allineamento, i margini e il posizionamento degli elementi nella presentazione di una pagina vanno gestiti tramite fogli di stile: lo sviluppatore, tuttavia, deve garantire la possibilità di lettura dei contenuti della pagina anche in mancanza dei fogli di stile stessi, al fine di garantirne l'accesso e la fruibilità agli utenti che utilizzano lettori solo testo o vocali.

**Allineamento:** utilizzando le proprietà `text-indent`, `text-align`, `word-spacing`, `letter-spacing` e `white-space` è possibile formattare il testo aumentando/diminuendo gli spazi, definendo l'allineamento e controllando il comportamento dello spazio bianco. Con la regola `text-align: center`, per esempio, sarà possibile eliminare l'elemento `<center>`, ormai obsoleto.

**Margini:** le proprietà `margin`, `margin-top`, `margin-right`, `margin-bottom`, `margin-left` consentono di definire i margini dell'elemento (`<div>`, `<blockquote>`,...) all'interno del quale verranno forniti i contenuti.

**Posizionamento:** le proprietà `float`, `position`, `top`, `right`, `bottom`, `left` consentono di posizionare l'oggetto nella pagina.

Si consiglia di evitare l'uso degli stili creati con l'attributo HTML `"style"`: nelle versioni più recenti di XHTML (1.1 e successive) tale elemento è stato deprecato per garantire una maggiore separazione di contenuto e presentazione.

In HTML utilizzando i `frame`, si consente l'utilizzo dell'attributo `target` che consente di definire la destinazione del collegamento. Questa pratica dedicata ai `frame` è stata poi spesso utilizzata in modo scorretto, a causa della disponibilità di alcuni valori per l'attributo `target` che consentono di aprire nuove finestre o di aprire il contenuto di un `frame` sull'intera finestra del browser.

```
<a href="ultime_notizie.htm" target="_blank">Ultime Notizie</a>
```

Utilizzando l'attributo `target="_blank"` selezionando il collegamento verrà aperta la pagina `ultime_notizie.html` in una nuova finestra. L'esempio così come riportato non è corretto rispetto ai requisiti di questo punto di controllo, in quanto non informiamo l'utente che la pagina sarà aperta in una nuova finestra. È quindi quantomeno necessario informare l'utente, utilizzando l'attributo `title`:

```
<a href="ultime_notizie.htm" target="_blank"
title="Attenzione: si aprirà una nuova Finestra">Ultime Notizie</a>
```



Inoltre nella nuova finestra l'utente non troverebbe la propria cronologia di navigazione, in quanto non esistendo una pagina precedente il pulsante Precedente del browser risulterebbe disattivato, e perderebbe quindi l'orientamento.

Anche per questo motivo l'attributo target non è disponibile in HTML 4.01 Strict e XHTML 1.0 Strict e scompare completamente a partire da XHTML 1.1.

## Verifica del requisito

---

Il controllo del codice va effettuato secondo il tipo di grammatica utilizzata. Per quanto riguarda i contenuti web, solitamente si utilizza il validatore per il linguaggio di marcatura del W3C mentre per i fogli di stile CSS si utilizza il validatore dei fogli di stile fornito sempre dal W3C.

### *Controllo della validità del linguaggio di marcatura*

Il W3C rende disponibile agli sviluppatori un sistema di controllo automatico della sintassi di una pagina Web, ovvero il Markup Validation Service <<http://validator.w3.org/>>. La pagina Web si presenta con una veste grafica gradevole contenente due moduli.

- **Validate by URL.** Richiede l'indirizzo della pagina Web che si desidera validare.
- **Validate by File.** Richiede la selezione di un documento in locale dovrà essere trasferito on-line per essere validato.

Selezionando Extended interface, è possibile personalizzare la modalità di validazione del contenuto e dove dovremmo selezionare le opzioni:

- **outline:** per la visualizzazione della struttura degli elementi di intestazione. In caso di mancanza di un elemento, il validatore segnalerà l'elemento mancante (es: <h2> missing!).
- **show source:** per visualizzare il sorgente della pagina, al fine di poter identificare con un click la riga che causa l'errore di validazione.

Impostando correttamente questi parametri e avendo soprattutto cura di rispettare le grammatiche formali dichiarate per il documento, possiamo segnalare la pagina al validatore che, in caso di errori, ci comunicherà:

- la riga e la colonna dove inizia l'errore, con un collegamento ipertestuale alla riga di codice riportata nella pagina;
- la rappresentazione di parte della linea di codice contenente l'errore, con posizionamento del punto di errore;
- il messaggio di errore, in alcuni casi con collegamento ad una pagina descrittiva del tipo di errore.

Se nella modalità avanzata selezioniamo anche la voce "Verbose Output", verranno fornite indicazioni aggiuntive relative agli errori individuati all'interno del documento. Un errore dei più comuni è il seguente:

```
non SGML character number 146
```

Questo errore normalmente si riscontra quando, all'interno di sistemi CMS o di editor visuali, si procede a un copia e incolla di contenuti provenienti da sistemi di gestione testi (word processor).

L'errore indica che è stato utilizzato un carattere illegale. In HTML/XHTML per la codifica dei caratteri (salvo diversa dichiarazione) si utilizza solitamente lo standard ISO8859-1, che dichiara 65 caratteri non definiti (da 0 a 31 inclusi e da 127 a 159 inclusi). Il W3C Markup Validator ha individuato nel testo uno di questi caratteri, nel nostro caso il carattere 146, che in un browser potrà essere mostrato come un apice rovesciato o una particolare virgoletta, ma che nel sistema di un altro utente potrebbe essere un carattere totalmente diverso o non essere visibile del tutto.

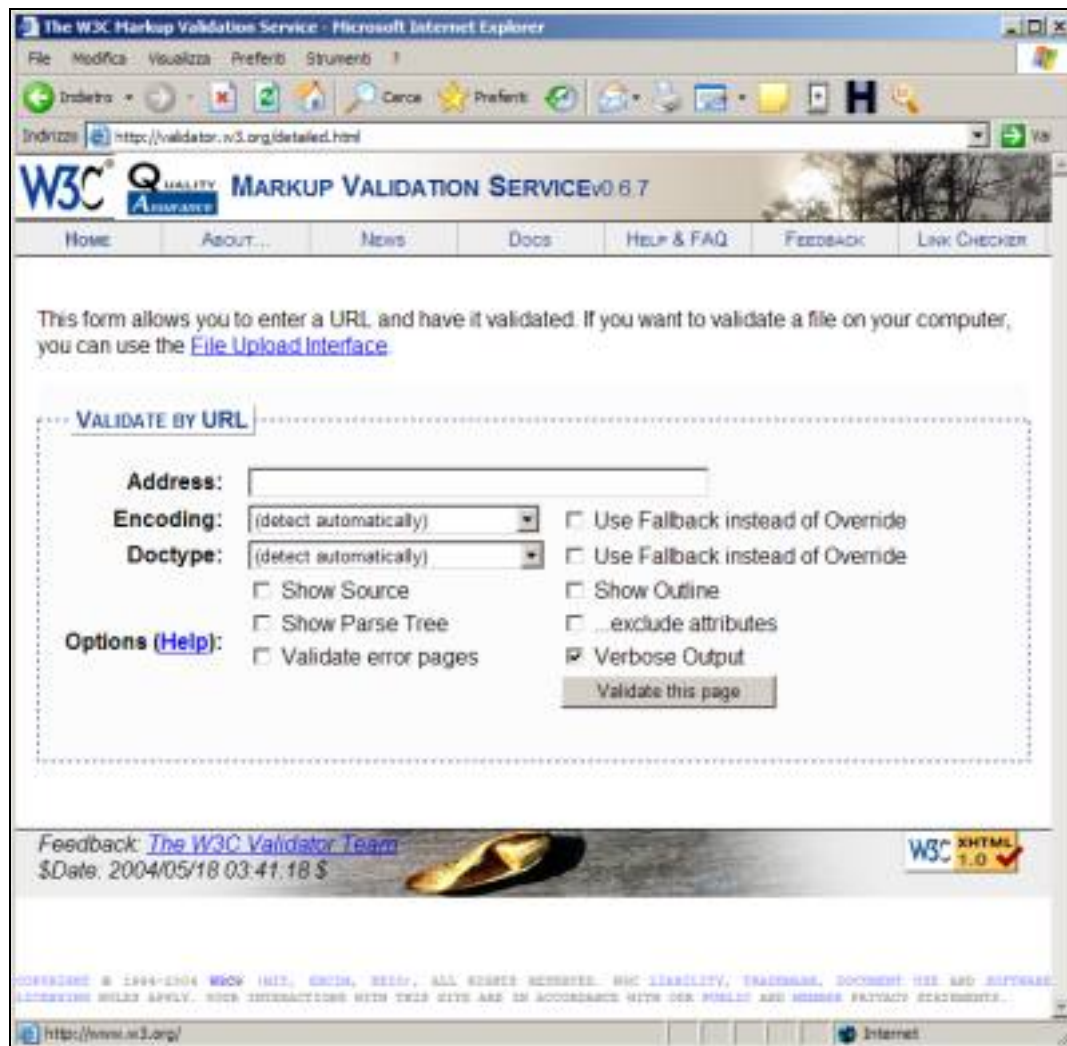


Figura 4.1 Interfaccia del W3C Markup Validator.

Il W3C consiglia di sostituire questo carattere con il suo equivalente tra i caratteri definiti in ASCII tra il numero 32 e il numero 126, oppure di utilizzare l'appropriata entità identificativa del carattere come indicato sul sito del W3C <<http://www.w3.org/MarkUp/html3/latin1.html>>.

Nel caso di contenuti in lingue diverse da quelle europee, è consigliato utilizzare come codifica utf-8 che consente l'uso di un maggior numero di caratteri e riconosce anche alcuni caratteri solitamente indicati come errori SGML in ISO8859-1 quali, ad esempio, le virgolette tipografiche e i tre puntini di sospensione.

Quando si utilizza un sistema di gestione dei contenuti (CMS), un altro classico errore è dato dai collegamenti che contengono le variabili necessarie a far funzionare correttamente la nuova pagina:

```
<a href="pagina.asp?id=1&cat=2&zzz=a">Pagina 1</a>
```

Il validatore, in presenza di tale istruzione, indicherà diversi errori di definizione di identità:

```
l'identità "cat" non è definita
```

```
l'identità "zzz" non è definita
```

L'errore è causato da un errato utilizzo della codifica HTML: quando il validatore individua il carattere &, ritiene che successivamente sarà definita l'entità identificativa di un carattere e quindi l'utilizzo della semplice & per la separazione dei parametri è un errore: bisogna codificare & con la sua equivalente entità identificativa, &amp;. Nel nostro esempio, per superare la validazione automatica del W3C Markup Validation Service il codice dovrà essere modificato come segue:

```
<a href="pagina.asp?id=1&amp;cat=2&amp;zzz=a">Pagina 1</a>
```

Altri errori possono essere causati dall'utilizzo di attributi proprietari definiti da alcuni editor visuali, attributi che non fanno parte delle DTD dichiarate e quindi non utilizzabili e per i quali il messaggio di errore è:

```
There is no attribute "XXXXX"
```

Attenzione all'effetto a cascata degli errori: spesso gli errori successivi all'attuale vengono visualizzati esclusivamente perché un determinato elemento (quello con l'errore corrente) non risulta chiuso.

L'elenco delle domande più frequenti sugli errori di validazione è disponibile all'interno del sito del W3C nella sezione dedicata <<http://validator.w3.org/docs/errors.html>> al W3C Markup Validation Service.

Se al termine della validazione verrà visualizzato il messaggio No errors found!, è possibile apporre il logo Valid HTML o XHTML (a seconda del tipo di documento) sulla pagina, seguendo le indicazioni riportate nella pagina dedicata alla spiegazione delle norme per la riproduzione del logo.

Logicamente, se si utilizzano ulteriori linguaggi di marcatura nel documento (X)HTML, per esempio codice MathML, anch'essi verranno validati secondo lo schema DTD assegnato.

Nel sito del W3C sono presenti alcune DTD con integrati linguaggi come MathML o SVG utilizzabili a tal fine <<http://www.w3.org/QA/2002/04/valid-dtd-list.html>>.

Attenzione comunque che il validatore del W3C non è infallibile: non controlla la ricchezza di marcatura (quindi acronimi, abbreviazione e citazioni o quoting) e soprattutto non controlla eventuali modifiche del codice della pagina. Tramite DOM Injection, ad esempio, con del codice javascript richiamato da un documento esterno è possibile modificare degli attributi e degli elementi contenuti nella pagina in modo da superare la validazione automatica ma variando il contenuto della pagina web aggiungendo elementi ed attributi non validi. Il caso più classico è un javascript esterno per aggiungere l'attributo target="\_blank" alle pagine XHTML 1.0 Strict.

Ecco un esempio da evitare in quanto se un valutatore identifica tale errata procedura dovrà segnalarlo sul modulo di valutazione come mancato superamento del Requisito 1:

```
function linkesterno() {  
for (var i=0; i<document.links.length; i++) {  
if (document.links[i].className=="linkEsterno") {  
document.links[i].target="_blank";  
}  
}  
}
```

In questo esempio a qualsiasi elemento contrassegnato dalla classe "linkesterno" il codice Javascript forza l'inserimento dell'elemento target="\_blank". All'interno della pagina il codice sarà simile al seguente:

```
<a href="pagina.html" class="linkEsterno">Pagina</a>
```

mentre il codice reale fornito al browser ed identificabile con applicativi come DOM Inspector <Hhttp://www.ieinspector.com> sarà:

```
<a href="pagina.html" class="linkEsterno" target="_blank">Pagina</a>
```

Tramite questo codice il W3C Markup Validator non potrà identificare a violazione della DTD, mentre l'esperto valutatore potrà chiaramente identificarla.

Il valutatore, al fine di verificare l'applicazione del requisito dovrà inoltre utilizzare strumenti di supporto per:

- verificare che la DTD dichiarata sia conforme al requisito 1, ovvero per tutti i nuovi siti internet sia utilizzata la DTD HTML 4.01 Strict, XHTML 1.0 Strict o XHTML 1.1;

- verificare il corretto utilizzo degli elementi di intestazione, delle citazioni e delle liste;
- verificare la presenza di immagini che sostituiscono oggetti rappresentabili tramite linguaggi di marcatura;
- verificare l'utilizzo e il corretto utilizzo di elementi ed attributi secondo le specifiche del linguaggio di marcatura utilizzato.

Tramite la Barra dell'accessibilità è possibile utilizzare:

## **Validazioni**

*W3C HTML Validator - Valida HTML (nuova finestra)*. Consente di inviare direttamente il documento alla validazione (Punto di controllo 3.2) e di verificare la corretta corrispondenza degli elementi di intestazione (Punto di controllo 3.5).

## **CSS**

*Valida il CSS*. Controlla il codice dei CSS della pagina corrente con il validatore CSS del W3C (Punto di controllo 3.2).

*Elementi ed attributi deprecati (Nuova finestra)*. Visualizza gli elementi e gli attributi deprecati (Punto di controllo 11.2).

## **Struttura**

*Intestazioni*. Visualizza tutti gli elementi d'intestazione sulla pagina corrente (Punto di controllo 3.5).

*Struttura delle intestazioni (Nuova finestra)*. Visualizza il titolo del documento e le sue intestazioni (da <h1> ad <h6> compreso il contenuto) in una nuova finestra. (Punto di controllo 3.5).

*Elementi di lista*. Visualizza le liste ordinate, non ordinate, di intestazioni presenti sulla pagina corrente (Punto di controllo 3.6).

*Blockquote / q*. Visualizza gli elementi <blockquote> e <q> presenti all'interno della pagina (Punto di controllo 3.7).

## **Sorgente**

*Visualizza codice generato*. Consente di visionare il codice generato per verificare l'eventuale modifica di elementi ed attributi tramite DOM (Punto di controllo 3.2).

*Visualizza parte del codice.* Consente di visionare il codice generato per il contenuto selezionato al fine di consentire l'identificazione di utilizzo scorretto di elementi ed attributi, modificati tramite DOM (Punto di controllo 3.2).

*Evidenzia nel sorgente.* Contiene una serie di sottovoci che consentono di evidenziare elementi ed attributi specifici all'interno del sorgente generato.

DOM Inspector. Consente di verificare con dei programmi esterni le eventuali modifiche al codice tramite DOM (Punto di controllo 3.2).

L'esperto dovrà quindi accedere al codice della pagina HTML e verificare manualmente l'uso corretto degli elementi e degli attributi secondo le regole definite dalle specifiche delle grammatiche formali utilizzate.

### ***Controllo della validità dei fogli di stile***

Come per le pagine Web, anche per i fogli di stile il W3C ha creato un'applicazione di validazione, disponibile sia on-line sia con possibilità di utilizzo in locale. Nel nostro esempio utilizzeremo la versione on-line <<http://jigsaw.w3.org/css-validator/>>, di cui ho personalmente curato la traduzione in lingua italiana.

Per procedere con la validazione del foglio di stile, sarà sufficiente digitare l'URL del foglio di stile o della pagina Web che lo contiene (se lo stile è interno al documento). Il risultato della validazione, in caso di mancanza di errori gravi, può essere di due tipi: validazione corretta ma con avvisi oppure validazione corretta senza avvisi. Nel primo caso il foglio di stile risulta valido ma sono presenti alcuni avvisi relativi, ad esempio, al mancato utilizzo di caratteri di tipo generico, fatto che può compromettere il livello di accessibilità delle pagine.

Nel caso fossero presenti errori di sintassi, ogni errore verrà segnalato con indicazione della riga e dell'attributo errato consentendo un facile intervento per la correzione del foglio di stile.

Gli errori più frequenti nei fogli di stile sono i seguenti:

- mancanza degli elementi di chiusura: è necessario apporre il punto e virgola al termine della dichiarazione di ogni proprietà e apporre la parentesi graffa chiusa per terminare l'elenco delle proprietà di un elemento;
- utilizzo di gruppi di caratteri senza prevedere almeno un carattere di tipo generico (per esempio, sans-serif), oppure utilizzo di nomi per tipi di

caratteri contenenti spazi che non vengono racchiusi tra apici (per esempio, "Times New Roman");

- utilizzo di valori non validi per i colori: si consiglia di usare sempre valori di tipo esadecimale antepoendo il carattere #;
- utilizzo di nomi non validi, come ".bell'immagine".

Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)



## Requisito 2

Enunciato: Non è consentito l'uso dei frame nella realizzazione di nuovi siti. In sede di prima applicazione, per i siti Web esistenti già realizzati con frame è consentito l'uso di HTML 4.01 o XHTML 1.0 con DTD frameset, ma con le seguenti avvertenze:

- evitare di utilizzare, all'interno del linguaggio a marcatori con il quale la pagina è realizzata, elementi ed attributi per definirne le caratteristiche di presentazione della pagina (per esempio, caratteristiche dei caratteri del testo, colori del testo stesso e dello sfondo, ecc.), ricorrendo invece ai Fogli di Stile CSS (Cascading Style Sheets) per ottenere lo stesso effetto grafico;
- fare in modo che ogni frame abbia un titolo significativo per facilitarne l'identificazione e la navigazione; se necessario, descrivere anche lo scopo dei frame e la loro relazione;
- pianificare la transizione a XHTML almeno nella versione 1.0 con DTD Strict dell'intero sito dandone comunicazione alla Presidenza del Consiglio dei Ministri - Presidenza del Consiglio dei Ministri - Dipartimento per l'innovazione e le tecnologie e al Centro nazionale per l'informatica nella pubblica amministrazione.

Riferimenti WCAG 1.0: 12.1, 12.2

Riferimenti Section 508: 1194.22 (i)

### Motivazione

---

L'impaginazione tramite frame ha la sua origine con le prime versioni di HTML ed è dovuta alla necessità, in alcuni casi, di dover organizzare i contenuti in modo da poter garantire la permanenza di visualizzazione di alcuni contenuti (ad esempio una serie di menu). I problemi che si incorrono con l'uso di tale tecnologia sono legati ai seguenti fattori:

- problemi di accessibilità tramite tecnologie assistive. L'utente con lettore di schermo, se i frame non vengono identificati con chiarezza, incorrono in problemi di navigazione e devono "saltare" tra i vari frame rischiando di perdere l'orientamento nella navigazione;

- problemi di accesso ai contenuti. Con l'uso di frame è possibile che un utente acceda direttamente ai contenuti di una pagina interna al frame (es: se proviene da un motore di ricerca) e pertanto non avrà possibilità di navigare il sito web, mancando di fatto i punti di riferimento contenuti in altri frame.

La normativa quindi richiede ai siti di nuova realizzazione di non utilizzare i frame in modo categorico: non è indicato, come in altri casi, di evitare ma non sono consentiti. I riferimenti alle WCAG 1.0 e le tecniche di verifica che andremo ad analizzare riguardano quindi esclusivamente la prima applicazione del requisito.

## Implementazione

---

Come vedremo dall'analisi dei riferimenti delle WCAG 1.0, un frame è parte di un frameset, ossia un insieme di frame che per essere accessibili agli utenti con tecnologie assistive necessitano di:

- chiare titolazioni;
- descrizioni relative allo scopo dei frame;
- possibilità di passare da un frame all'altro.

Non mi soffermo sull'implementazione di una struttura a frame in quanto la normativa non ne consente l'uso, ma riporto un esempio di codice relativo ad un frameset che sarà la nostra base di implementazione degli esempi di adeguamento dei frame in sede di prima applicazione.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="it" xml:lang="it">
<head>
<title>Esempio di frameset</html>
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-1" />
</head>
<frameset cols="20%, 80%">
<frame src="menu.html" />
<frame src="main.html" />
</frameset>
</html>
```

## Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

### **Punto di controllo 12.1 (Priorità 1). Assegnare un titolo a ogni frame per facilitarne l'identificazione e la navigazione**

Alcuni programmi utente non possono accedere a più frame contemporaneamente o possono essere configurati esclusivamente per visualizzare un solo frame: un esempio sono le applicazioni software di supporto per gli utenti non vedenti. Utilizzando l'attributo "title" per i frame è possibile fornire informazioni al visitatore sul contenuto di quella particolare pagina web, consentendo all'utente di selezionare il frame desiderato.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="it" xml:lang="it">

<head>
<title>Esempio di frameset</html>
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-1" />
</head>

<frameset cols="20%, 80%" title="Esempio di frameset" >
<frame src="menu.html" title="Navigazione" />
<frame src="main.html" title="Pagina iniziale" />
</frameset>

<noframes>
<body>
<ul>
<li><a href="menu.html">Navigazione</a></li>
<li><a href="main.html">Pagina iniziale</a></li>
</ul>
</body>
</noframes>

</html>
```

Nell'esempio abbiamo riportato anche l'elemento `<noframes>` al fine di garantire l'accessibilità della pagina anche ai programmi utente che non supportano i frame, per rispettare il requisito 3.

## **Punto di controllo 12.2 (Priorità 2). Descrivere lo scopo dei frame e il modo in cui questi interagiscono se non è evidente solo tramite i titoli dei frame**

Nel punto di controllo 12.1 abbiamo visto che utilizzando l'attributo `title` possiamo dare alcune informazioni sul contenuto dei frame. In alcuni casi però l'attributo `title` può non essere sufficiente per spiegare la struttura o lo scopo dell'elemento `<frame>` o dell'elemento di contenimento `<frameset>`. Poniamo il caso di una pagina con cinque frame: uno per la testata, uno a sinistra per la navigazione dei menu, uno centrale per i contenuti, uno a destra per le ultime notizie ed uno alla fine per la chiusura di pagina: come è possibile spiegarne la struttura tramite l'attributo `title`? Già in questo paragrafo per spiegare in modo rapido una struttura teorica ho utilizzato diverse righe di testo, che non avrei potuto utilizzare con l'attributo `title`. In questi casi è consigliabile utilizzare l'attributo `longdesc` per l'elemento `frame` al fine di rendere disponibile un collegamento ad un documento (pagina senza frame) che contiene una descrizione completa per una pagina con una struttura complessa di frame. Il nostro codice di pagina quindi cambierà nel modo seguente:

```
<frameset cols="20%, 80%" title="Esempio di frameset"
longdesc="descr-frameset.html" >
<frame src="menu.html" title="Navigazione" />
<frame src="main.html" title="Pagina iniziale"
longdesc="descr-main.html" />
</frameset>
```

## **Prima applicazione**

---

È bene ribadire che i frame sono supportati dalle raccomandazioni W3C che comunque ne sconsiglia l'utilizzo ma la normativa consente l'uso in fase di prima applicazione con alcuni accorgimenti.

Innanzitutto il requisito richiede di non utilizzare all'interno dei frame delle pagine web contenenti elementi ed attributi presentazionali. Questo significa che utilizzando la DTD Frameset è possibile utilizzare pagine con DTD Transitional (in quanto la DTD Strict non prevede l'attributo `"target"` per i link) ma evitando elementi ed attributi di formattazione, come già abbiamo visto nel primo requisito. Si richiede sempre in sede di prima applicazione di rendere i frame

chiaramente identificabili seguendo le indicazioni del paragrafo relativo ai punti di controllo 11.1 e 11.2 delle WCAG 1.0.

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la Barra dell'accessibilità per:

- verificare che la DTD dichiarata sia conforme al requisito, ovvero per i siti esistenti utilizzo di DTD Frameset e DTD Transitional per i singoli frame;
- verificare l'utilizzo di elementi ed attributi del linguaggio di marcatura utilizzati a scopo presentazionale;
- verificare la presenza di titoli ed eventuali descrizioni per i frame.

L'esperto dovrà quindi controllare i testi degli attributi "title" e valutare se siano comprensibili e significativi per garantire informazioni. Per effettuare questa verifica, dovrà immaginare di trovarsi al telefono e di dover operare delle scelte (premi 1 per xxx, premi 2 per yyy): se i testi risulteranno comprensibili, allora il sito internet avrà superato il requisito 2. Tramite la Barra dell'accessibilità è possibile utilizzare:

### **Validazione**

W3C HTML Validator - Valida HTML (nuova finestra). Consente di inviare direttamente il documento alla validazione per verificare la correttezza delle grammatiche formali del frameset.

### **Struttura**

Bordi dei frame. Evidenzia i bordi di ciascun elemento frame sulla pagina corrente (Punto di controllo 12.1).

Name / Title dei frame. Mostra (in una nuova finestra) una lista delle pagine contenute nel frameset insieme ai rispettivi attributi "name" ed al contenuto dell'attributo "title" (Punto di controllo 12.1 e 12.2).

In caso di presenza di frameset, l'esperto dovrà quindi identificare i singoli frame verificandone la conformità rispetto alla DTD dichiarata (solitamente DTD Transitional). Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a elenco dei requisiti](#)

## Requisito 3

Enunciato: Fornire una alternativa testuale equivalente per ogni oggetto non di testo presente in una pagina e garantire che quando il contenuto non testuale di un oggetto cambia dinamicamente vengano aggiornati anche i relativi contenuti equivalenti predisposti; l'alternativa testuale equivalente di un oggetto non testuale deve essere commisurata alla funzione esercitata dall'oggetto originale nello specifico contesto.

Riferimenti WCAG 1.0: 1.1, 6.2

Riferimenti Section 508: 1194.22 (a)

### Motivazione

---

Questo requisito dovrà assolutamente essere soddisfatto se desideriamo che il nostro sito possa essere considerato accessibile ai livelli minimi. Molti utenti utilizzano strumenti di navigazione che si basano sui soli contenuti testuali (esempio: i lettori di schermo per i non vedenti) è quindi necessario fornire equivalenti testuali per immagini, rappresentazioni grafiche di testo (compresi i simboli), zone sensibili delle immagini (mappe immagine), animazioni (per esempio, GIF animate), applet e oggetti, ASCII Art, frame, script, immagini usate come richiamo per elenchi, spaziatori, pulsanti grafici, suoni (azionati con o senza l'intervento dell'utente), file di solo audio, tracce audio di video e video. Allo stesso modo le informazioni che vengono rese disponibili in tempo reale, come le ultime notizie all'interno dei siti di informazione, sono chiamate "contenuto dinamico", ossia una tipologia di contenuto che può essere modificata.

In un sito di informazione, le ultime notizie solitamente scorrono in forma sintetica all'interno di applet Java e al clic del mouse viene aperta una nuova finestra contenente la notizia completa: i visitatori con problemi di mobilità, oppure gli utenti che non dispongono di Java non potranno accedere, o accederanno con difficoltà, ai contenuti, così come gli utenti con disabilità visive. È quindi necessario fornire una versione alternativa, in HTML, che consenta l'accesso ai contenuti a queste categorie di utenti, nascondendo questa versione a chi invece dispone delle tecnologie necessarie per la normale visualizzazione della pagina.

## Implementazione

---

Il requisito chiede di garantire a chiunque di poter accedere a informazioni relative ad immagini ed oggetti in formato testuale. Lo sviluppatore dovrà quindi seguire una serie di tecniche di implementazione che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti testuali equivalenti. La mancata accessibilità dovuta agli elementi non testuali è uno dei problemi maggiori riscontrati nei siti web attuali e pertanto, se dobbiamo dare un peso al requisito, il requisito 3 risulta essere tra i più importanti per gli utenti non vedenti - ribadendo che comunque per le pubbliche amministrazioni è fatto obbligo rispettare tutti i 22 requisiti nello sviluppo di nuovi siti internet.

## Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

### **Punto di controllo 1.1 (Priorità 1)**

#### **Fornire un equivalente testuale per ogni elemento non di testo**

Come è stato chiaramente indicato nella motivazione di applicazione del requisito, è necessario garantire un equivalente testuale per una serie di contenuti che andremo ad analizzare qui di seguito.

### *Immagini*

Riportiamo un esempio d'uso dell'attributo alt relativo a un'immagine. Il testo alt è destinato a fornire un testo alternativo, da usare principalmente quando l'immagine non viene visualizzata. L'errore più comune consiste nell'offrire una descrizione testuale dell'immagine senza valutare quale fosse la sua funzione nella pagina, con risultati incongrui o, in alcuni casi, assurdi. Talvolta questa scelta potrebbe sembrare la più opportuna, ma nella pratica molto spesso è quella sbagliata.

Un esempio di utilizzo dell'attributo alt è il seguente:

```

```

In questo caso un testo alternativo "Grafico vendite 2004" non sarà sufficiente a fornire informazioni equivalenti ad un utente non vedente e sarà necessario

fornire una descrizione più completa del contenuto dell'immagine. Si ricorrerà allora all'attributo `longdesc`, che permette di accedere a una descrizione estesa dell'immagine specificando l'URI del documento che contiene tale descrizione.

Se nella pagina sono presenti diverse immagini e ciascuna necessita di una descrizione estesa di norma si creano file diversi, oppure è lecito creare un unico file contenente i puntatori che consentono di raggiungere direttamente la descrizione richiesta. I sistemi CMS (Content Management System) più recenti generano dinamicamente i documenti per l'attributo `longdesc`, ove richiesto.

```

```

Da HTML 4.0x l'attributo "alt" per le immagini è obbligatorio: la mancanza di tale attributo non consentirà di superare la validazione formale della pagina (requisito 1). Anche se le immagini sono esclusivamente a scopo decorativo è comunque necessario inserire l'attributo alt, senza valorizzarlo. Un lettore vocale leggerebbe descrizioni non contestualizzate, evitate quindi di usare codice simile al seguente in quanto l'utente con lettore di schermo per ogni voce di testo otterrà la lettura di "Pallino".

```
<p>
 Testo 1<br />
 Testo 2<br />
 Testo 3
</p>
```

### ***Mappe immagine***

Le mappe immagine sono un caso speciale di questa categoria. In alcune situazioni, le mappe immagine svolgono funzioni non replicabili con altri tipi di oggetti (per esempio, nel caso delle cartine geografiche usate come strumento di navigazione), ma possono svolgere anche altre funzionalità, come un indice alfabetico o una casella di ricerca, che possono risultare utili a tutti.

Non bisognerebbe quindi tralasciare la necessità di inserire gli equivalenti testuali in quanto si renderebbe difficile la navigazione agli utenti non vedenti che utilizzano dei lettori di schermo. Anche l'elemento `<area>` supporta il testo alt, che può essere utilizzato per realizzare menu in modalità testuale (come con l'attributo alt dell'elemento `<img>`). Secondo le raccomandazioni W3C HTML 4.0 e successive, questo attributo è obbligatorio e lo è a buon motivo, dal momento che può provvedere una valida alternativa per gli utenti con browser testuali.



Si riporta a titolo di esempio parte del codice di una mappa dell'Italia suddivisa per regioni, estraendo solo alcuni dati di tipo "rect" e "poly" (si tratta di una mappa lato client):

```

<map id="csmmap" name="csmmap">
<area shape="rect" alt="Sardegna" coords="114,238,160,31"
href="sardegna.htm" />
<area shape="poly" alt="Sicilia" coords="304,330,292,354"
href="sicilia.htm" />
<area shape="poly" alt="Calabria" coords="304,283,319,12"
href="calabria.htm" />
</map>
```

L'esempio di mappa rispetta la specifica XHTML 1.0 Strict. XHTML 1.1 ha cambiato la gestione delle immagini mappate lato client, in quanto ha eliminato l'utilizzo degli attributi "name" sostituendoli con gli attributi "id", ridefinendo inoltre l'uso dell'attributo "usemap" trasformandolo da tipo URI a tipo IDREF. Pertanto il precedente esempio diventa:

```

<map id="csmmap">
<area shape="rect" alt="Sardegna" coords="114,238,160,31"
href="sardegna.htm" />
<area shape="poly" alt="Sicilia" coords="304,330,292,354"
href="sicilia.htm" />
<area shape="poly" alt="Calabria" coords="304,283,319,12"
href="calabria.htm" />
</map>
```

Da alcuni test effettuati, attualmente il supporto dei browser per questo codice è pressoché nullo: pertanto, in caso di utilizzo di pagine in formato XHTML 1.1 si consiglia - per le pagine che contengono mappe - di utilizzare XHTML 1.0 Strict.

Nel caso si utilizzino delle mappe immagine lato server al fine di rispettare il requisito 3 è necessario fornire testi alternativi per l'immagine (magari correlati di un elemento longdesc per le mappe più complicate seguire alcuni accorgimenti previsti dal Requisito 8.

### ***Applet e oggetti***

Quando si utilizzano oggetti, come le animazioni Macromedia Flash, è necessario predisporre una versione equivalente testuale per tutti gli utenti che non

posseggono i plug-in necessari alla loro visualizzazione o per gli utenti che a causa di disabilità non possono interagire con tali contenuti. È bene ricordare che il W3C promuove l'utilizzo dell'elemento `<object>`, mentre considera errato l'uso dell'elemento proprietario `<embed>` (necessario per compatibilità con vecchie versioni di Netscape) e deprecato l'uso di `<applet>`. Nell'esempio seguente ci occupiamo dell'inserimento di una animazione Macromedia Flash in una pagina Web.

Il codice standard predefinito da Macromedia per le animazioni Flash è il seguente.

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase=http://download.macromedia.com
/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0
width="400" height="300" id="movie" align="">
<param name="movie" value="movie.swf">
<embed src="movie.swf" quality="high" width="400"
height="300" name="movie" align=""
type="application/x-shockwave-flash"
pluginspage="http://www.macromedia.com/go/getflashplayer">
</object>
```

Per rendere accessibile questo contenuto è necessario modificarlo come indicato di seguito, che consente di presentare agli utenti una versione alternativa (si auspica equivalente) dei contenuti dell'animazione Flash nonché di visualizzare l'oggetto su diversi browser. In mancanza del necessario plug-in, sarà visualizzata un'immagine allo scopo di indicare tale situazione, e verranno fornite le stesse opzioni contenute nell'animazione Flash in formato testuale.

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
width="400" height="300">
<param name="movie" value="movie.swf" />

<ul title="opzioni alternative">
<li><a href="pagina1.htm">Pagina 1</a>
<li><a href="pagina2.htm">Pagina 2</a>
</ul>
<!--[if !IE]> <-->
<object type="application/x-shockwave-flash" data="movie.swf"
width="400" height="300">
<param name="movie" value="movie.swf" />

<ul title="opzioni alternative">
```

```
<li><a href="pagina1.htm">Pagina 1</a>
<li><a href="pagina2.htm">Pagina 2</a>
</ul>
</object>
<!--> <![endif]-->
</object>
```

## ***Frame***

Per i frame è consigliato consultare gli esempi della sezione WCAG 1.0 per il requisito 2 in cui chiaramente si incentiva l'uso dell'attributo title per consentirne l'identificazione. In questo caso, è consigliabile utilizzare pure l'elemento `<noframes>` al fine di garantire un equivalente testuale agli utenti che non hanno accesso ai frame.

## ***Pulsanti***

Per ottenere una grafica accattivante, frequentemente si utilizzano le immagini in sostituzione dell'elemento `<input type="button" />` dei moduli (form). Anche tali elementi, al fine di essere fruibili anche da utenti che utilizzano sistemi di navigazione testuale (lettori di schermo), necessitano dell'attributo `alt`:

```
<form action="submit" method="post">
<input type="image" src="cerca.gif" id="cerca" alt="Cerca" />
</form>
```

In mancanza di tale attributo un utente non vedente non potrà sapere di che pulsante si tratti. È necessario che il testo alternativo del pulsante corrisponda, se presente, al testo contenuto nell'immagine oppure in mancanza dello stesso alla funzionalità prevista per l'immagine. Per esempio, l'immagine `cerca.gif` indicata nell'esempio potrebbe essere una lente d'ingrandimento ed un testo alternativo "lente d'ingrandimento" non sarebbe funzionale all'immagine.

## ***Script***

Quando si parla di script, comunemente ci si riferisce a JavaScript. È bene precisare immediatamente che JavaScript non è Java, con cui frequentemente lo si confonde: Java è stato creato da Sun Microsystems ed è un linguaggio di programmazione semicompilato, con cui è possibile creare applicazioni autonome mentre JavaScript (sviluppato da Netscape Corporation), è un linguaggio di programmazione interpretato, ovvero per poter funzionare deve essere contenuto in una pagina Web ed essere eseguito all'interno di un browser in grado di

interpretarlo. Un programma Java può essere incluso in una pagina Web come oggetto, mentre gli script in JavaScript dipendono per il loro funzionamento dal client che li ospita. Quando il browser rileva codice JavaScript all'interno di un documento HTML, lo esegue e ne visualizza gli eventuali risultati sulla pagina Web. Sul computer che esegue il browser deve quindi essere attivo un interprete JavaScript, in grado di comprendere le istruzioni e di eseguirle. HTML può creare solamente pagine statiche, caratterizzate da una limitata interazione con l'utente. HTML non è in grado di "agire": non può eseguire calcoli matematici, lavorare con le variabili o visualizzare dinamicamente un qualsiasi contenuto. JavaScript rende "reattiva" la pagina: è un linguaggio di programmazione, e quindi permette agli sviluppatori di implementare nelle pagine piccoli programmi, che saranno in grado di svolgere azioni semplici come modificare un elemento grafico quando il puntatore del mouse vi passa sopra ma anche di risolvere complesse formule e procedimenti matematici. È necessario tenere bene a mente che non tutti gli utenti possono fruire di queste possibilità di interazione e predisporre quindi una opportuna sezione `<noscript>` per rendere disponibile un contenuto alternativo. Nell'esempio seguente, lo script richiama un file esterno (`sceltasiti.js`) che consente di raggiungere un sito selezionandolo in un elenco a discesa predefinito: gli utenti che non possono utilizzare tale funzionalità, otterranno lo stesso elenco nella sezione `<noscript>`, che li presenterà come normali collegamenti invece che come voci di un elenco a discesa.

```
<script type="text/javascript" src="sceltasiti.js">
</script>
```

```
<noscript>
<ul>
<li><a href="/">Pagina Iniziale</a></li>
<li><a href="http://www.canale5.com/">Canale 5</a></li>
<li><a href="http://www.italia1.it/">Italia 1</a></li>
</ul>
</noscript>
```

Va comunque posta l'attenzione sul fatto che `<noscript>` è utilizzato dai programmi utente che non supportano gli script o con gli script disabilitati: un utente con tecnologia assistiva potrebbe non fruire di questo contenuto equivalente e, trovandosi davanti ad uno script inaccessibile (ovvero non rispettoso dei Requisiti 15, 16 e 17) potrebbe non poter interagire con il contenuto.

## Arte ASCII ed emoticon

Sul Web è facile incontrare immagini generate con caratteri alfabetici. Vengono chiamate Arte ASCII e sono un residuo storico dei tempi delle BBS, quando i computer erano lenti e con scarse capacità grafiche.

Provate a leggere l'immagine seguente con un sistema di sintesi vocale: il risultato sarà assolutamente incomprensibile.

```

      .-._.-.
    ( 0    0 )
  /   . .   \
  .`._____.'
 /(\           )\
 _/ \ \ \ / / \ _
 .~ ` \ \ / / ` ~.
 {   -. \ v / .- }
 _ _`. \ | | | / .' _ _
 >_   _} | | | { _   _<
 / . - ~ , _-' .^. ` _ , ~ - .\
      \-' | / \ | `-'`

```

Sarà necessario fornire un testo equivalente e la possibilità di saltare la lettura dell'Arte ASCII:

<p> <a href="#ascii-art">salta la ASCII art</a> </p>

```

      .-._.-.
    ( 0   0 )
  /   . .   \
  .'_____.'
 /((           )\
_/_ \ \ \ / / \ _
.~ ` \ \ / / ` ~.
{   -. \ v / .- }
_ _`. \ | | | / .' _ _
>_   _} | | | { _   _<
/. - ~ ,_-'. ^. `-_ , ~ - .\
    \-'|/ \ |`-`

```

<p> <a name="ascii-art">Immagine di una rana<a> </p>

L'Arte ASCII resterà visibile agli utenti normovedenti e allo stesso tempo sarà possibile informare gli utenti non vedenti o con disabilità di tipo cognitivo del contenuto dell'immagine stessa grazie al testo nel link di destinazione del "salto".

Le emoticons, vale a dire le oramai famose faccine che rappresentano le emozioni "digitali", fanno parte della famiglia delle ASCII Art. Il RNIB (Royal National Institute for Blind) ne sconsiglia l'uso

<[http://www.rnib.org.uk/xpedio/groups/public/documents/publicwebsite/public\\_asciiart.hcsp](http://www.rnib.org.uk/xpedio/groups/public/documents/publicwebsite/public_asciiart.hcsp)> mentre il W3C consiglia di utilizzare l'elemento <abbr>:

<p><abbr title="sorriso">:-)</abbr>

Nel caso invece si stia agendo in ambienti come le chat, dove le emoticons sono trasformate automaticamente in immagini, è necessario che l'attributo alt fornisca una chiara spiegazione equivalente e non la semplice conversione in formato testo dello smile:



## Punto di controllo 6.2 (Priorità 1)

### Garantire che all'aggiornamento dei contenuti dinamici vengano aggiornati anche i contenuti equivalenti

Come abbiamo avuto modo di indicare nella motivazione del requisito, in presenza di contenuti dinamici è necessario fornire una versione alternativa, (preferibilmente in formato HTML o testuale) che consenta l'accesso ai contenuti a queste categorie di utenti, nascondendo questa versione a chi invece dispone delle tecnologie necessarie per la normale visualizzazione della pagina.

Utilizzando l'elemento `<object>`, se il browser non supporta quel particolare oggetto sarà reso disponibile il contenuto alternativo, a cascata. Queste versioni dovrebbero restare nascoste per utenti che invece dispongono delle tecnologie per la corretta visualizzazione dei contenuti. Utilizzando l'elemento `<object>` nel caso non sia disponibile il supporto a tale oggetto sarà reso disponibile il contenuto alternativo, il tutto configurabile a cascata.

Per dimostrare le potenzialità dell'elemento `<object>` per l'accessibilità, nell'esempio seguente è ipotizzato l'uso di una applet java per la visualizzazione delle ultime novità.

```
<object classid="java:News.class" width="468" height="60">
<object data="news.mpeg" type="video/mpeg">
<ul>
<li><a href="news1.html">Notizia 1</a></li>
<li><a href="news2.html">Notizia 2</a></li>
<li><a href="news3.html">Notizia 3</a></li>
</ul>
</object>
</object>
```

Innanzitutto come è facile notare si è utilizzato l'elemento `<object>` poiché l'elemento `<applet>` è stato deprecato. Se il visitatore del sito non ha la possibilità di caricare l'applet, sarà caricato un filmato .mpeg alternativo che presenterà le news e consentirà di accedere alle pagine relative. Se l'utente non ha neppure la possibilità di visualizzare i filmati, allora sarà presentata una lista di link (collegamenti), che potranno essere "vestiti" tramite CSS facendo presente che i contenuti dovranno essere fruibili e comprensibili anche senza l'uso dei fogli di stile (requisito 11).

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la barra dell'accessibilità per:

- verificare la presenza di testi alternativi per le immagini (comprese le mappe immagine ed i pulsanti);
- verificare la presenza di contenuti equivalenti per script ed oggetti;
- verificare che i contenuti equivalenti di contenuti dinamici siano aggiornati con la stessa frequenza dei contenuti principali.

L'esperto dovrà quindi controllare i testi degli attributi "alt" e valutare se siano comprensibili e significativi per garantire informazioni. Per effettuare questa verifica potrà utilizzare la barra dell'accessibilità abilitando i testi alternativi in sostituzione delle immagini: se i testi risulteranno comprensibili, allora il sito internet avrà superato il requisito 3.

Tramite la Barra dell'accessibilità è possibile utilizzare:

### **Immagini**

*Elenco delle immagini utilizzate.* Visualizza (in una nuova finestra) una lista delle immagini insieme al corrispondente elemento `<img>` (Punto di controllo 1.1).

*Alterna immagini/testi alternativi.* Rimpiazza tutti gli elementi `<img>` sulla pagina corrente con il contenuto dei corrispondenti attributi `alt`. Se un'immagine è priva di attributo `alt` verrà rimpiazzata con il testo `NoAlt!` (Punto di controllo 1.1).

*Visualizza le immagini.* Visualizza il contenuto dell'elemento `<img>` a fianco a ciascuna immagine (insieme all'attributo `alt` o la scritta `NoAlt!` se l'immagine ne è priva) e crea un margine rosso intorno a ciascuna immagine per renderla chiaramente identificabile (Punto di controllo 1.1).

*Visualizza le mappe immagine.* Mostra gli elementi `<map>` e il contenuto delle aree sensibili suddividendo le mappe immagine lato server dalle mappe immagine lato client (Punto di controllo 1.1).

### **Struttura**

*Name / Title dei frame.* Elenca (in una nuova finestra) una lista delle pagine contenute nel frameset insieme ai rispettivi attributi "name" ed al contenuto dell'attributo "title" (Punto di controllo 1.1).



*Javascript.* Elenca su nuova finestra i javascript presenti nella pagina. L'esperto dovrà quindi verificare la disponibilità di elementi `<noscript>` (Punto di controllo 1.1).

### **Informazioni documento**

*Elenco dei frame.* Visualizza (in una nuova finestra) un elenco degli URL delle pagine inserite nei frame, con il nome dei rispettivi elementi `<frame>`, contenuti nella pagina corrente (Punto di controllo 1.1).

*Identifica Applet / Script.* Visualizza l'elenco delle applet e degli script presenti all'interno della pagina (Punto di controllo 1.1). L'esperto dovrà quindi verificare la presenza di contenuti equivalenti per tali oggetti (Punto di controllo 6.2).

Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 4

Enunciato: Garantire che tutti gli elementi informativi e tutte le funzionalità siano disponibili anche in assenza del particolare colore utilizzato per presentarli nella pagina.

Riferimenti WCAG 1.0: 2.1

Riferimenti Section 508: 1194.22 (c)

### Motivazione

---

I problemi legati alla percezione del colore sono più diffusi di quanto si pensi, ed hanno cause ed effetti diversi: si pensi che studi attuali fanno notare che addirittura una persona su dodici ne soffre. Per veicolare l'informazione il colore viene usato in maniera massiccia anche sul Web, e ciò può rendere difficoltosa o addirittura impossibile la navigazione o l'uso di un software per queste persone. Se inoltre pensiamo al costante innalzamento dell'età media dei navigatori abituali, è facile comprendere come per un designer sia importante conoscere queste problematiche, in modo da evitare non solo l'uso di combinazioni cromatiche che siano di fatto un ostacolo alla corretta fruizione dei contenuti nelle pagine Web da realizzare (requisito 6) ma soprattutto di non affidare esclusivamente ai colori il compito di dare informazioni.

### Implementazione

---

Il requisito chiede di garantire la fornitura di informazioni indipendente da particolari colori. Lo sviluppatore dovrà quindi seguire le semplici raccomandazioni ispirate alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti.

### Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 per fornire allo sviluppatore le informazioni necessarie per uno sviluppo conforme.

## Punto di controllo 2.1 (Priorità 1)

### Garantire che tutta l'informazione veicolata dal colore sia disponibile anche in assenza di colori

Gli utenti con problemi di corretta visione dei colori o chi utilizza periferiche di navigazione in bianco e nero (per esempio, vecchi modelli di palmari) non potrà accedere al sito se forniamo informazioni basandoci esclusivamente sui colori. Per esempio, se si utilizzano i fogli di stile per cambiare il colore dei collegamenti rimuovendone la sottolineatura (il formato predefinito dei collegamenti ipertestuali), gli utenti del sito con le suddette problematiche si troveranno in forte difficoltà a distinguere i collegamenti dal normale testo. Per soddisfare questo punto di controllo quindi è consigliabile rendere chiaramente identificabili le parti di testo che richiedono attenzione da parte dell'utente. Se, per esempio, desideriamo evidenziare i collegamenti ipertestuali, è consigliabile impostare nel foglio di stile il valore `underline` (sottolineatura) della proprietà `text-decoration` e, a ulteriore definizione, usare un carattere con peso maggiore.

```
a {
font-weight: bold;
text-decoration: underline;
color: #000;
background: transparent;
}
```

Spesso invece accade che per questioni "estetiche" viene richiesto allo sviluppatore di eliminare la sottolineatura dei collegamenti ipertestuali e di utilizzare un altro colore per identificare il collegamento attivo. Un utente con disabilità legate ai colori o disabilità cognitive troverà difficoltoso poter identificare i collegamenti e ciò renderà la pagina inaccessibile a tali categorie di utenti.

Questo punto di controllo richiede allo sviluppatore di controllare che tutte le informazioni disponibili nel sito siano indipendenti dai colori: se il sito presenta informazioni sul bilancio comunale e abbiamo evidenziato con una tonalità di rosso le voci negative, gli utenti precedentemente citati non potranno comprendere se una determinata voce sia positiva o negativa. In questo caso, l'indicare chiaramente con segno negativo i valori negativi rappresenterà la soluzione migliore per ogni categoria di utente. In un sito di un ente se si evidenzia con una tonalità di rosso un bando in scadenza, gli utenti con disabilità visive e utenti con tecnologie assistive potrebbero non ottenere tale importante informazione. In questo caso, è consigliato usare elementi di marcatura, come ad esempio `<strong>`.

Spesso, inoltre, si utilizzano formattazioni CSS per identificare, ad esempio, l'area attiva di una particolare sezione utilizzando uno specifico colore. Al fine di rispettare il requisito, è necessario fornire anche informazioni alternative che non siano vincolate al colore.

Un esempio che consente il rispetto del requisito 4 è il seguente:

```
<ul>
<li><a href="m1.asp">Menu 1</a></li>
<li><a href="m2.asp" class="attivo" title="(attivo)">Menu 2</a></li>
<li><a href="m3.asp">Menu 3</a></li>
</ul>
```

In questo modo lo sviluppatore può sbizzarrirsi assegnando diversi colori alle aree selezionate tramite la classe "attivo" ma, in mancanza del foglio di stile, dei colori o a causa della presenza di particolari colori l'utente potrà sempre ottenere informazioni sul collegamento attivo tramite l'attributo "title".

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la Barra dell'accessibilità per:

- verificare la presenza di informazioni legate a specifici colori.

Tramite la Barra dell'accessibilità è possibile utilizzare:

### Immagini

*Attiva/Disattiva i CSS. Attiva/disattiva i fogli di stile esterni* (Punto di controllo 2.1).

*Disattiva i CSS inline.* Rimuove l'attributo `style` (se presente) da tutti gli elementi della pagina al fine di consentire l'identificazione di eventuali link (Punto di controllo 2.1).

*Visualizza foglio di stile.* Consente di verificare se all'interno del foglio di stile sono stati utilizzate in modo corretto le proprietà per gli stili dei collegamenti ipertestuali (Punto di controllo 2.1).

## **Colori**

*Scala di grigi.* Visualizza il contenuto della pagina corrente in scala di grigi in modo da identificare visivamente eventuali problematiche di accessibilità legate a specifici colori (Punto di controllo 2.1).

L'esperto dovrà quindi controllare, ad esempio, che i collegamenti ipertestuali siano forniti della classica sottolineatura e che siano chiaramente identificabili. Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 5

Enunciato: Evitare oggetti e scritte lampeggianti o in movimento le cui frequenze di intermittenza possano provocare disturbi da epilessia fotosensibile o disturbi della concentrazione, ovvero possano causare il malfunzionamento delle tecnologie assistive utilizzate; qualora esigenze informative richiedano comunque il loro utilizzo, avvertire l'utente del possibile rischio prima di presentarli e predisporre metodi che consentano di evitare tali elementi.

Riferimenti WCAG 1.0: 7.1, 7.2, 7.3

Riferimenti Section 508: 1194.22 (j)

### Motivazione

---

Il requisito intende riferirsi ai problemi legati all'uso di oggetti in movimento (testo ed immagini) che possono presentare le seguenti problematiche:

- epilessie fotosensibili: in questo caso si richiede di evitare frequenze di lampeggio comprese tra 2 Hz e 55 Hz;
- disturbi della concentrazione: oggetti in movimento, seppur realizzati nel rispetto del punto precedente, possono generare disturbi della concentrazione. In questo caso, in linea di massima, i problemi nascono sulla base della numerosità e della posizione nella pagina degli oggetti in movimento;
- malfunzionamento delle tecnologie assistive: quando si utilizzano scritte scorrevoli o pagine che si autoaggiornano le tecnologie assistive hanno difficoltà a renderne il contenuto all'utente;
- esigenze informative: ci si riferisce al caso di pagine che mostrino esempi di cattivo utilizzo di oggetti in movimento, per spiegare ad esempio il significato del presente requisito  
<<http://ncam.wgbh.org/richmedia/examples/127.html>>, oppure di pagine che mostrino esempi di arte visiva con oggetti in movimento.

### Implementazione

---

Il requisito chiede di garantire la fornitura di informazioni che non impediscano ad alcune categorie di utenti di poter fruire dei contenuti della pagina web. Lo

sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti.

## Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

### **Punto di controllo 7.1 (Priorità 1). Fino a quando i programmi utente non permetteranno agli utenti di controllare lo sfarfallio, evitare di far sfarfallare lo schermo**

Gli utenti con epilessia fotosensibile possono avere dei problemi con testi o immagini ad intermittenza con intervalli compresi tra 4 e 59 flash per secondo (Hertz). Uno schermo con testi e immagini in movimento o lampeggianti può causare delle crisi epilettiche agli utilizzatori che soffrono di epilessia fotosensibile perciò i professionisti che curano i contenuti dovrebbero evitare di far lampeggiare o scorrere velocemente contenuti sullo schermo. Un attacco epilettico può essere provocato da sfarfallii o lampeggiamenti nella fascia di valori che va da 4 a 59 lampeggiamenti al secondo (Hertz), con un picco di sensibilità a 20 lampeggiamenti al secondo, oppure da veloci passaggi dall'oscurità alla luce (come luci intermittenti, banner pubblicitari che segnalano novità oppure offerte speciali). Tali valori nella normativa americana Section 508 sono definiti invece tra 2 e 55 Hz e la nostra normativa ha deciso di recepire tale indicazione.

Effetti come l'effetto di transizione di pagina di Microsoft Internet Explorer, spesso utilizzato dagli sviluppatori, devono quindi essere controllati. L'effetto di transizione viene effettuato all'ingresso o all'uscita dalla pagina:

```
<meta http-equiv="Page-Enter"
content="revealTrans(duration=2, Transition=6)" />
```

Se la transizione è troppo lenta, i visitatori troveranno noiosa la navigazione, mentre se è troppo veloce può causare sfarfallii e lampeggiamenti pericolosi per gli utenti predisposti ad attacchi di epilessia. Questi effetti di transazione di pagina, non avendo degli scopi funzionali, è preferibile non utilizzarli. Lo stesso problema può essere riscontrabile nelle immagini animate, nei filmati e specialmente nei banner pubblicitari, quindi attenzione! Nella valutazione del requisito vedremo come un italiano ha prodotto il primo strumento al mondo in

grado di valutare se delle immagini gif animate rientrano negli intervalli critici previsti dalla normativa.

## **Punto di controllo 7.2 (Priorità 2)**

### **Fino a quando i programmi utente non permetteranno agli utenti di controllare il lampeggiamento, evitare di far lampeggiare il contenuto**

Alcune tecnologie assistive, come i lettori di schermo riscontrano difficoltà di interazione con i testi lampeggianti: in alcuni casi essi si bloccano sul campo di testo, in altri lo leggono a ripetizione o addirittura il sistema diventa instabile.

È importante inoltre sottolineare che l'utilizzo dell'elemento `<blink>` (lampeggiamento) non consente la validazione del codice secondo le raccomandazioni W3C HTML 4.x / XHTML 1.x in quanto è una definizione creata da Netscape per pURI effetti visivi. E' comunque disponibile un valore della proprietà CSS `text-decoration (blink)` che consente di ottenere lo stesso effetto di lampeggiamento ma per il superamento del requisito ne sconsiglio l'utilizzo:

```
.lampeggiante
{
...
text-decoration: blink;
}
```

L'utilizzo di testi lampeggianti distrae i visitatori dal resto del contenuto della pagina: è pressoché inutile utilizzare l'elemento `<blink>` per attirare l'attenzione in quanto l'utente di fatto è già presente all'interno della vostra pagina Web e, verosimilmente, desidera fruirne i contenuti senza condizionamenti. In ogni caso, se non viene offerta la possibilità all'utente di disabilitare il lampeggiamento, l'uso di tale elemento (o del valore CSS `blink`) non consente il superamento del requisito. Se si ritiene assolutamente necessario indirizzare l'attenzione di un visitatore su un argomento importante, è possibile utilizzare elementi di enfasi (`<em>`) impostando per essi una classe nel foglio di stile:

```
<em>Contenuto da evidenziare</em>
```

È necessario precisare che l'uso di elementi di enfasi va moderato, in quanto nella modalità di visualizzazione predefinita è definito un carattere di tipo corsivo/italico che può causare problemi di lettura (affaticamento) ad alcune categorie di utenti e che, se si trattasse esclusivamente di utilizzo decorativo, è necessario utilizzare l'elemento `<i>` anziché l'elemento `<em>`.



## Punto di controllo 7.3 (Priorità 2)

### Fino a quando i programmi utente non permetteranno agli utenti di bloccare il contenuto in movimento, evitare il movimento nelle pagine

Come per i testi lampeggianti, anche i testi scorrevoli possono creare dei problemi alle tecnologie assistive, in particolare ai lettori di schermo. In presenza di una scritta scorrevole, un utente con disabilità visiva o cognitiva può riscontrare dei problemi di lettura di testi scorrevoli e quindi problemi di fruibilità dei contenuti. Un testo scorrevole inoltre distrae l'attenzione dai contenuti della pagina. Per la stessa motivazione è necessario utilizzare con parsimonia le GIF animate ed in generale qualsiasi tipo di animazione (filmati Macromedia Flash, effetti DHTML, ecc.) in quanto l'uso sconsiderato di animazioni, oltre a distrarre i visitatori del sito, in alcuni casi (vedi Punto di controllo 7.1) può causare problemi di epilessia fotosensibile. L'elemento <marquee> non è presente nelle raccomandazioni W3C (è stato introdotto da Microsoft in Internet Explorer come "risposta" all'elemento <blink> durante il periodo della "guerra dei browser") e pertanto non deve essere utilizzato se si desidera rispettare la conformità del codice e questo punto di controllo: se si desidera attirare l'attenzione dell'utente su un particolare testo, è possibile farlo utilizzando i fogli di stile con gli elementi di enfasi. Spesso l'effetto di movimento dei testi viene generato da script o applet: in tal caso è necessario fornire un meccanismo all'interno di uno script o applet per permettere agli utenti di bloccare il movimento o gli aggiornamenti. Il fatto di usare i fogli di stile insieme con gli script per creare il movimento consente agli utenti di disabilitare oppure tenere sotto controllo gli effetti con maggiore facilità.

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la Barra dell'accessibilità per:

- verificare la presenza di scritte animate o lampeggianti;
- verificare la presenza immagini animate (gif) misurandone la frequenza.

Tramite la Barra dell'accessibilità è possibile utilizzare:

### Immagini

*GIF Flicker Test.* Analizza le immagini .gif presenti nella pagina generando un report di conformità al requisito (Punto di controllo 7.1).

## **Struttura**

*Visualizza altri elementi.* Attiva una micro applicazione in Javascript che identifica le istanze sulla pagina corrente dell'elemento inserito, come ad esempio <blink> (Punto di controllo 7.2) e <marquee> (Punto di controllo 7.3).

## **Informazioni documento**

*Informazioni metadata.* Visualizza (in una nuova finestra) l'elenco di tutti gli elementi meta ed il relativo contenuto della pagina corrente. (Punto di controllo 7.1).

*Identifica Applet / Script.* Visualizza l'elenco delle applet e degli script presenti all'interno della pagina. (Punto di controllo 1.1) L'esperto dovrà quindi verificare la presenza di contenuti equivalenti per tali oggetti (Punti di controllo 7.1, 7.2 e 7.3).

Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 6

Enunciato: Garantire che siano sempre distinguibili il contenuto informativo (foreground) e lo sfondo (background), ricorrendo a un sufficiente contrasto (nel caso del testo) o a differenti livelli sonori (in caso di parlato con sottofondo musicale); evitare di presentare testi in forma di immagini; ove non sia possibile, ricorrere agli stessi criteri di distinguibilità indicati in precedenza.

Riferimenti WCAG 1.0: 2.2

Riferimenti Section 508: non presente

### Motivazione

---

Alcune disabilità visive non consentono una corretta visualizzazione dei colori rendendo quindi di difficile comprensione alcuni accostamenti cromatici.

Ipotizzando di partire da un'immagine originale contenente dei fiori di colore rosa immersi in un insieme di foglie verdi si potranno avere le seguenti variazioni cromatiche:

- un utente affetto da deutranopia avrà una visione del fiore di colore grigio-azzurro anziché rosa mentre vedrà le foglie di colore verde oliva anziché il classico colore verde prato.
- un utente affetto da protanopia avrà una visione del fiore simile all'utente con deutranopia mentre vedrà le foglie di un colore tendente al giallo senape anziché il classico colore verde prato.
- un utente affetto da tritanopia avrà una visione del fiore di colore rosa chiaro mentre vedrà le foglie di un colore tendente all'azzurro.

Un strumento che consente di simulare le disabilità del colore è un tool prodotto da Fujitsu, ColorDoctor

<<http://design.fujitsu.com/en/universal/assistance/colordocor/>> (il prodotto è operativo esclusivamente in ambiente Microsoft Windows e richiede il .net Framework 1.1). All'esecuzione del programma, verrà presentato un browser in cui l'utente potrà digitare l'indirizzo della pagina web ed effettuare una visualizzazione di come un utente con disabilità visiva legata al colore interpreta i contenuti di una pagina web.

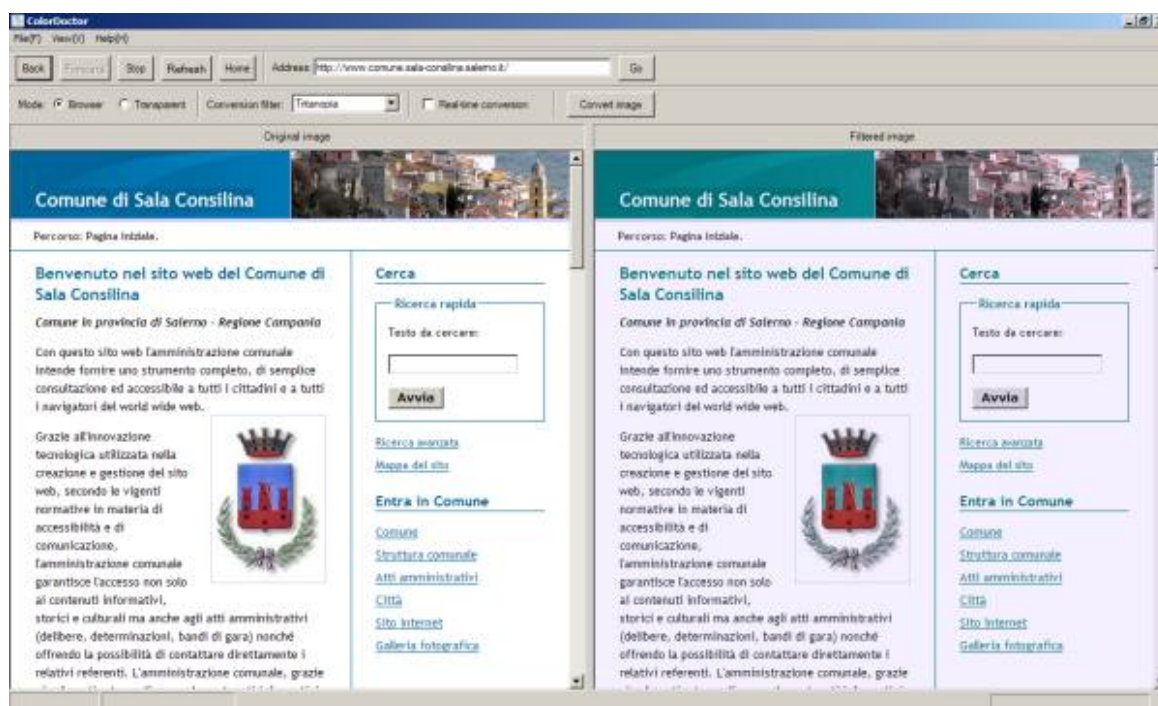


Figura 4.2 Color Doctor.

Pensiamo quindi al caso in cui si forniscano informazioni basandosi su abbinamenti di colore che alcune categorie di utenti possono confondere tra di loro (ad esempio i soggetti affetti da daltonismo) oppure soggetti che possono non vedere un determinato colore oppure ad utenti che lavorando in condizioni di non ottimale visibilità (ad esempio, con un notebook all'aperto) non riescono a vedere chiaramente alcune tonalità di colore. Il requisito, paradossalmente, è maggiormente dedicato agli utenti normovedenti in quanto utenti con disabilità che richiedono contrasti elevati solitamente utilizzano dei fogli di stile personalizzati per la navigazione dei siti internet.

Il requisito richiede inoltre di garantire lo stesso beneficio per i contenuti audio, ovvero in presenza di dialoghi ed effetti sonori, è necessario garantire la comprensione dei dialoghi.

## Implementazione

Il requisito chiede di garantire la fornitura di informazioni che non impediscano ad alcune categorie di utenti di poter fruire dei contenuti della pagina web. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti. Sarà quindi richiesto di sviluppare testi ed immagini in modo che il colore

principale sia distinguibile dallo sfondo secondo un particolare algoritmo pubblicato nel decreto all'interno della metodologia della verifica tecnica (Punto 2 Lettera d). L'algoritmo non è però utilizzabile per la verifica della comprensione del parlato rispetto al sonoro di un contenuto audio/video.

## Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

### **Punto di controllo 2.2 (Priorità 2 per le immagini, Priorità 3 per i testi)**

#### **Garantire che le combinazioni di colori tra il primo piano e lo sfondo forniscano un sufficiente contrasto se osservati da qualcuno con deficit percettivi del colore o quando visualizzati su un monitor in bianco e nero**

Questo punto di controllo nelle WCAG 1.0 riguarda la Priorità 2 per le immagini e la Priorità 3 per il testo, e richiede che il contrasto tra il colore di sfondo e il colore in primo piano sia sufficiente a evitare problemi di lettura ai visitatori del sito. Il paradosso che viene sempre rappresentato nei seminari dedicati all'accessibilità è relativo alla priorità 3 per il testo: è possibile creare un sito rispettoso delle priorità 2 e con testi neri su fondo nero per ottenere la conformità al livello "AA", ma rendendo chiaramente il sito inaccessibile ad un utente normovedente. Per la normativa italiana, invece tale requisito è obbligatorio.

Due colori permettono una buona visibilità se la differenza di luminosità e di colore tra i due è superiore a una determinata serie di valori. Il W3C ha determinato un algoritmo per calcolare il contrasto  [<http://www.w3.org/TR/AERT#color-contrast >](http://www.w3.org/TR/AERT#color-contrast) che sarà utilizzato in questa verifica grazie agli strumenti di controllo inseriti nel CD-ROM allegato al libro. L'algoritmo, sviluppato da Chris Ridpath, Jutta Treviranus, Patrice L. (Tamar) Weiss, è ancora in fase di sviluppo ed è possibile che in futuro venga modificato. Ha comunque avuto una serie di valutazioni e test con utenti, come comprovato dalla pagina informativa predisposta da Ridpath  [<http://checker.atrc.utoronto.ca/docs/WebPageColors.html >](http://checker.atrc.utoronto.ca/docs/WebPageColors.html).

La seguente formula è suggerita dal World Wide Web Consortium (W3C) e serve a calcolare la luminosità di un colore RGB.

```
((valore Rosso X 299) + (valore Verde X 587) + (valore Blue X 114)) /  
1000
```

La differenza tra la luminosità dello sfondo e la luminosità del colore principale deve essere maggiore di 125.

La seguente formula del W3C consente invece di calcolare la differenza tra due colori.

```
(massimo (valore Rosso 1, valore Rosso 2) - minimo (valore Rosso 1,  
valore Rosso 2)) + (massimo (valore Verde 1, valore Verde 2) - minimo  
(valore Verde 1, valore Verde 2)) + (massimo (valore Blue 1, valore  
Blue 2) - minimo (valore Blue 1, valore Blue 2))
```

La differenza tra colore di sfondo e colore principale deve essere uguale o superiore a 500. Hewlett Packard (HP) fornisce uno strumento di verifica del contrasto fra colori che utilizza gli algoritmi del W3C, ma fissa il livello minimo di differenza di colore a 400, il che aumenta il numero di combinazioni sfondo/primo piano considerate accettabili. È comunque facile immaginare che colori chiari su fondo chiaro o colori scURI su fondo scuro non rappresentano la scelta opportuna, così come va ricordato quanto descritto nell'enunciato precedente, ossia che esistono molte disabilità visive legate ai colori.

A ulteriore supporto inoltre è disponibile una trasposizione della tabella dei 216 colori "sicuri" per alcune delle più diffuse disabilità relative ai colori <<http://more.btexact.com/people/rigden/colours/colours1.htm>>. Poiché esistono moltissime varianti delle disabilità legate al colore, risulta impossibile rendere un sito adatto a tutti gli utenti. Per questo motivo si consiglia di definire fogli di stile (CSS) alternativi che permettano all'utente di personalizzare a proprio piacimento o esigenza i colori del sito stesso. Per ottenere questo risultato, si utilizzerà l'elemento <link> nella sezione <head> della pagina per richiamare i fogli di stile. Il foglio di stile principale avrà il valore stylesheet per l'attributo rel, mentre gli altri fogli di stile saranno definiti dal valore alternate stylesheet.

```
<link href="main.css" rel="stylesheet" type="text/css"  
title="Standard" />  
<link href="1.css" rel="alternate stylesheet" type="text/css"  
title="Alto contrasto" />  
<link href="2.css" rel="alternate stylesheet" type="text/css"  
title="Tema Sahara" />
```

Il visitatore del sito potrà selezionare il foglio di stile desiderato nell'elenco che gli verrà proposto, creato utilizzando l'attributo title.

Per i browser che non supportano questa opzione, è possibile utilizzare codice JavaScript. Joe Clark ha predisposto un elenco di esempi pratici <<http://joeclark.org/f2f/recommendations.html?GL#SWITCHER-EXAMPLES>> per l'utilizzo di tali tecniche con differenti modalità di utilizzo di diversi fogli di stile.

È chiaro che il requisito riguarda non solo i testi, la cui formattazione dovrebbe avvenire tramite fogli di stile, ma anche le immagini.

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la barra dell'accessibilità per:

- verificare il contrasto dei colori nel foglio di stile;
- verificare il contrasto dei colori per le immagini e gli oggetti presenti nella pagina.

Tramite la Barra dell'accessibilità è possibile utilizzare:

### **Colori**

*Colour Contrast Analyzer.* Esegue un'applicazione che consente di verificare il rispetto degli algoritmi per qualsiasi contenuto presente nella pagina, immagini comprese. (Punto di controllo 2.2).

### **Strumenti**

*Juicy Studio Tools / CSS Accessibility Analyzer.* Consente di analizzare gli elementi all'interno di un foglio di stile per verificare il rispetto degli algoritmi (Punto di controllo 2.2).

*Simulazioni.* Consente di accedere a una serie di funzionalità per emulare la visualizzazione della pagina secondo diverse disabilità visive (Punto di controllo 2.2).

Uno strumento utile per la valutazione del requisito è un'applicazione creata dagli stessi autori della barra dell'accessibilità, Colour Contrast Analyzer <[http://www.nils.org.au/ais/web/resources/contrast\\_analyser/index.html](http://www.nils.org.au/ais/web/resources/contrast_analyser/index.html)>, disponibile all'interno alla barra dell'accessibilità. L'applicazione consente di effettuare l'analisi della differenza tra contenuto principale e sfondo e della luminosità per testi ed immagini. Per avviare l'analisi basta selezionare l'icona del contagocce, posizionarla sopra il contenuto di cui si desidera acquisire il codice di

colore il valore relativo sarà importato nell'applicazione. Con la stessa modalità è necessario selezionare il colore dello sfondo ed automaticamente l'applicazione visualizzerà se i colori sono utilizzabili o meno.

Nel caso siano presenti dei contenuti audio, l'esperto dovrà valutare il grado di comprensione verificando - ove possibile - se l'audio consente l'esclusione delle tracce relative al sonoro rispetto al parlato. Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.



Figura 4.3 Color Contrast Analyzer.

[Torna a Elenco dei requisiti](#)



## Requisito 7

Enunciato: Utilizzare mappe immagine sensibili di tipo lato client piuttosto che lato server, salvo il caso in cui le zone sensibili non possano essere definite con una delle forme geometriche predefinite indicate nella DTD adottata.

Riferimenti WCAG 1.0: 9.1

Riferimenti Section 508: 1194.22 (f)

### Motivazione

---

Abbiamo già visto nel requisito 3 come le mappe immagine necessitino di testi alternativi per le selezioni di tipo grafico al fine di garantire la possibilità di utilizzare i contenuti della mappa anche ad utenti dotati di tecnologie assistive o di configurazioni particolari. La mappa immagine su lato server è una tecnologia "vecchia". In pratica, quando l'utente clicca su un punto di una mappa le coordinate del mouse (x,y) vengono passate all'applicazione che controlla se tale area è all'interno di un'area sensibile e - in caso positivo - esegue l'azione (solitamente apertura di collegamenti). Siccome tutte le informazioni sono gestite sul lato server di fatto risulta impossibile fornire alternative testuali.

La mappa immagine sul lato client invece è quella che utilizziamo normalmente. In essa sono definiti alcuni oggetti con forme fisse:

- `default`: specifica l'intera immagine sensibile.
- `rect`: definisce un'area rettangolare.
- `circle`: definisce un'area circolare.
- `poly`: definisce un'area poligonare.

Le mappe immagine lato client, supportate ormai da pressoché tutti i browser, sono maggiormente adatte alla modalità di navigazione testuale di quanto lo fossero le progenitrici mappe immagine lato server, a condizione che gli elementi area che le compongono siano dotati del relativo testo alt.

### Implementazione

---

Il requisito chiede di garantire la fornitura di informazioni che non impediscano ad alcune categorie di utenti di poter fruire dei contenuti della pagina web. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano

alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti. Sarà quindi richiesto di verificare se le funzionalità predisposte tramite mappa immagine lato server possono essere rese fruibili anche tramite mappa immagine lato client: ove ciò non fosse possibile, è necessario implementare mappe sensibili su lato server rispettose del requisito 8.

## Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

### **Punto di controllo 9.1 (Priorità 1)**

#### **Fornire mappe immagine sul lato client invece di mappe immagine sul lato server, con l'eccezione dei casi nei quali le zone non possono essere definite con una forma geometrica valida**

Le mappe immagine su lato server vengono utilizzate prevalentemente per applicazioni in cui si richiede una costante interazione tra contenuto Web e server. Un esempio può essere un'applicazione di rappresentazione del piano regolatore di un comune o altra applicazione GIS in cui l'utente deve posizionare il cursore del mouse in un determinato punto di una mappa creata dinamicamente secondo le selezioni operate dall'utente. L'uso più comune che attualmente si riscontra di tali applicazioni è legato ai servizi di consultazione pubblica dei numeri telefonici che forniscono la possibilità di localizzare un determinato utente nella mappa di una città consentendo inoltre di creare dei percorsi e/o di fornire altre informazioni correlate al punto selezionato dall'utente.

È chiaro che, utilizzando questo sistema e senza seguire le indicazioni del requisito 8, la mappa immagine sul lato server non sarà accessibile agli utenti che utilizzano browser testuali o che possono operare esclusivamente tramite tastiera o con tecnologie assistive che non emulano il movimento e le funzionalità del mouse.

Le sole ragioni che potrebbero portare a preferire le mappe immagine sul lato server sono quindi legate alla complessità della rappresentazione tramite le normali figure geometriche previste per le mappe immagine lato client.

Un ulteriore esempio di necessità delle mappe immagine sul lato server può essere il caso in cui le aree sulla mappa sono volutamente mantenute "segrete": per esempio, la mappa di una caccia al tesoro on-line: le destinazioni dei

collegamenti non possono essere rese pubbliche. Le immagini sensibili lato server risultano molto utili nel caso di mappe con informazioni di tipo geografico dove ogni punto è attivo e quindi raggiungibile da un evento tramite mouse: sviluppare lo stesso risultato con una immagine sensibile lato client risulterebbe molto complicato e di fatto ingestibile dagli sviluppatori.

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la Barra dell'accessibilità per:

- verificare la presenza di mappe immagine sensibili lato server.

Tramite la Barra dell'accessibilità è possibile utilizzare:

### **Immagini**

*Visualizza le mappe immagine.* Visualizza gli elementi `<map>` e il contenuto delle aree sensibili suddividendo le mappe immagine sensibili lato server dalle mappe immagine sensibili lato client (Punto di controllo 9.1).

Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 8

Enunciato: In caso di utilizzo di mappe immagine lato server, fornire i collegamenti di testo alternativi necessari per ottenere tutte le informazioni o i servizi raggiungibili interagendo direttamente con la mappa.

Riferimenti WCAG 1.0: 1.2

Riferimenti Section 508: 1194.22 (e)

### Motivazione

---

Abbiamo visto nei requisiti precedenti come sia necessario fornire informazioni testuali alternative per immagini ed oggetti. Nel caso delle mappe immagini su lato client la presenza di testi alternativi per le singole aree sensibili consente di poter accederne ai contenuti.

In presenza di mappe immagine lato server, la maggior parte del codice si trova sul server Web. Al clic sull'area sensibile, il browser non deve fare altro che concatenare le coordinate dei pixel dell'area all'URL previsto e inviarle al server, a cui tocca tutto il resto del lavoro. Questo chiaramente rende difficile la fruibilità di tali mappe ad un utente non vedente, ad un utente con disabilità motorie, cognitive e ad utenti che non utilizzano il mouse (es: utenti con computer palmari): se non vengono forniti degli equivalenti testuali gestibili tramite tastiera per ogni area definita come cliccabile.

### Implementazione

---

Il requisito chiede di garantire la fornitura di informazioni che non impediscano ad alcune categorie di utenti di poter fruire dei contenuti della pagina web. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti. Sarà quindi richiesto di verificare se le funzionalità predisposte tramite mappa immagine lato server possono essere rese fruibili anche ad utenti non vedenti, utenti con disabilità motorie e/o cognitive ed utenti che non utilizzano il mouse. Nel caso sia indispensabile usare una mappa immagine lato server, è necessario utilizzare collegamenti di tipo testuale che consentano la fruibilità anche agli utenti che utilizzano la tastiera o tecnologie di input alternative che non consentono di interagire con la mappa immagine non rendendo quindi accessibili le destinazioni dei collegamenti.

## Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

### **Punto di controllo 1.2 (Priorità 1)**

#### **Fornire collegamenti di testo ridondanti per ogni zona attiva di una mappa immagine sul lato server**

In presenza di mappe immagine lato server, la maggior parte del codice si trova sul server Web. Al clic sull'area sensibile, il browser non deve fare altro che concatenare le coordinate dei pixel dell'area all'URL previsto e inviarle al server, a cui tocca tutto il resto del lavoro. Questo chiaramente rende difficile ad un utente non vedente la fruibilità di tali mappe se non vengono forniti degli equivalenti testuali gestibili tramite tastiera per ogni area definita come cliccabile.

Per creare una mappa immagine lato server che rispetti il requisito 3, sono necessari i seguenti elementi:

- l'immagine deve essere in un formato supportato (gif, jpeg, png);
- il linguaggio di marcatura deve essere conforme al requisito 1;
- un file di mappa immagine che definisca l'azione delle aree sensibili;
- una serie di collegamenti ipertestuali equivalenti alle aree sensibili, oppure una funzionalità equivalente al contenuto dinamico generato.

Il codice generato si presenterà del tutto simile al seguente.

```
<p>
<a href="http://www.miosito.com/generamappa.asp">

</a>
</p>
<ul>
<li><a href="inizio.htm">Punto di Partenza</a></li>
<li><a href="ristoro.htm">Punto di Ristoro</a></li>
<li><a href="fine.htm">Punto di Arrivo</a></li>
</ul>
```

L'elemento `<img>` necessita quantomeno dell'attributo `alt` ed eventualmente dell'attributo `longdesc`: in questo caso, per esempio, l'attributo `longdesc`

potrebbe essere necessario se la mappa descrive con precisione un percorso. L'utente dotato di lettore di schermo otterrà quindi un risultato come il seguente:

```
Immagine. Mappa della gara
Collegamento. Punto di partenza
Collegamento. Punto di ristoro
Collegamento. Punto di arrivo
```

Una versione testuale dei collegamenti è necessaria, poiché la mappa immagine lato server invia al browser le coordinate all'interno delle quali l'utente può azionare il proprio mouse: gli utenti che non ne sono dotati o non possono utilizzarlo potranno fruire dei contenuti grazie alla versione alternativa, in questo esempio i link testuali che consentono di aprire il collegamento relativo al punto di partenza e al punto di arrivo della mappa. I testi equivalenti di supporto per le mappe immagini lato server risultano essenziali non solo ad utenti non vedenti ma anche ad utenti con disabilità di tipo motorio e/o di tipo cognitivo.

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la barra dell'accessibilità per:

- verificare la presenza di mappe immagine sensibili lato server;
- verificare la presenza di collegamenti ipertestuali equivalenti alle zone sensibili della mappa immagine sensibile lato server.

Tramite la Barra dell'accessibilità è possibile utilizzare:

### **Immagini**

*Visualizza le mappe immagine.* Visualizza gli elementi <map> e il contenuto delle aree sensibili suddividendo le mappe immagine sensibili lato server dalle mappe immagine sensibili lato client (Punto di controllo 9.1). Sarà quindi compito dell'esperto, in presenza di mappe immagini sensibili lato server, verificare all'interno della pagina web la presenza di contenuti collegamenti ipertestuali equivalenti alle aree sensibili (Punto di controllo 1.2).

Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 9

Enunciato: Per le tabelle dati usare gli elementi (marcatori) e gli attributi previsti dalla DTD adottata per descrivere i contenuti e identificare le intestazioni di righe e colonne.

Riferimenti WCAG 1.0: 5.1, 5.5, 5.6

Riferimenti Section 508: 1194.22 (g)

### Motivazione

---

Il requisito riguarda esclusivamente le tabelle dati, ossia tabelle non utilizzate per impaginazione ma che contengono informazioni in forma tabellare. Gli sviluppatori spesso utilizzano erroneamente attributi ed elementi previsti per le tabelle dati: un esempio è l'uso dell'elemento `<th>` per scrivere dei testi in grassetto).

In altri casi di necessità, invece, tali attributi non sono utilizzati o sono utilizzati in modo non corretto: a titolo di esempio, si riscontra la mancanza di utilizzo degli elementi ed attributi che consentono agli utenti non vedenti di poter accedere ai contenuti di una tabella dati. La creazione di una tabella dati, infatti, va curata garantendo la possibilità agli utenti di navigarne i contenuti consentendo di collegare una determinata cella di dati (di una colonna o riga) ad una cella di intestazione (di colonna o riga), un po' come avviene nel gioco della battaglia navale. Il requisito chiede quindi di garantire la possibilità di lettura dei contenuti della tabella in modo comprensibile a tutti gli utenti.

### Implementazione

---

Il requisito chiede di garantire la fornitura di informazioni che non impediscano ad alcune categorie di utenti di poter fruire dei contenuti delle tabelle dati contenute in una pagina web. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti. Sarà quindi richiesto di verificare l'utilizzo di elementi ed attributi in modo conforme alle specifiche e soprattutto l'accessibilità dei dati contenuti nelle tabelle.

## Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

### **Punto di controllo 5.1 (Priorità 1)**

#### **Per le TABELLE DATI, identificare le intestazioni per righe e colonne**

Questo punto di controllo per il Livello 1 informa lo sviluppatore che, nel caso vengano utilizzate delle tabelle per presentare dei dati, egli deve identificare in modo chiaro le celle contenenti i dati dalle intestazioni di righe e colonne. Pertanto, le tabelle dati richiedono l'uso degli elementi `<tr>`, `<td>`, `<th>` e `<caption>` e dei relativi attributi.

L'elemento `<caption>` identifica il titolo della tabella. È posizionato solitamente all'inizio della tabella ed è personalizzabile graficamente tramite fogli di stile.

L'elemento `<tr>` identifica una riga di una tabella e può contenere elementi `<th>` ed elementi `<td>`. La sua modalità di visualizzazione può essere personalizzata tramite fogli di stile.

L'elemento `<th>` identifica una cella di intestazione, ovvero un punto di riferimento per la lettura dei contenuti dei dati tabellari ed è contenuto all'interno dell'elemento `<tr>`. Solitamente tali celle vengono evidenziate dal browser con testo in grassetto rispetto all'impostazione predefinita del carattere (ove tale impostazione non viene modificata tramite fogli di stile).

L'elemento `<td>` identifica una cella contenente dati ed è contenuto all'interno dell'elemento `<tr>`. La sua modalità di visualizzazione può essere personalizzata tramite fogli di stile.

Di seguito è rappresentato un esempio di tabella dati semplice. Come si noterà l'elemento `<table>` contiene l'attributo "summary" che, come vedremo nell'analisi del presente requisito, è richiesto per tutte le tipologie di tabelle.

```
<table border="1" summary="Sommaro tabella...">
<caption>Esempio di tabella dati</caption>
<tr>
<td></td>
<th>Intestazione Colonna 1</th>
<th>Intestazione Colonna 2</th>
```



```

</tr>
<tr>
<th>Intestazione Riga 1</th>
<td>Col. 1 Riga 1</td>
<td>Col. 1 Riga 2</td>
</tr>
<tr>
<th>Intestazione Riga 2</th>
<td>Col. 2 Riga 1</td>
<td>Col. 2 Riga 2</td>
</tr>
</table>

```

Nel nostro caso ho utilizzato l'attributo "border" solamente per rendere visibili le dimensioni della tabella nell'esempio senza quindi utilizzare i CSS collegati: in realtà, anche il bordo si deve comunque controllare con i CSS.

Il codice di esempio precedente in un browser genererà una tabella come mostrato nella figura successiva: sarà compito dei fogli di stile e di un corretto utilizzo del selettore CSS "class" e dell'attributo HTML "style" il "vestire" graficamente la tabella. Vorrei comunque sconsigliare l'uso dell'attributo style in quanto non presente nella DTD XHTML 1.1 e successive: per migliorare l'aspetto della tabella si dovranno quindi utilizzare i fogli di stile utilizzando l'attributo HTML "class".

Esempio di tabella dati

	<b>Intestazione Colonna 1</b>	<b>Intestazione Colonna 2</b>
<b>Intestazione Riga 1</b>	Col. 1 Riga 1	Col. 1 Riga 2
<b>Intestazione Riga 2</b>	Col. 2 Riga 1	Col. 2 Riga 2

**Figura 4.4** Tabella di esempio.

Nel caso di tabelle con più righe o colonne, come informare il programma di lettura dello schermo del corretto ordine di lettura dei contenuti? Un aiuto ci viene fornito dall'attributo headers, che specifica l'elenco delle intestazioni collegate a una cella contenente dati.

Per utilizzare questa funzionalità, ogni cella di intestazione deve aver valorizzato l'attributo id. Si riporta in versione italiana un esempio tratto dal sito W3C, che prevede la creazione di una tabella contenente i nomi di due persone (nel nostro caso, i due onorevoli firmatari del primo disegno di legge sull'accessibilità

presentato da IWA/HWG) e le informazioni relative alle preferenze personali riguardo il caffè: quantità, tipo e se il caffè viene zuccherato o meno.

Caffè consumato da ogni onorevole

Nome	Tazze	Tipo di caffè	Zucchero?
A. Palmieri	10	Espresso	No
C. Campa	5	Decaffeinato	Sì

Dalla tabella è chiaro che per consentire una corretta lettura dei contenuti a un utente che utilizza un lettore dello schermo è necessario che la tabella sia strutturata in modo da garantire la comprensione delle informazioni. Nel nostro caso, il lettore dovrà quindi leggere ogni volta la riga di intestazione (riga in grassetto) con il relativo valore, ossia per la prima riga:

Nome: A. Palmieri, Tazze: 10, Tipo: Espresso, Zucchero: No

Come si noterà dal codice seguente, valorizzando l'attributo id e assegnando i relativi valori alle celle contenenti dati (celle di tipo <td>) potremmo affiliare le intestazioni ai dati in modo semplice.

```
<table summary="Questa tabella contiene la
rappresentazione del numero
di caffè consumati da ogni onorevole,
con il tipo di caffè (decaffeinato o normale),
e se consumato con zucchero.">
<caption>Caffè consumato da ogni onorevole</caption>
<tr>
<th id="intestazione1">Nome</th>
<th id="intestazione2">Tazze</th>
<th id="intestazione3" abbr="Tipo">Tipo di caffè</th>
<th id="intestazione4">Zucchero</th>
</tr>
<tr>
<td headers="intestazione1">A. Palmieri</td>
<td headers="intestazione2">10</td>
<td headers="intestazione3">Espresso</td>
<td headers="intestazione4">No</td>
</tr>
<tr>
<td headers="intestazione1">C. Campa</td>
<td headers="intestazione2">5</td>
<td headers="intestazione3">Decaffeinato</td>
```

```
<td headers="intestazione4">Sì</td>
</tr>
</table>
```

Il programma di lettura dello schermo leggerà i dati della tabella nel modo seguente:

```
Caption: Caffè consumato da ogni onorevole
Summary: Questa tabella contiene la rappresentazione del
Numero di caffè consumati da ogni onorevole,
con il tipo di caffè (decaffeinato o normale),
e se consumato con zucchero.
Nome: A. Palmieri, Tazze: 10, Tipo: Espresso, Zucchero: No
Nome: C. Campa, Tazze: 5, Tipo: Decaffeinato, Zucchero: Sì
```

Si noterà che anziché leggere "Tipo di caffè" il sistema legge "Tipo", in quanto è stata definita un'abbreviazione (attributo `abbr`) per non rendere ridondante e pesante la lettura ad ogni riga di una frase lunga. È necessario precisare che in questo caso si parla di attributo `abbr` che ha funzionalità inversa dell'elemento `<abbr>` e che è spiegato nel Punto di controllo 5.6 richiesto da questo stesso requisito. Un sommario delle tabelle (definito con l'attributo `summary`) è particolarmente utile per le tecnologie assistive utilizzate dagli utenti non vedenti. Qualsiasi tabella di dati dovrebbe contenere delle informazioni preventive, valorizzando l'elemento `<caption>`, eventualmente sostituibile dall'attributo `title` se si desidera renderlo visibile esclusivamente agli utenti non vedenti.

### **Punto di controllo 5.5 (Priorità 3)**

#### **Per ogni tabella definire un sommario dei contenuti**

Mentre l'elemento `<caption>` e l'attributo `title` rappresentano un "titolo" della tabella, l'attributo `summary` è di fatto una guida all'utente sul contenuto e sull'organizzazione della tabella dati. Nella nostra tabella di esempio sui caffè consumati dagli onorevoli, l'attributo `summary` descriveva chiaramente l'organizzazione dei contenuti.

```
<table summary="Questa tabella contiene la
rappresentazione del numero
di caffè consumati da ogni onorevole,
con il tipo di caffè
(decaffeinato o normale),
e se consumato con zucchero.">
...
...
</table>
```

Gli utenti non vedenti, ricevendo informazioni descrittive dei contenuti grazie al contenuto dell'attributo `summary`, potranno valutare se fruire del contenuto della tabella o passare al paragrafo successivo.

### Punto di controllo 5.6 (Priorità 3)

#### Fornire abbreviazioni per le etichette di intestazione

Un attributo interessante per l'elemento `<th>` è senz'altro `abbr`: grazie a tale attributo un lettore dello schermo leggerà il valore contenuto in `abbr` in sostituzione del contenuto della cella, in modo da eliminare pesanti ripetizioni. Nel nostro esempio (caffè consumati dagli onorevoli) era presente la seguente riga di codice:

```
<th id="intestazione3" abbr="Tipo">Tipo di caffè</th>
```

Il lettore dello schermo leggerà "Tipo" invece che "Tipo di caffè", seguito dal valore della relativa cella contenente le informazioni richieste.

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la Barra dell'accessibilità per:

- verificare la presenza di tabelle dati;
- verificare l'uso corretto degli elementi `<caption>`, `<tr>`, `<th>` e `<td>` secondo le specifiche;
- verificare la presenza dell'attributo `summary`.

Tramite la Barra dell'accessibilità è possibile utilizzare:

### Struttura

*Tabelle dati semplici:* Visualizza gli elementi `<table>`, `<th>`, `<td>` sulla pagina corrente insieme agli attributi raccomandati per il linguaggio di marcatura di tabelle di dati semplici (`summary`, `scope`). Un attributo vuoto indica che l'attributo non è stato utilizzato pur essendone raccomandato l'uso (Punti di controllo 5.1, 5.5 e 5.6). Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 10

Enunciato: Per le tabelle dati usare gli elementi (marcatori) e gli attributi previsti nella DTD adottata per associare le celle di dati e le celle di intestazione che hanno due o più livelli logici di intestazione di righe o colonne.

Riferimenti WCAG 1.0: 5.2

Riferimenti Section 508: 1194.22 (h)

### Motivazione

---

Abbiamo già visto nel requisito precedente come sia necessario consentire una corretta lettura dei contenuti nelle tabelle dati. Se una tabella dati è complessa, sarà necessario applicare una serie di tecniche: è possibile utilizzare ulteriori elementi che consentono agli utenti con tecnologie assistive di poter comprendere i contenuti e di fruirne secondo l'ordine logico stabilito dallo sviluppatore.

La finalità del requisito chiede quindi, anche in questo caso, di poter garantire la fruibilità di una tabella dati ad utenti dotati di tecnologie assistive o di configurazioni particolari.

### Implementazione

---

Il requisito chiede di garantire la fornitura di informazioni che non impediscano ad alcune categorie di utenti di poter fruire dei contenuti delle tabelle dati complesse contenute in una pagina web. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti. Sarà quindi richiesto di verificare l'utilizzo di elementi ed attributi in modo conforme alle specifiche e soprattutto l'accessibilità dei dati contenuti nelle tabelle complesse.

### Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

**Punto di controllo 5.2 (Priorità 1). Per le TABELLE DATI che hanno due o più livelli logici di intestazioni di righe o colonne, usare marcatori per associare le celle di dati e le celle di intestazione**

Come abbiamo già accennato nelle motivazioni del requisito, non sempre le tabelle dati hanno una struttura semplice come quella rappresentata nel requisito precedente. Per le tabelle con strutture più complesse, è possibile utilizzare gli elementi e gli attributi idonei:

- l'elemento `<thead>` raggruppa le righe di intestazione (elementi `<tr>`, `<th>` e `<td>`);
- l'elemento `<tbody>` raggruppa le righe interne alla tabella (elementi `<tr>` e `<td>`);
- l'elemento `<tfoot>` raggruppa le righe di chiusura della tabella (elementi `<tr>` e `<td>`);
- gli elementi `<col>` e `<colgroup>` raggruppano le colonne;
- gli attributi "axis", "scope" ed "headers" descrivono relazioni più complesse fra i dati.

Immaginiamo ora una tabella contenente i dati relativi a delle spese per viaggi: mentre nella tabella di esempio per il requisito precedente leggere i contenuti con un lettore dello schermo era piuttosto semplice, una tabella come di questo tipo risulterà di difficile interpretazione senza l'utilizzo di elementi ed attributi adeguati. Per esempio, utilizzando l'attributo `headers` in modo scorretto non verranno lette le intestazioni prima del valore di una cella dati: sfido chiunque a ricordarsi, dopo una ventina di righe di tabella, quali erano i valori delle intestazioni. Un esempio di tabella dati complessa è rappresentato qui di seguito. È comprensibile come gli attributi `headers` siano necessari per rendere comprensibile il contenuto agli utenti che utilizzano lettori dello schermo. Di seguito è rappresentato il codice della tabella in cui sono utilizzati elementi ed attributi in modo conforme a quanto richiesto dal requisito.

Esempio di tabella dati

	<b>Intestazione Colonna 1</b>	<b>Intestazione Colonna 2</b>
<b>Intestazione Riga 1</b>	Col. 1 Riga 1	Col. 1 Riga 2
<b>Intestazione Riga 2</b>	Col. 2 Riga 1	Col. 2 Riga 2

```
<table summary="Questa tabella contiene le spese generali
suddivise per affiliazione (nominativo ed ufficio
di competenza e per descrizione (Tipologia di spesa
ed importo">
```

```
<caption>Spese generali anno 2005</caption>
```

```
<colgroup span="2" />
```

```
<colgroup span="2" />
```

```
<thead>
```

```
<tr>
```

```
<th colspan="2" id="affiliazione">Affiliazione</th>
```

```
<th colspan="2" id="descrizione">Descrizione</th>
```

```
</tr>
```

```
<tr>
```

```
<th id="nominativo">Nominativo</th>
```

```
<th id="ufficio">Ufficio</th>
```

```
<th id="tipologia" abbr="Tipo">Tipologia di spesa</th>
```

```
<th id="importo">Importo</th>
```

```
</tr>
```

```
</thead>
```

```
<tfoot>
```

```
<tr>
```

```
<td colspan="4">Aggiornata al 27 maggio 2005</td>
```

```
</tr>
```

```
</tfoot>
```

```
<tbody>
```

```
<tr>
```

```
<td headers="affiliazione nominativo">Bianchi Michele</td>
```

```
<td headers="affiliazione ufficio">Amministrazione</td>
```

```
<td headers="descrizione tipologia">Carburante</td>
```

```
<td headers="descrizione importo">250,00</td>
```

```
</tr>
```

```
<tr>
```

```
<td headers="affiliazione nominativo">Rossi Mario</td>
```

```
<td headers="affiliazione ufficio">Acquisti</td>
```

```

<td headers="descrizione tipologia">Cancelleria</td>
<td headers="descrizione importo">112,15</td>
</tr>
<tr>
<td headers="affiliazione nominativo">Scano Roberto</td>
<td headers="affiliazione ufficio">Vendite</td>
<td headers="descrizione tipologia">Consumabili</td>
<td headers="descrizione anno">400,00</td>
</tr>
</tbody>

</table>

```

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la barra dell'accessibilità per:

- verificare la presenza di tabelle dati complesse;
- verificare la presenza degli elementi ed attributi previsti dalle specifiche per le tabelle dati semplici;
- verificare l'uso corretto degli elementi <thead>, <tbody>, <tfoot>, <col>, <colgroup> e degli attributi "axis", "scope", "headers" secondo le specifiche.

Tramite la Barra dell'accessibilità è possibile utilizzare:

### Struttura

*Tablelle dati complesse.* Visualizza gli elementi <table>, <th>, <td>, <thead>, <tbody>, <tfoot>, <col>, <colgroup> sulla pagina corrente insieme agli attributi raccomandati per il linguaggio di marcatura di tabelle di dati complesse (summary, scope, id, headers). Un attributo vuoto indica che l'attributo non è stato utilizzato pur essendone raccomandato l'uso (Punto di controllo 5.2).

*Ordinamento delle celle in tabella.* Visualizza l'ordine (numerico) di tabulazione ed i bordi delle celle delle tabelle sulla pagina corrente (Punto di controllo 5.2). Grazie a tale funzionalità l'esperto può verificare il corretto ordine di lettura. Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)



## Requisito 11

Enunciato: Usare i fogli di stile per controllare la presentazione dei contenuti e organizzare le pagine in modo che possano essere lette anche quando i fogli di stile siano disabilitati o non supportati.

Riferimenti WCAG 1.0: 3.3, 6.1

Riferimenti Section 508: 1194.22 (d)

### Motivazione

---

La separazione del contenuto dalla presentazione è uno dei punti cardine per l'accessibilità. Ciò significa consentire a qualsiasi utente di poter fruire dei contenuti ed allo stesso tempo di gestirne la rappresentazione in diverse modalità. Garantire la fruibilità della pagina disabilitando i fogli di stile significa sviluppare i contenuti e l'impaginazione (layout) in modo che un utente che utilizza tecnologie assistive, e che quindi leggono il codice della pagina (non la sua rappresentazione grafica), possa comunque accedere ai contenuti in un ordine di lettura comprensibile.

### Implementazione

---

Il requisito chiede di garantire la fornitura di informazioni che non impediscano ad alcune categorie di utenti di poter fruire dei contenuti della pagina web definiti graficamente (impaginati) tramite fogli di stile. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti. Sarà quindi richiesto di verificare l'effettiva fruibilità del contenuto, ad esempio disabilitando i fogli di stile.

### Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

## Punto di controllo 3.3 (Priorità 2). Usare i fogli di stile per controllare l'impaginazione e la presentazione

L'utilizzo dei fogli di stile (CSS) è essenziale per lo sviluppo dei siti Web, in quanto consente di creare siti professionali e maggiormente accessibili: non è più necessario utilizzare i seguenti elementi:

```
<center>, <font>, <basefont>, <b>, <i>, <tt>, <big>, <small>,
<strike>, <s>, <u>
```

Con i fogli di stile è possibile ottenere i medesimi risultati in modo più efficiente. È importante ricordare che gli elementi di tipo `<h1>` non vanno utilizzati per ottenere effetti grafici di ridimensionamento del carattere, che potranno essere ottenuti creando delle classi dedicate nel foglio di stile.

```
.testogrande {
font-family: Georgia, "Times New Roman", Times, serif;
font-size: 1.3em;
color: #000;
background-color: #fff;
border: 1px solid #000;
padding: 0px;
}
```

L'allineamento, i margini e il posizionamento degli elementi nella presentazione di una pagina vanno gestiti tramite fogli di stile: lo sviluppatore deve garantire la possibilità di lettura dei contenuti della pagina anche in mancanza dei fogli di stile stessi, al fine di garantirne l'accesso e la fruibilità agli utenti che utilizzano lettori solo testo o vocali.

**Allineamento:** utilizzando gli attributi `text-indent`, `text-align`, `word-spacing`, `font-stretch` (quest'ultimo eliminato in CSS 2.1) è possibile formattare il testo aumentando/diminuendo gli spazi, definendo l'allineamento, ecc. Con l'attributo `text-align: center` sarà possibile eliminare l'elemento `<center>`, ormai obsoleto.

**Margini:** `margin`, `margin-top`, `margin-right`, `margin-bottom`, `margin-left` consentono di definire i margini dell'oggetto (`<div>`, ...) all'interno del quale verranno forniti i contenuti.

**Posizionamento:** `float`, `position`, `top`, `right`, `bottom`, `left` consentono di posizionare l'oggetto nella pagina.

Si consiglia di evitare l'uso degli stili creati con l'attributo HTML "style": dalla versione 1.1 di XHTML tale elemento è stato deprecato per garantire una maggiore separazione di contenuto e presentazione.

## Punto di controllo 6.1 (Priorità 1). Organizzare i documenti in modo che possano essere letti anche in assenza dei fogli di stile

È necessario inoltre sviluppare i siti in modo che possano essere fruiti anche se il browser utilizzato dall'utente non supporta i fogli di stile: la finalità dello sviluppatore deve essere quella di garantire la fruibilità del sito e, di conseguenza, la e l'usabilità dei suoi contenuti. Un esempio di contenuti con una particolare formattazione generata da fogli di stile è il seguente.



Figura 4.5 Esempio d'uso dei fogli di stile.

La frase "The quick brown fox jumped over the lazy dogs" viene spesso utilizzata in questo tipo di esempi poiché, contenendo tutte le lettere dell'alfabeto, consente di valutare la resa della rappresentazione dei singoli caratteri. Il testo nella figura è stato generato e posizionato utilizzando i fogli di stile, definendo quattro elementi div.

```
<div class="part1">The quick</div>  
<div class="parte2">brown fox</div>  
<div class="parte3">jumped over</div>  
<div class="parte4">the lazy dogs.</div>
```

Per ottenere l'effetto grafico desiderato, per ogni blocco di testo è stata impostata una particolare combinazione di caratteri e colori.

```
.part1 /* The quick */ {  
color: red;  
font-size: 1.0em;
```

```
padding-left: 0;
margin-top: 40px;
font-family: copperplate gothic bold, fantasy, sans-serif }

.parte2 /* brown fox */ {
color: brown;
font-size: 0.7em;
padding-left: 100px;
margin-top: 30px;
font-family: times new roman, desdemona, serif }

.parte3 /* jumped over */ {
color: purple;
font-size: 0.8em;
padding-left: 200px;
margin-top: -60px;
font-family: desdemona, times new roman, serif }

.parte4 /* the lazy dogs */ {
color: blue; size: 2.4em;
padding-left: 350px;
margin-top: -80px;
margin-bottom: 100px;
font-family: fantasy, copperplate gothic bold, sans-serif }
```

Se utilizziamo un browser che consente di disattivare i fogli di stile, come Opera, il risultato sarà comunque un testo chiaramente leggibile e fruibile:

```
The quick
brown fox
jumped over
the lazy dogs.
```

Modificando il posizionamento dei testi, per un utente che utilizza un browser in grado di interpretare i CSS il risultato non sarebbe cambiato, mentre a un utente con tale funzionalità disabilitata verrebbe mostrato un testo senza senso: il contenuto della pagina non soddisferebbe questo requisito.

Per la stessa motivazione, quando si sviluppano layout tramite elementi <div> è necessario, disabilitando i fogli di stile, verificare che tutti i contenuti siano comprensibili secondo il normale ordine di lettura dei contenuti della pagina.

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la Barra dell'accessibilità per:

- verificare l'uso di elementi e/o attributi deprecati a favore dei fogli di stile;
- verificare, disabilitando i fogli di stile, che i contenuti siano ancora fruibili e comprensibili secondo il loro ordine di presentazione.

Tramite la Barra dell'accessibilità è possibile utilizzare:

### **CSS**

*Attiva/Disattiva i CSS:* Attiva/disattiva i fogli di stile esterni (Punti di controllo 3.3 e 6.1).

*Disattiva i CSS inline:* Rimuove l'attributo `style` (se presente) da tutti gli elementi della pagina (Punti di controllo 3.3 e 6.1).

*Elementi e attributi deprecati (Nuova finestra).* Visualizza gli elementi e gli attributi deprecati. (Punto di controllo 3.3).

### Struttura

*Bordi delle tabelle.* Evidenzia tutti i bordi delle tabelle/celle sulla pagina corrente (Punto di controllo 3.3).

*Ordine di tabulazione.* Visualizza per ogni elemento attivo il numero corrispondente al suo ordine di tabulazione, al fine di poter consentire all'esperto di verificare la correttezza dell'ordine di navigazione anche senza disabilitare i fogli di stile (Punto di controllo 6.1).

*Visualizza i Div.* Mostra l'ordine (numerico) di tabulazione ed i bordi degli elementi `<div>` presenti sulla pagina corrente (Punto di controllo 6.1).

Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 12

Enunciato: La presentazione e i contenuti testuali di una pagina devono potersi adattare alle dimensioni della finestra del browser utilizzata dall'utente senza sovrapposizione degli oggetti presenti o perdita di informazioni tali da rendere incomprensibile il contenuto, anche in caso di ridimensionamento, ingrandimento o riduzione dell'area di visualizzazione o dei caratteri rispetto ai valori predefiniti di tali parametri.

Riferimenti WCAG 1.0: 3.4

Riferimenti Section 508: non presente

### Motivazione

---

Questo requisito è particolarmente sentito dagli utenti ipovedenti, in quanto se non viene soddisfatto esistono grosse possibilità che un utente con tale disabilità visiva non possa accedere ai contenuti delle pagine. Considerando che in tale categoria di utenti ricadono anche le persone anziane, non rispettare questo punto di controllo provoca l'esclusione di una grossa fetta di utenza.

Il requisito inoltre garantisce ad utenti che utilizzano configurazioni particolari di poter accedere al contenuto che, se sviluppato secondo i criteri che andremo ad analizzare, si adatterà alla periferica di output.

### Implementazione

---

Il requisito chiede di garantire la fornitura di informazioni che non impediscano ad alcune categorie di utenti di poter fruire dei contenuti della pagina web indipendentemente dalla dimensione della finestra di visualizzazione e/o dalla dimensione dei caratteri. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti. Sarà quindi richiesto di verificare l'effettiva fruibilità del contenuto, ad esempio visualizzando la pagina in diverse risoluzioni. Il requisito non richiede comunque di praticare l'impossibile ma di garantire una certa visibilità dei contenuti anche a risoluzioni video 800 x 600 a caratteri molto grandi considerando che tale risoluzione è ormai un'impostazione minima consigliata per ogni configurazione hardware.

Relativamente a fogli di stile per i diversi media se questi vengono forniti devono anch'essi rispettare il requisito: un foglio di stile per la stampa dovrà adattarsi

alle impostazioni del browser dell'utente e consentire la stampa dei contenuti mentre un foglio di stile per i dispositivi handheld (palmari) dovrà consentirne la visualizzazione da parte dei browser che supportano tale tipologia del foglio di stile.

## Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

### **Punto di controllo 3.4 (Priorità 2). Usare unità relative anziché assolute nei valori degli attributi del linguaggio e per i valori delle proprietà del foglio di stile**

Come affermato nel requisito precedente, tutte le impostazioni relative alla rappresentazione devono essere definite all'interno dei fogli di stile. L'utilizzo di un'unità di misura di tipo assoluto, come pica (pc), punti (pt), inches (in), centimetri (cm), e millimetri (mm) ha come effetto il blocco delle dimensioni di un elemento a una dimensione fissa: un testo per noi di dimensioni "normali" potrebbe essere davvero troppo piccolo per un utente ipovedente o per chi utilizza monitor a risoluzioni elevate. Se invece le dimensioni del testo vengono definite con unità misura di tipo relativo, come .em o in percentuale (%), i visitatori del sito potranno facilmente adattare il carattere alle proprie esigenze. Ciò comporta da parte dello sviluppatore una fase di test delle pagine simulando tali visualizzazioni e controllando che anche a risoluzioni medio-basse (800x600) con caratteri molto grandi il proprio sito sia fruibile senza problemi.

Un caso particolare è relativo ai pixel. Nei documenti ufficiali W3C relativi alla raccomandazione CSS 2.0 è definito che la misura in pixel (.px) è relativa alla periferica di presentazione: un pixel di una slide in uno schermo del computer portatile di un docente avrà sicuramente dimensione differente di un pixel dello stesso schermo proiettato su di un pannello largo 3 metri.

Per questa relazione alla periferica (la dizione corretta è "relativo al dispositivo di visualizzazione"), in Microsoft Internet Explorer nella funzionalità di ingrandimento carattere gli elementi di testo dimensionati in pixel non vengono ridimensionati, con gravi problemi di accessibilità per gli ipovedenti (o semplicemente, per i miopi).

In ogni caso, al fine di rispettare la metodologia di verifica dei requisiti non è possibile utilizzare il pixel come unità di misura per i caratteri, in quanto la metodologia all'allegato A, lettera C) al punto 5 dichiara:

5. I contenuti della pagina siano fruibili in caso di utilizzo delle funzioni previste dai browser per definire la grandezza dei caratteri;

Non è inoltre possibile utilizzare script o altre tecniche per ridimensionare "al volo" i pixel, considerando che sempre la metodologia alla lettera C) al punto 7 dichiara:

7. i contenuti e le funzionalità della pagina siano ancora fruibili, anche in modalità diverse, in caso di disattivazione di fogli di stile, script e applet ed altri oggetti di programmazione;

Per garantire l'accessibilità sarà necessario non utilizzare i pixel per assegnare dimensioni ai testi di una pagina, utilizzando invece le dimensioni espresse in "em". Per lo stesso motivo, è consigliabile utilizzare - ove possibile - le dimensioni in percentuale per tutti gli attributi di impostazione di larghezza ed altezza degli elementi presentazionali.

Nelle tecniche di applicazione dei CSS per le WCAG 2.0 è chiarito definitivamente il problema <http://www.w3.org/WAI/GL/WCAG20/WD-WCAG20-CSS-TECHS/#syntax-data-types>:

Utilizzare "em" o dimensioni percentuali per le proprietà che necessitano di essere modificate. Tali dimensioni sono "font-size" e "line-height".

Utilizzare "px" per le proprietà che non necessitano di essere modificate. Tali dimensioni sono "height" e "width" per le immagini raster, "margin" e "border".

Relativamente ai layout a dimensioni fisse i valori per padding, margin, top, left, width e height possono essere espressi in px, ma - al fine di rispettare il requisito - non devono provocare sovrapposizioni tra i contenuti.

Di seguito un esempio di come preparare un foglio di stile (CSS) che consenta l'utilizzo di caratteri con dimensioni di tipo relativo, utilizzando le dimensioni di tipo em, ed elementi con dimensioni in percentuale.

```
body {  
font-family: Georgia, "Times New Roman", Times, serif;  
color: #000;  
background-color: #369;
```



```
font-size: .9em;
}
a {
color: #036;
text-decoration: underline;
background-color: transparent;
}
.codice {
font-family: "CoURier New", CoURier, serif;
background-color: transparent;
color: #666;
font-size: 0.9em;
}
div {
color: #036;
width: 25%;
background-color: transparent;
}
```

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la Barra dell'accessibilità per:

- verificare la presenza di dimensioni assolute e/o dimensioni in pixel all'interno del foglio di stile o degli stili in linea;
- verificare che al ridimensionamento dei caratteri il contenuto rimanga leggibile e comprensibile;
- verificare che al ridimensionamento della finestra il contenuto rimanga leggibile e comprensibile.

Tramite la Barra dell'accessibilità è possibile utilizzare:

### **Ridimensiona**

*800 x 600.* Ridimensiona l'attuale finestra del browser ad una larghezza di 800 pixel e ad un'altezza di 600 pixel.

*1024 x 768.* Ridimensiona l'attuale finestra del browser ad una larghezza di 1024 pixel e ad un'altezza di 768 pixel.

*Misure personalizzate.* Ridimensiona la finestra corrente del browser ad un'altezza/larghezza definita dall'utente.

*Misure personalizzate.* Controllo della grandezza dello schermo Collegamento ad un test on-line per la misura dello schermo e presenta delle informazioni sulle varie configurazioni per un vasto insieme di browser/schermi e risoluzioni/piattaforme.

## **CSS**

*Visualizza gli stili applicati (Nuova finestra).* *Visualizza gli stili (in una nuova finestra) associate col contenuto posizionato al di sotto del puntatore del mouse.* L'esperto potrà quindi verificare la presenza di caratteri o di elementi in cui non sono state utilizzate dimensioni di tipo relativo e/o dimensioni in pixel (Punto di controllo 3.4).

*Visualizza il foglio di stile (Nuova finestra).* *Visualizza il contenuto dei fogli di stile richiamati dalla pagina corrente (in una nuova finestra).* L'esperto potrà quindi verificare la presenza di caratteri o di elementi in cui non sono state utilizzate dimensioni di tipo relativo e/o dimensioni in pixel (Punto di controllo 3.4).

## **Strumenti**

*Juicy Studio Tools / CSS Accessibility Analyzer (Nuova finestra).* Invia al sistema di valutazione di JuicyStudio il foglio di stile della pagina corrente. CSS Accessibility Analyzer verificherà la conformità del foglio di stile (segnalando errori o avvisi) nonché la presenza del contrasto dei colori (Requisito 6) e segnalando l'eventuale presenza di caratteri o elementi in dimensioni assolute oppure l'uso di dimensioni tramite pixel (Punto di controllo 3.4).

L'esperto potrà inoltre utilizzare le funzionalità di ingrandimento caratteri previste dai diversi browser con cui effettua il test come indicato al Punto 2 Paragrafo c) della metodologia per la verifica tecnica. Un ottimo strumento di supporto è senz'altro browsercam.com, che consente di acquisire delle immagini per una determinata pagina web nonché di poter utilizzare una connessione remota ad un personal computer in configurazioni non disponibili all'utente (per esempio, un utente dotato di PC con Microsoft Windows potrà trovare utile testare la pagina con MAC, con Linux ed i relativi browser).

Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 13

Enunciato: In caso di utilizzo di tabelle a scopo di impaginazione, garantire che il contenuto della tabella sia comprensibile anche quando questa viene letta in modo linearizzato e utilizzare gli elementi e gli attributi di una tabella rispettandone il valore semantico definito nella specifica del linguaggio a marcatori utilizzato.

Riferimenti WCAG 1.0: 5.3, 5.4

Riferimenti Section 508: non presente

### Motivazione

---

È bene ricordare che gli sviluppatori dovrebbero utilizzare i fogli di stile (CSS) per lo sviluppo dell'impaginazione (layout) di una pagina Web. Ancora oggi le tabelle vengono utilizzate spesso per impaginazione anziché per contenere dati. Per tale motivo nelle linee guida per l'accessibilità dei contenuti del W3C e quindi all'interno di questo requisito si richiede agli sviluppatori - nel caso utilizzassero ancora le tabelle a scopo di impaginazione - di non usare elementi ed attributi dedicati alle tabelle dati.

### Implementazione

---

Il requisito chiede di garantire la fornitura di informazioni che non impediscano ad alcune categorie di utenti di poter fruire dei contenuti delle tabelle di impaginazione contenute in una pagina web. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti. Sarà quindi richiesto di verificare l'utilizzo di elementi ed attributi in modo conforme alle specifiche e soprattutto la linearizzazione - quindi la comprensibilità - dei contenuti presentati tramite le tabelle di impaginazione.

### Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

### **Punto di controllo 5.3 (Priorità 2). Non usare le TABELLE PER IMPAGINAZIONE a meno che la tabella non sia comprensibile se linearizzata**

Se è necessario utilizzare tabelle di impaginazione, il contenuto della tabella deve poter essere linearizzato, ossia il contenuto delle celle verrà interpretato come una serie di paragrafi successivi. Per questo motivo, è necessario che la lettura dei contenuti (che avviene riga per riga, a seconda della direzione del testo) conservi un senso.

Utilizzando le tabelle di impaginazione si possono riscontrare dei problemi con le tecnologie assistive: utilizzando una tabella come quella di seguito rappresentata allo scopo di dividere la pagina in due colonne, i programmi di lettura dello schermo leggeranno cella dopo cella, ovvero nel nostro esempio per primo il contenuto della prima cella della prima colonna, il contenuto della prima cella della seconda colonna, quindi la seconda cella della prima colonna e così di seguito.

```
<table summary="Indicazioni sulla struttura della tabella">
<tr>
<td>Riga 1 Colonna 1</td>
<td>Riga 1 Colonna 2</td>
</tr>
<tr>
<td>Riga 2 Colonna 1</td>
<td>Riga 2 Colonna 2</td>
</tr>
</table>
```

Nella verifica del requisito vedremo come siano disponibili degli strumenti per verificare la linearizzazione delle tabelle.

### **Punto di controllo 5.4 (Priorità 2). Se si utilizza una TABELLA DI IMPAGINAZIONE non utilizzare alcun marcatore di struttura a scopo di formattazione**

Se si utilizzano le tabelle per sviluppare un layout, è necessario ricordare che gli elementi `<th>` non devono essere utilizzati a scopo di formattazione. Solitamente l'elemento `<th>` (table header) viene visualizzato in grassetto e centrato nella cella: se la cella non rappresenta una intestazione di riga o colonna, la formattazione deve avvenire tramite foglio di stile. Pertanto per una tabella di impaginazione non va utilizzato alcun elemento strutturale (`<th>`, `<thead>`,

<tfoot>, <tbody>, <colgroup>, <col>) in quanto ciò confonderebbe le tecnologie assistive che hanno lo scopo di linearizzarne e leggerne i contenuti.

Per esempio, l'uso di:

```
<th>Testo centrato e grassetto</th>
```

è scorretto, mentre è necessario utilizzare codice come il seguente:

```
<td class= "evidenzia">Testo centrato e grassetto</td>
```

in cui la classe "evidenzia" conterrà delle formattazioni tramite fogli di stile per riprodurre l'effetto desiderato.

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la barra dell'accessibilità per:

verificare la presenza di tabelle di impaginazione;

- verificare l'eventuale errata presenza degli elementi ed attributi previsti dalle specifiche per le tabelle dati;
- verificare che il contenuto della tabella sia linearizzabile.

Tramite la Barra dell'accessibilità è possibile utilizzare:

### **Struttura**

*Bordi delle tabelle.* Evidenzia tutti i bordi delle tabelle/celle sulla pagina corrente al fine di identificare la struttura della pagina (Punto di controllo 5.4).

Tablette dati complesse. Visualizza gli elementi <table>, <th>, <td>, <thead>, <tbody>, <tfoot>, <col>, <colgroup> sulla pagina corrente insieme agli attributi raccomandati per il linguaggio di marcatura di tabelle di dati complesse (summary, scope, id, headers). Un attributo vuoto indica che l'attributo non è stato utilizzato. L'esperto potrà quindi verificare l'errato utilizzo di tali elementi per le tabelle di impaginazione (Punto di controllo 5.4).

Ordinamento delle celle in tabella. Visualizza l'ordine (numerico) di tabulazione ed i bordi delle celle delle tabelle sulla pagina corrente (Punto di controllo 5.3). Grazie a tale funzionalità l'esperto può verificare il corretto ordine di lettura.

Linearizza (Rimuovi le tabelle). Rimuove tutti gli elementi <table>, <th>, <tr>, <td> e <div> dalla pagina corrente per controllare la linearizzazione dei contenuti (Punto di controllo 5.3).

Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 14

Enunciato: Nei moduli (form), associare in maniera esplicita le etichette ai rispettivi controlli, posizionandole in modo che sia agevolata la compilazione dei campi da parte di chi utilizza le tecnologie assistive.

Riferimenti WCAG 1.0: 10.2, 12.4

Riferimenti Section 508: 1194.22 (n)

### Motivazione

---

I moduli (form) consentono l'interazione tra l'utente ed i contenuti nella pagina web: moduli di ricerca, moduli di contatto, questionari sono attualmente largamente diffusi.

Ogni elemento contenuto nel modulo deve perciò essere chiaramente identificabile da parte degli utenti, indipendentemente dalla disabilità. È quindi necessario garantire che l'utente possa comprendere l'etichetta che accompagna i campi di inserimento e di selezione implementando delle tecniche che consentano alle tecnologie assistive di affiliare l'elemento alla propria etichetta informativa. Nel caso queste etichette non siano presenti o non siano posizionate correttamente l'utente non potrà procedere con la compilazione del modulo, che quindi sarà inaccessibile.

### Implementazione

---

Il requisito chiede di garantire la fornitura di informazioni che garantiscono a tutti gli utenti di poter utilizzare i moduli contenuti in una pagina web. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti. Sarà quindi richiesto di verificare l'utilizzo di elementi ed attributi in modo conforme alle specifiche.

### Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito. I riferimenti alle WCAG riguardano sia le etichette implicite che le etichette esplicite mentre di fatto il requisito riguarda le etichette esplicite. È

possibile ipotizzare che tale definizione sia stata inserita richiedendo di garantire la possibilità di accedere ai moduli anche per le etichette non fornite in modo esplicito: l'accesso ai moduli, quindi, deve in ogni caso essere garantito.

## **Punto di controllo 10.2 (Priorità 2). Fino a quando i programmi utente non supporteranno esplicite associazioni fra etichette e controlli dei moduli, garantire che l'etichetta sia posizionata correttamente per tutti i controlli dei moduli che hanno etichette associate implicitamente**

Nella motivazione del requisito è chiaramente spiegato come sia necessario fornire un'etichetta per ogni elemento di un modulo:

- elemento `<input>`, ad esclusione degli elementi con `type="button"`;
- elemento `<textarea>`;
- elemento `<select>`.

Le etichette vengono identificate tramite l'elemento `<label>` che, secondo quanto richiesto dal Punto di controllo 10.2, viene associato al modulo in modo implicito:

```
<label>  
Nome Utente: <input type="text" size="20"  
id="utente" name="utente"/>  
</label>
```

In questo modo una tecnologia assistiva è in grado di identificare che il testo dell'etichetta è affiliato all'elemento input identificato come "utente".

Utilizzando un normale browser è possibile cliccare sull'etichetta "Nome Utente:" e si verrà trasferiti all'interno del campo di input. Questa caratteristica risulta utile quindi non solo agli utenti non vedenti ma anche ad utenti con disabilità o difficoltà motorie. La normativa italiana però chiede di riferirsi alle etichette esplicite, descritte nel punto di controllo 12.4 ed ha anticipato quanto richiesto attualmente dalle techniques delle WCAG 2.0, ovvero di utilizzare la modalità esplicita anziché la modalità implicita per assegnare le etichette ai rispettivi controlli.



## Punto di controllo 12.4 (Priorità 2). Associare esplicitamente le etichette ai loro controlli.

Per soddisfare questo punto di controllo, è necessario utilizzare l'elemento `<label>` in modo esplicito tramite l'attributo `for` che deve corrispondere all'attributo `id` dell'elemento interno al modulo:

```
<p>
<label for="utente">Nome Utente:</label>
<input type="text" size="20" id="utente" name="utente" />
</p>
```

Come si noterà, tramite l'attributo `for` è possibile associare un elemento `<label>` in modo esplicito a un determinato elemento interno al modulo, ricordando che il valore dell'attributo `for` dell'elemento `<label>` deve corrispondere al valore dell'attributo `id` dell'elemento interno al modulo.

Perché è comunque necessario usare la modalità esplicita? Ipotizziamo di utilizzare in un sito una tabella di impaginazione per presentare un modulo di raccolta dati. In questo caso, ci troveremmo ad avere una colonna della tabella contenente gli elementi `<label>` e una colonna con elementi interni al modulo (campi di input testuale, ecc.). Se si utilizzasse la modalità implicita, l'elemento `<label>` non riuscirebbe ad individuare il proprio elemento interno al modulo, per cui è necessario utilizzare l'attributo `for`. Riporto una serie di esempi.

```
<tr>
<td><label for="nome">Nome:</label></td>
<td><input type="text" size="20"
id="nome" name="nome" />
</td>
</tr>
```

Nel primo caso abbiamo una tabella contenente l'etichetta nella cella di sinistra e il campo di input nella cella di destra: cliccando o selezionando l'elemento `<label>` l'utente verrà trasferito nel campo di input. Prendiamo invece il caso di due pulsanti di selezione (radio button) che condividono lo stesso attributo `name` ma devono utilizzare due diversi ID, questo per poter affiliare in modo corretto gli elementi `<label>`.

```
<fieldset>
<legend>Sesso</legend>
<label for="sessoM">Maschio</label>
<input type="radio" size="20"
id="sessoM" name="sesso" /> <br />
```

```

<label for="sessoF">Femmina</label></td>
<td><input type="radio" size="20"
id="sessoF" name="sesso" />
</fieldset>

```

In modo simile è possibile definire delle caselle di controllo (checkbox) che, se numericamente sostanziose potrebbero richiedere l'uso di elementi di raggruppamento come <fieldset>, altrimenti possono in ogni caso essere intabellati.

```

<tr>
<td><label for="cliente">Cliente XYZ</label></td>
<td><input type="checkbox" size="20"
id="cliente" name="cliente" />
</td>
</tr>

```

Per quanto riguarda invece gli elenchi di selezioni (elemento <select>) l'etichetta corrisponde all'id dell'elemento select. Nell'esempio seguente utilizziamo, oltre all'elemento <option> anche l'elemento <optgroup> per raggruppare le diverse opzioni e consentire agli utenti di poter saltare tra i diversi gruppi.

```

<tr>
<td><label for="citta">Città</label></td>
<td><select name="citta" id="citta">
<optgroup label="Lombardia">
<option value="Brescia">Brescia</option>
</optgroup>
<optgroup label="Veneto">
<option value="Venezia">Venezia</option>
</optgroup>
</td>
</tr>

```

Per finire riporto l'esempio di una area di inserimento testuale (elemento <textarea> la cui etichetta è collegata tramite l'attributo ID come per gli altri elementi di input.

```

<tr>
<td><label for="note">Annotazioni:</label></td>
<td><textarea rows="2" cols="70"
name="note" id="note">
</textarea>

```

```
</td>  
</tr>
```

Utilizzando il codice appena rappresentato, in cui sono evidenziati esempi per ogni campo del modulo che necessita l'elemento `<label>`, sarà possibile garantire la conformità al requisito. Tengo a precisare che l'impaginazione tabellare è stata utilizzata esclusivamente per rendere maggiormente chiara la problematica delle etichette per i moduli: evitare quindi l'uso delle tabelle per impaginazione, delegando invece tale funzionalità ai fogli di stile ed agli elementi strutturali come l'elemento `<div>`.

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la Barra dell'accessibilità per:

- verificare la presenza di moduli (form);
- verificare l'eventuale presenza di campi che necessitano dell'attributo `<label>`;
- verificare la corretta affiliazione degli elementi `<label>`.

Tramite la barra dell'accessibilità è possibile utilizzare:

### Struttura

*Elementi Fieldset / Label:* Visualizza gli elementi `<fieldset>`, `<legend>` e `<label>` sulla pagina corrente, e l'attributo "for" dell'elemento `<label>` (se presente), altrimenti visualizza il testo 'NoFor!'. Inoltre, visualizza l'attributo `id` [`id=""`], sui controlli dei form, se presente (Punto di controllo 12.4). Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 15

Enunciato: Garantire che le pagine siano utilizzabili quando script, applet, o altri oggetti di programmazione sono disabilitati oppure non supportati; ove ciò non sia possibile fornire una spiegazione testuale della funzionalità svolta e garantire una alternativa testuale equivalente, in modo analogo a quanto indicato nel requisito n. 3.

Riferimenti WCAG 1.0: 6.3

Riferimenti Section 508: 1194.22 (l), 1194.22 (m)

### Motivazione

---

Gli sviluppatori devono creare contenuti per il web fruibili anche quando gli script, applet o altri oggetti di programmazione vengono disattivati o se il browser in uso non ne supporta l'esecuzione. Questo significa fornire la possibilità di accedere ai contenuti anche ad utenti che non posseggono un particolare plug-in, sistema operativo o la cui tecnologia assistiva non riesce ad interagire con la tecnologia utilizzata. Come vedremo dall'analisi del requisito, spesso gli sviluppatori utilizzano funzionalità esclusivamente usando i suddetti oggetti di programmazione, non fornendo alternative e quindi possibilità a talune categorie di utenti di fruire dei contenuti.

### Implementazione

---

Il requisito chiede di garantire la fornitura di informazioni che non impediscano ad alcune categorie di utenti di poter fruire dei contenuti a causa di oggetti di programmazione contenuti in una pagina web. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti. Sarà quindi richiesto di verificare l'utilizzo di oggetti di programmazione e, ove individuati, garantire un'alternativa equivalente.

Il requisito richiede inoltre, ove non sia possibile garantire l'utilizzo dei contenuti della pagina web senza l'uso degli oggetti di programmazione, di descrivere la funzionalità degli oggetti di programmazione oltre a fornire un'alternativa testuale equivalente.

## Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

### **Punto di controllo 6.3 (Priorità 1). Garantire che le pagine siano utilizzabili quando script, applet, o altri oggetti di programmazione sono disabilitati oppure non supportati. Se questo non è possibile, fornire informazione equivalente in una pagina accessibile alternativa**

Gli sviluppatori devono creare contenuti per il Web fruibili anche quando gli script vengono disattivati o se il browser in uso non li supporta. Un'istruzione come la seguente, che consente di tornare alla pagina precedente, non potrà essere utilizzata dagli utenti che hanno disabilitato gli script o che utilizzano browser che non li supportano, come mostrato di seguito.

```
<a href="javascript:history.go(-1)">  
Pagina Precedente  
</a>
```

Per ottenere particolari risultati spesso si ricorre a DHTML (combinando JavaScript, CSS e HTML). Per esempio, immaginiamo un menu realizzato con JavaScript e CSS: come potrà essere fruito dagli utenti in caso di disabilitazione di script ed oggetti? Usando l'elemento `<noscript>` si potrà fornire una versione alternativa e il menu JavaScript verrà riprodotto anche in HTML.

```
<script type="text/javascript" src="menu.js"></script>  
<noscript>  
<ul>  
<li><a href="pagina1.html">Menu 1</a></li>  
<li><a href="pagina2.html">Menu 2</a></li>  
<li><a href="pagina3.html">Menu 3</a></li>  
</ul>  
</noscript>
```

Per quanto riguarda gli oggetti di programmazione e le applet abbiamo già analizzato degli esempi per il requisito 3, a cui si riferisce anche il presente requisito ma va chiaramente ribadito che ogni caratteristica o funzionalità vitale dello script deve essere riproposta utilizzando del codice HTML.

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la Barra dell'accessibilità per:

- verificare la presenza di script, applet e oggetti di programmazione;
- verificare il funzionamento della pagina disabilitando l'uso di script, applet ed oggetti di programmazione;
- verificare la presenza di contenuti alternativi per script, applet ed oggetti di programmazione.

Tramite la Barra dell'accessibilità è possibile utilizzare:

### **Informazioni**

*Identifica Applet/Script.* Apre una nuova finestra elencando le applet e gli script presenti all'interno della pagina (Punto di controllo 6.3).

### **Opzioni IE**

*Attiva/Disattiva Javascript.* Attiva/Disattiva Javascript col comando di IE Strumenti>Opzioni Internet>Protezione>Livello personalizzato>Abilita/disabilita Esecuzione Script (Punto di controllo 6.3).

*Attiva/Disattiva ActiveX.* Attiva/Disattiva ActiveX col comando di IE Strumenti>Opzioni Internet>Protezione>Livello personalizzato>Abilita/disabilita Controlli e plug-in ActiveX (Punto di controllo 6.3).

Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 16

Enunciato: Garantire che i gestori di eventi che attivano script, applet o altri oggetti di programmazione o che possiedono una propria specifica interfaccia, siano indipendenti da uno specifico dispositivo di input.

Riferimenti WCAG 1.0: 6.4, 9.2, 9.3

Riferimenti Section 508: 1194.22 (l), 1194.22 (m)

### Motivazione

---

L'errore più comune da parte degli sviluppatori è assegnare una determinata funzionalità ad un determinato evento dipendente da un determinato dispositivo di input. L'esempio più classico è l'uso di comandi legati al mouse (clic, doppio clic) che di fatto diventano ingestibili dagli utenti che non possono utilizzare il mouse, ad esempio, utenti non vedenti che navigano tramite lettori di schermo o utenti che utilizzano periferiche di navigazione sprovviste di mouse. Non tutti gli utenti navigano il Web tramite interfaccia grafica, utilizzando il mouse come periferica di input: molti utenti - anche a causa di disabilità - utilizzano la tastiera, tastiere alternative o comandi vocali per accedere a collegamenti ipertestuali, per accedere o inviare dati da un modulo, ecc. È quindi importante che qualsiasi elemento contenuto nella pagina sia interagibile tramite tastiera in modo che sia fruibile anche da sistemi assistivi di navigazione (oltre all'uso del mouse).

### Implementazione

---

Il requisito chiede di garantire la fornitura di informazioni che non impediscano ad alcune categorie di utenti di poter fruire dei contenuti a causa di impostazioni legate a particolari dispositivi di input per oggetti di programmazione contenuti in una pagina web. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti. Sarà quindi richiesto di verificare l'utilizzo di oggetti di programmazione e, ove individuati, garantire l'accessibilità indipendentemente dal dispositivo di input utilizzato.

## Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

### **Punto di controllo 6.4 (Priorità 2). Garantire che i gestori di eventi per gli script e le applet siano indipendenti dai dispositivi di input**

Un gestore di eventi, o event handler, è un insieme di istruzioni che vengono richiamate quando un particolare evento viene soddisfatto (per esempio, il movimento del mouse, la pressione di un tasto, il caricamento della pagina, e così via). In HTML 4.01 e XHTML gli event handler associati agli elementi vengono gestiti dai seguenti attributi, ordinati per tipo di evento.

```
onload onunload
onclick ondblclick
onmousedown onmouseup onmouseover onmousemove onmouseout
onfocus onblur
onkeypress onkeydown onkeyup
onsubmit
onreset
onselect
onchange
```

Alcuni di questi eventi, come `onsubmit`, sono indipendenti dai dispositivi di input, altri invece richiedono l'utilizzo del mouse (tutto il gruppo `onmouse..` e `onclick`). Altri ancora richiedono l'utilizzo della tastiera (`onkey..`). Se fosse necessario usare eventi che richiedano interazione all'utente, bisogna garantirne l'esecuzione, che l'utente utilizzi una periferica di puntamento, una tastiera o un emulatore di tastiera. È quindi necessario definire un equivalente comando da tastiera per ogni comando tramite mouse: i comandi da tastiera vengono spesso utilizzati per le tecnologie assistive e possono essere attivati anche tramite comandi vocali.

Possiamo quindi affiliare alcuni eventi nel modo seguente:

- utilizzare `onmousedown` con `onkeydown`;
- utilizzare `onmouseup` con `onkeyup`;
- utilizzare `onclick` con `onkeypress`.



Nota: non esiste un comando da tastiera equivalente al doppio clic (`ondblclick`) in HTML 4.01 e XHTML.

Il codice seguente è un esempio mostra come usare correttamente gli eventi, utilizzando JavaScript per aprire una nuova finestra. Se JavaScript non è disponibile, selezionando il collegamento la nuova pagina sarà aperta nella stessa finestra mentre se è attivo il clic sul collegamento ipertestuale o l'uso di qualsiasi tasto (con il focus sul collegamento ipertestuale) aprirà una nuova finestra.

```
<a href="nuovapagina.html"
onclick="window.open(this.href); return false;"
onkeypress="window.open(this.href); return false;">Nuova Pagina</a>
```

### **Punto di controllo 9.2 (Priorità 2). Garantire che ogni elemento dotato di una sua specifica interfaccia possa essere gestito in una modalità indipendente dal dispositivo**

Gli elementi con una propria interfaccia (come per esempio degli applet java) devono essere accessibili e compatibili con le tecnologie assistive. Nel caso non sia possibile rendere accessibile l'interfaccia, per esempio per limitazioni legate al tipo di oggetto, è necessario provvedere una versione alternativa: per ulteriori indicazioni su come creare interfacce accessibili per gli oggetti di programmazione è possibile consultare il requisito 17.

### **Punto di controllo 9.3 (Priorità 2). Negli script, specificare gestori di evento logici piuttosto che gestori di evento dipendenti dal dispositivo**

Questo punto di controllo può definirsi una accentuazione di quanto richiesto dal 6.4 con particolare interesse per gli script. Ricordo che i comandi tastiera vengono spesso utilizzati dalle tecnologie assistive e possono essere utilizzati per attivare funzionalità tramite comandi vocali. Rispetto quindi ai comandi visionati nel punto di controllo 6.4, il requisito 9.3 consiglia di utilizzare espressamente i seguenti eventi, considerando comunque che alcuni, come ad esempio `onchange`, possono creare problemi di accessibilità se utilizzati con alcuni elementi nei moduli (form):

```
onload onunload
onfocus onblur
onsubmit
onreset
onselect
onchange
```

La maggior parte dei produttori di tecnologie assistive preferisce quindi appoggiarsi alle caratteristiche di accessibilità dei sistemi operativi sui quali sviluppa le applicazioni, in modo da consentirne la piena interazione e compatibilità. È quindi importante specificare sempre eventi che non siano legati, per esempio, al movimento del mouse ma che abbiano sempre un equivalente tramite comando da tastiera in quanto su tali comandi si basano principalmente le tecnologie assistive.

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la barra dell'accessibilità per:

- verificare la presenza di script, applet e oggetti di programmazione;
- verificare la presenza di eventi dipendenti dal dispositivo di input.

Tramite la Barra dell'accessibilità è possibile utilizzare:

### Informazioni

*Identifica Applet/Script.* Apre una nuova finestra elencando le applet e gli script presenti all'interno della pagina.

### Struttura

*Gestione eventi.* Visualizza la presenza o meno di eventi indipendenti dal dispositivo di input (Punti di controllo 6.4, 9.2 e 9.3).

Visualizza un simbolo di avvertimento accanto ad un elemento, se l'elemento:

- ha un attributo `onmouseover` senza il corrispondente `onfocus`;
- ha un attributo `onclick` senza il corrispondente `onkeypress`;
- è un elemento `<select>` ed ha un attributo `onchange`;
- ha un attributo `onfocus` ma non è in grado di ricevere il focus.

Visualizza un simbolo di informazione accanto ad un elemento, se:

- ha un attributo `onmouseover` con un `onfocus` corrispondente;
- ha un attributo `onclick` con un `onkeypress` corrispondente.

Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 17

Garantire che le funzionalità e le informazioni veicolate per mezzo di oggetti di programmazione, oggetti che utilizzano tecnologie non definite da grammatiche formali pubblicate, script e applet siano direttamente accessibili.

Riferimenti WCAG 1.0: 8.1

Riferimenti Section 508: 1194.22 (l), 1194.22 (m)

### Motivazione

---

Tutti i programmatori dovrebbero garantire l'accessibilità dei loro programmi, specialmente se fruibili dal grande pubblico tramite un canale di comunicazione come Internet. Gli script, le applet e qualsiasi altro oggetto fornito di propria interfaccia deve esser reso compatibile con le tecnologie assistive, come gli i lettori di schermo e gli ingranditori di schermo (screen magnifier). Sarà quindi compito dello sviluppatore di tali oggetti utilizzare le tecniche di accessibilità delle interfacce stabilite dal produttore della tecnologia.

### Implementazione

---

Il requisito chiede di garantire l'accesso di una pagina web tramite tecnologie assistive anche quando sono presenti oggetti di programmazione. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti. Sarà quindi richiesto di verificare l'utilizzo di oggetti di programmazione e, ove individuati, garantire l'accessibilità tramite tecnologie assistive oppure fornire una soluzione equivalente che abbia la stessa finalità dell'oggetto non accessibile. Ad esempio, se si tratta di menu creati con Macromedia Flash bisogna garantire che tali menu siano accessibili, oppure è necessario fornire un contenuto alternativo all'oggetto. Nel caso, ad esempio, di un plug-in per un CMS con finalità di generare codice conforme al primo requisito, se non è possibile generare un oggetto accessibile è necessario fornire una soluzione alternativa che garantisca comunque la pubblicazione di contenuti e preferibilmente con la stessa finalità del plug-in (es: generazione di codice conforme al linguaggio di marcatura utilizzato).

Tra gli oggetti vanno considerati anche i plug-in che consentono la visualizzazione di contenuti all'interno del browser e questo significa che se nel sito internet è offerta la possibilità di aprire direttamente dei documenti pdf, rtf, ecc. questi documenti dovranno essere sviluppati seguendo le linee guida per l'accessibilità

definite dal produttore della tecnologia: qualsiasi cosa venga fornito tramite web dovrà quindi essere accessibile, indipendentemente dalla tecnologia utilizzata.

## Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

### **Punto di controllo 8.1 (Priorità 1 se importante, altrimenti priorità 2). Fare in modo che elementi di programmi come script e applet siano direttamente accessibili o compatibili con le tecnologie assistive**

Prendiamo come esempio un'applet (creata con l'elemento `<object>` in quanto l'elemento `<applet>` non fa parte delle raccomandazioni del W3C): se richiede l'iterazione con l'utente (per esempio la facoltà di inserire delle informazioni per effettuare dei calcoli) e queste funzionalità non possono essere rese disponibili in una versione alternativa è necessario rendere l'applet accessibile. Se l'interfaccia non può essere resa accessibile a causa delle limitazioni del tipo di oggetto è importante garantire una funzionalità equivalente che fornisca all'utente la possibilità di ottenere la medesima informazione o funzionalità/obiettivo dell'oggetto. Nel caso si utilizzi per esempio Macromedia Flash MX, il prodotto contiene alcune funzionalità che consentono di rendere maggiormente accessibile il contenuto generato in Flash (come per esempio la possibilità di inserire l'attributo `alt` per le immagini) ma questo di fatto non è sufficiente per definire i contenuti in Flash "accessibili". Alcune indicazioni su come rendere maggiormente accessibile il contenuto di Macromedia Flash sono disponibili sul sito di Macromedia  
<<http://www.macromedia.com/macromedia/accessibility/features/flash/hints.html>>.

Spesso questi oggetti si basano sulle API definite dai diversi sistemi operativi. Ad esempio, Macromedia Flash risulta "accessibile" solamente su piattaforma Microsoft Windows con Microsoft Internet Explorer.

Nel caso di Java, invece, è disponibile un pacchetto per la piattaforma Java 2: `javax.accessibility` che rende disponibile una serie di funzionalità  
<<http://java.sun.com/j2se/1.3/docs/api/javax/accessibility/package-summary.html>> per ottimizzare l'accessibilità dell'applicazione.

A titolo di esempio, usiamo l'oggetto `AccessibleContext` che fornisce informazioni sull'accessibilità di un particolare componente: i componenti che non contengono testi modificabili otterranno automaticamente un nome e una descrizione in formato accessibile:

```
AccessibleContext context = email.getAccessibleContext();
context.setAccessibleName("Email");
context.setAccessibleDescription("Indirizzo e-mail");
```

Gli oggetti devono essere indipendenti dalla periferica di input: in Java è possibile utilizzare dei valori mnemonici e degli acceleratori tramite tastiera che consentono l'utilizzo di funzionalità come per esempio l'accesso ad un menu con qualsiasi periferica di input.

```
` Associa una etichetta ad un componente
JLabel label = new JLabel("Nome Utente:");
label.setDisplayedMnemonic('U');
label.setLabelFor(username);
` Valore mnemonico per un menu
JMenu menu = new JMenu("File");
menu.setMnemonic('F');
` Tasto di scelta rapida per una voce di menu
JMenuItem item = new JMenuItem("Apri");
item.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_O,
KeyEvent.SHIFT_MASK));
```

In fase di definizione dei tasti di scelta rapida è necessario controllare sempre che non siano in conflitto con altre funzionalità del sistema operativo. In questo caso consultare le linee guida per le applicazioni in quanto i plug-in installati su personal computer dovrebbero ispirarsi ai requisiti software.

È da ricordare inoltre che i documenti contenenti script devono essere comunque accessibili se gli script non sono supportati o se sono stati disattivati (requisito 15).

## Verifica del requisito

---

L'unico modo per testare gli oggetti è utilizzando una tastiera verificando che le funzionalità e le informazioni siano accessibili e verificando che per le immagini siano presenti dei testi alternativi equivalenti.

È altresì necessario valutare l'oggetto con un lettore dello schermo per verificare la presenza di informazioni non visibili (esempio: identificazione elementi/

oggetti, testi alternativi, ecc.). Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 18

Enunciato: Nel caso in cui un filmato o una presentazione multimediale siano indispensabili per la completezza dell'informazione fornita o del servizio erogato, predisporre una alternativa testuale equivalente, sincronizzata in forma di sotto-titolazione o di descrizione vocale, oppure fornire un riassunto o una semplice etichetta per ciascun elemento video o multimediale tenendo conto del livello di importanza e delle difficoltà di realizzazione nel caso di trasmissioni in tempo reale.

Riferimenti WCAG 1.0: 1.3, 1.4

Riferimenti Section 508: 1194.22 (b)

### Motivazione

---

Il web è diventato un nuovo canale di comunicazione di massa e grazie alla diffusione dei contenuti multimediali in futuro è possibile che superi la forza comunicativa della televisione.

Come per i programmi televisivi, anche per i contenuti multimediali del Web è necessario predisporre una versione utilizzabile anche dagli utenti con difficoltà uditive e/o non udenti, presentando in formato testuale le informazioni necessarie a comprenderne il significato, di cui beneficeranno anche gli utenti non vedenti. Pertanto, gli utenti che a causa di disabilità di tipo visivo o tecnologico (indisponibilità di plug-in, di hardware di ultima generazione, ecc.) non possono visualizzare i contenuti della presentazione multimediale, dovranno poter disporre di una descrizione audio alternativa dei contenuti diversi dai dialoghi. Per qualsiasi azione di interesse per l'utente all'interno della presentazione multimediale (immagini in movimento, localizzazione della scena, dettagli tecnici, ecc.) deve essere predisposta una descrizione di tipo audio. È bene chiarire che il requisito non obbliga a fornire tutte queste alternative per tutti i documenti multimediali ma nel caso tale caratteristica sia necessaria per la fruibilità del servizio.

### Implementazione

---

Il requisito chiede di garantire l'accesso ai contenuti multimediali di una pagina web agli utenti non udenti e agli utenti non vedenti nonché, ove possibile, agli utenti sordomuti tramite linguaggio dei segni. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di



garantire l'accesso ai contenuti alle diverse categorie di utenti. Sarà quindi richiesto di verificare la presenza di contenuti equivalenti per il multimedia.

Multimedia significa contenuti formati da almeno due elementi, solitamente audio e video. Il contenuto multimediale, vale a dire gli elementi che contengono audio e video, è uno dei settori dove l'accessibilità è un tabù.

Mentre, per esempio, è normale vedere i sottotitoli nei programmi televisivi e consideriamo necessaria la sottotitolazione in lingua italiana dei film in inglese o altri linguaggi, spesso non si considera che a un utente non udente è necessario fornire l'equivalente testuale di un filmato sul Web, e a un utente non vedente deve essere fornito un contenuto equivalente (testuale o audio) che rappresenti i contenuti visivi. Il multimedia è ormai parte integrante di molti siti e diversi servizi forniti tramite internet: nei siti di informazione, di intrattenimento e soprattutto nella formazione in linea (e-learning) vengono utilizzati sempre più diffusamente contenuti multimediali.

## Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

### **Punto di controllo 1.3 (Priorità 1). Fino a quando i programmi utente non potranno leggere automaticamente l'equivalente testuale del filmato di una presentazione multimediale fornire una descrizione audio delle informazioni essenziali**

Grazie alla diffusione delle linee ad alta velocità (ISDN, DSL, CDN e altre), sui siti Web è sempre più frequente l'uso di filmati e presentazioni multimediali. Per soddisfare il punto di controllo 1.3 della Priorità 1 delle linee guida anche questi elementi devono essere resi accessibili fornendo una versione audio equivalente dei contenuti essenziali finché i programmi utente non si adegueranno e saranno in grado di leggere automaticamente l'equivalente testuale. Oggi i maggiori media player sul mercato sono in grado di farlo e pertanto è sufficiente fornire:

- una trascrizione testuale dei dialoghi (per utenti non udenti);
- una descrizione audio equivalente per i contenuti visivi importanti per la comprensione dei dialoghi (per utenti non vedenti).

Pertanto, gli utenti che a causa di disabilità di tipo visivo o tecnologico (indisponibilità di plug-in, di hardware di ultima generazione, ecc.) non possono visualizzare i contenuti della presentazione multimediale, dovranno poter disporre di una descrizione alternativa. Per qualsiasi azione visiva di interesse per l'utente all'interno della presentazione multimediale (immagini in movimento, localizzazione della scena, dettagli tecnici, ecc.) dovrà quindi essere predisposta una descrizione di tipo audio.

Se non fosse possibile predisporre una versione audio di supporto, è necessario fornire una versione testuale. In questo caso si parlerà quindi di versione alternativa, che dovrà essere studiata per garantire all'utente le stesse emozioni, o quanto meno paragonabili, ottenute dagli utenti che possono fruire dei contenuti visuali originali. Un'ampia analisi ed esempi di applicazione sono disponibili nel libro "Accessibilità: dalla teoria alla realtà" al Capitolo 31, dedicato all'accessibilità dei contenuti multimediali.

### **Punto di controllo 1.4 (Priorità 1). Per ogni presentazione multimediale temporizzata, sincronizzare le alternative equivalenti con la presentazione**

Abbiamo già visto nel punto di controllo 1.3 come sia necessario fornire dei contenuti audio equivalenti per i contenuti visivi al fine di garantire l'accessibilità del multimedia agli utenti non vedenti. Per gli utenti non udenti, invece, si parla sempre di contenuti testuali descrittivi dei dialoghi e questa funzionalità può risultare utile anche agli utenti con disabilità cognitive e/o per utenti la cui lingua principale è differente dalla lingua utilizzata nel filmato (esempio: stranieri, sottotitoli di interviste in lingua straniera, ecc.).

Il problema che ora coinvolge gli sviluppatori dei contenuti è il seguente: come è possibile sincronizzare questi contenuti in modo che il testo venga rappresentato contestualmente al dialogo e la descrizione audio contestualmente all'azione video? Sono disponibili tre formati/linguaggi che supportano l'aggiunta di audio equivalente per i contenuti video e testo sincronizzato per i dialoghi (sottotitoli) e sono: QuickTime (Apple) e Real Player (Real) che utilizzano il linguaggio SMIL (Synchronized Multimedia Integration Language - W3C) e Windows Media (Microsoft) che utilizza il linguaggio Microsoft SAMI. Inoltre, diverse aziende stanno sviluppando sistemi di integrazione per gli utenti non udenti, come la possibilità di integrare i filmati con il linguaggio dei segni (per ora per i soli linguaggi americano e inglese). Il punto chiave di tutte queste tecnologie dovrebbe essere il collegamento a sistemi di marcatura standard come SMIL per l'abbinamento e la temporizzazione degli equivalenti testuali di audio e video.

Procediamo quindi con un esempio reale in cui vengono forniti i sottotitoli per un'intervista che ho effettuato al Ministro On. Lucio Stanca in occasione del primo anniversario dell'approvazione della legge (gli esempi ed il video sono contenuti nel CD-ROM allegato al libro).

Per predisporre i contenuti nei tre formati necessitiamo del seguente materiale:

- il documento multimediale in un formato compatibile con il programma di esecuzione multimediale;
- il documento contenente la sottotitolazione temporizzata nel formato compatibile con il programma di esecuzione multimediale;
- il documento SMIL/SAMI che garantisca la sincronizzazione tra la sottotitolazione temporizzata ed il contenuto del filmato multimediale;
- il codice di marcatura rispettoso del Requisito 1 per l'inclusione dell'oggetto multimediale.

Innanzitutto è necessario porsi la domanda: qual'è il formato più idoneo alla rappresentazione dei contenuti multimediali? Dai test effettuati da esperti di IWA/HWG risulta che il formato maggiormente supportato - ovvero che non richiede particolari plug-in ed è eseguibile dalla maggior parte dei media player - è il formato AVI standard. Questo formato però ha il problema della pesantezza dei documenti, in quanto è caratterizzato dalla presenza dei singoli fotogrammi in formato bitmap non compresso. Una buona soluzione è di utilizzare un unico documento multimediale in formato mpeg (standard internazionale) ma spesso i media player necessitano di particolari plug-in per la visualizzazione.

Pertanto, ove possibile, è consigliato utilizzare i formati nativi dei vari media player presentando quindi all'utente l'oggetto multimediale ottimizzato per le sue potenzialità.

Relativamente al contenuto testuale equivalente, è necessario chiarire che ogni Media Player ha un suo formato di documento:

- Quicktime: QuickTimeText (.qt)
- Real Player: RealText (.rt)
- Microsoft Windows Media: Sami (.smi)

Solamente Microsoft Windows Media, che a differenza degli altri formati non utilizza SMIL, organizza tutto (testi e sincronizzazione) in un unico documento.

La domanda che sicuramente sorgerà spontanea ora è: come si possono creare tali documenti? Una serie di prodotti che consentono la creazione di sottotitoli e

descrizioni equivalenti è disponibile in un sito in lingua inglese <<http://www.captions.org/softlinks.cfm>>, completamente dedicato al captioning. Negli esempi seguenti useremo MAGpie <<http://ncam.wgbh.org/Webaccess/magpie/>> versione 2.01, un editor specializzato nella titolazione e disponibile gratuitamente che consente di eseguire l'operazione di titolazione con qualsiasi formato multimediale (la versione precedente del prodotto, MAGpie 1.0, poteva essere usata solamente con Windows Media Player). Consiglio vivamente di utilizzare la nuova versione perché consente di operare anche con i documenti Real Video, che non sono convertibili in formato AVI e quindi non gestibili con la precedente versione del programma.

All'interno del CD-ROM allegato al libro è presente la versione 2.01 di MAGpie disponibile sia per ambiente Microsoft Windows che per Macintosh OSX. E' necessario inoltre scaricare il programma QuickTime Player e la Java Virtual Machine seguendo le indicazioni riportate in seguito, in quanto per questioni di licensing non è stato possibile inserirle nel CD-ROM allegato.

Per quanto riguarda la trascrizione testuale, MAGpie consente di importare il testo che verrà utilizzato per la titolazione. Per velocizzare l'ottimizzazione è necessario seguire alcune regole che migliorano l'importazione del testo.

Quando si importa del testo, MAGpie lo colloca in una tabella e il programma determina quali informazioni vadano posizionate in una cella e quali invece debbano essere suddivise in celle diverse analizzando la formattazione del testo:

- l'invio a capo comunica al programma di posizionarsi nella linea successiva della stessa titolazione;
- il doppio invio a capo è interpretato da MAGpie come richiesta di creazione di una nuova titolazione.

Per esempio, il codice seguente riporterà in MAGpie le prime due righe del testo nella stessa cella, mentre le successive verranno inserite in una cella diversa:

```
Allora innanzitutto  
voglio rivolgere un saluto a tutti i partecipanti  
a questo importante convegno.
```

```
Esprimere anche il mio  
rammarico per non essere con Voi  
a celebrare questa data così importante.
```

Nelle scene in cui sono presenti pause nei dialoghi, è necessario nascondere il testo della titolazione relativa al dialogo precedente. Per far comprendere a MAGpie questa situazione, è sufficiente inserire due righe vuote dopo l'ultimo dialogo, in modo che il programma possa riconoscere una titolazione vuota e consenta di inserire delle opzioni (per esempio, la durata).

Lo stesso accorgimento va applicato al termine del testo, in quanto MAGpie deve eliminare l'ultima titolazione.

Una volta avviato MAGpie, è possibile quindi selezionare il video ed importare il testo. Tramite dei pulsanti disponibili sulla barra degli strumenti è possibile quindi avviare il filmato ed inserire il punto di inizio e di fine del dialogo: MAGpie inserirà automaticamente i valori temporali come indicato nell'immagine qui di seguito riportata.



Figura 4.6 Interfaccia di MAGpie.

In dialoghi in cui sono presenti più persone è possibile specificare il nome della persona a cui affiliare il dialogo inserendone il nome nel campo "speaker" al fine di consentire la visualizzazione del nome accanto al dialogo. In questo caso il multimedia si dimostra maggiormente accessibile rispetto ai DVD in cui nei sottotitoli i dialoghi per le diverse persone vengono rappresentati in colori diversi, rendendo quindi l'informazione legata ad uno specifico colore.

Con MAGpie è possibile quindi salvare i sottotitoli al termine del lavoro nei tre diversi formati. Ecco di seguito un esempio relativo al video presente nel CD-ROM in formato RealText.

```
<window bgcolor="#000000" wordwrap="true"
duration="0:00:45.67" width="80"
type="generic" extraspaces="use" >
<font size="2" face="Arial"
color="#FFFFFF" bgcolor="#000000">
<center>
<time begin="00:00:04.68"/><clear/>Allora innanzitutto
voglio rivolgere un saluto a tutti i partecipanti
a questo importante convegno.
<time begin="00:00:10.11"/><clear/>
<time begin="00:00:10.12"/><clear/>Esprimere anche il mio
rammarico per non essere con Voi
a celebrare questa data così importante.
<time begin="00:00:18.56"/><clear/>
</center>
</font>
</window>
```

Lo stesso testo, esportato in formato QuickTime Text, darà il seguente risultato:

```
{ QTtext } { timescale:100 } { font:Arial } { size:18 } {
backColor:0,0,0}
{ textColor:65535,65535,65535 } { width:320 } { justify:center}

[00:00:04.68]
Allora innanzitutto voglio rivolgere
un saluto a tutti i partecipanti
a questo importante convegno.

[00:00:10.11]

[00:00:10.12]
Esprimere anche il mio
```

rammarico per non essere con Voi  
a celebrare questa data così importante.

[00:00:18.56]

Come abbiamo già detto Microsoft Windows Media esporta direttamente il testo  
nel documento SAMI, di seguito riportato.

```
<SAMI>
<HEAD>
<STYLE TYPE="text/css">
<!--
P {
font-size: 18pt;
font-family: Arial;
font-weight: normal;
color: #FFFFFF;
background: #000000;
text-align: center;
padding-left: 5px;
padding-right: 5px;
padding-bottom: 2px;
}
.Captions { Name: Captions; lang: IT_IT_CC; }
#Source {
font-size: 12pt;
font-family: Arial;
font-weight: normal;
color: #FFFFFF;
background: #000000;
text-align: left;
margin-bottom: -14pt;
}
-->
</STYLE>
</HEAD>

<BODY>
<SYNC Start="4677">
<P Class="Captions" ID="Source">&nbsp;&nbsp;&nbsp;</P>
<P Class="Captions">Allora innanzitutto voglio
rivolgere un saluto a tutti i partecipanti
a questo importante convegno.</P>
</SYNC>
```

```

<SYNC Start="10110">
<P Class="Captions" ID="Source">&nbsp;</P>
<P Class="Captions">&nbsp;</P>
</SYNC>
<SYNC Start="10120">
<P Class="Captions" ID="Source">&nbsp;</P>
<P Class="Captions">Esprimere anche il mio rammarico
per non essere con Voi
a celebrare questa data così importante.</P>
</SYNC>
<SYNC Start="18562">
<P Class="Captions" ID="Source">&nbsp;</P>
<P Class="Captions">&nbsp;</P>
</SYNC>

</BODY>
</SAMI>

```

Riporto ora un esempio del codice SMIL generato da MAGpie per Real Player. Come si noterà il codice è conforme a SMIL 1.0 e definisce una serie di impostazioni tra cui l'area di visualizzazione, il documento video di riferimento ed il documento RealText contenente i sottotitoli.

```

<?xml version="1.0" encoding="UTF-8"?>
<smil xmlns="http://www.w3.org/TR/REC-smil">
<head>
<meta content="Ministro per L'Innovazione..." name="title" />
<meta content="Roberto Scano" name="author" />
<meta content="IWA/HWG" name="copyright" />

<layout>
<root-layout width="330" height="295"
background-color="black" />
<region top="5" width="320" height="200" left="5"
background-color="black" id="videoregion" />
<region top="205" width="320" height="80" left="5"
background-color="black" id="textregion" />
</layout>
</head>

<body>
<par>
<video region="videoregion" src="stanca.avi" />
<textstream region="textregion" src="stanca.it_IT.real.rt" />

```



```
</par>
</body>

</smil>
```

Nel caso si desiderino utilizzare le estensioni Real per SMIL 2.0, è necessario dichiararle nel documento:

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language"
xmlns:rn="http://features.real.com/2001/SMIL20/Extensions">
```

Per QuickTime, la versione SMIL 1.0 è leggermente diversa in quanto aggiunge un namespace proprietario di Apple.

```
<?xml version="1.0" encoding="UTF-8"?>
<smil
xmlns:qt="http://www.apple.com/quicktime/resources/smilextensions"
xmlns="http://www.w3.org/TR/REC-smil" qt:time-slider="true">
<head>
<meta content="Ministro per L'Innovazione..." name="title" />
<meta content="Roberto Scano" name="author" />
<meta content="IWA/HWG" name="copyright" />

<layout>
<root-layout width="330" height="295"
background-color="black" />
<region top="5" width="320" height="200" left="5"
background-color="black" id="videoregion" />
<region top="205" width="320" height="80" left="5"
background-color="black" id="textregion" />
</layout>
</head>

<body>
<par>
<video region="videoregion" src="stanca.avi" />
<textstream region="textregion" src="stanca.it_IT.qt.txt" />
</par>
</body>

</smil>
```

Ora che abbiamo a disposizione i documenti richiesti per la pubblicazione, non resta che inserire gli oggetti all'interno della pagina web. Di seguito porterò alcuni

esempi di inserimento di oggetti supportati da alcuni browser: si riscontrano parecchie problematiche nella visualizzazione dei sottotitoli nelle attuali versioni dei browser della famiglia Mozilla nonché in Safari per MAC. La maggiore problematica risulta la visualizzazione all'interno dei browser (tramite l'elemento <object>) mentre la visualizzazione diretta nei media player garantisce una maggiore rappresentazione dei testi sincronizzati.

In molte guide on-line dedicate al captioning (ovvero alla generazione dei sottotitoli per i dialoghi e dei contenuti audio equivalenti per i contenuti video) si consiglia di non inserire oggetti all'interno del browser ma di fornire all'utente un collegamento (link) al documento SAMI/SMIL in modo da consentirne l'apertura diretta.

Per i tre formati è necessario procedere come segue:

- QuickTime. È possibile richiamare direttamente il file SMIL (ad esempio: video\_quicktime.smi).
- Real Player. L'applicazione non è in grado di avviare direttamente il file SMIL e necessita quindi di un documento .ram (o .rpm se si esegue all'interno di una pagina web (<<http://service.real.com/learnnav/rfsmil2.html>> in cui inserire l'indirizzo del documento (se in locale: file://video\_real.smi)).
- Windows Media Player.

È possibile avviare direttamente il video indicando un percorso nomevideo.avi?SAMI=video\_sami.smi. Per Windows Media in alternativa è possibile utilizzare un file con estensione .asx il cui codice è riportato qui di seguito.

```
<asx version="3.0">
<entry>
<ref href="nomevideo.wmv?SAMI=nomevideo.smi" />
</entry>
</asx>
```

È chiaro che il problema attuale è inviare al browser una "comunicazione" per garantire l'apertura del corretto applicativo: come possiamo informare il browser sulla tipologia di applicativo da avviare? Se un browser identifica un documento .smi, come lo avvierà? Con QuickTime, Real Player, Windows Media Player o con altra applicazione configurata dall'utente per l'apertura dei documenti .smi? Una possibilità è offerta dall'attributo "type" previsto per i collegamenti ipertestuali: tramite il valore contenuto in tale attributo si informa il browser della tipologia di

contenuto (MIME) del documento di destinazione. In questo caso il browser dovrebbe identificare l'applicazione idonea alla riproduzione del contenuto multimediale, ovvero l'applicazione che l'utente ha configurato per la visualizzazione del tipo di documento MIME.

```
<ul>
<li><a href="realmedia.ram"
type="application/vnd.rn-realmedia">Real</a></li>
<li><a href="video_quicktime.smi "
type="video/quicktime">Quicktime</a></li>
<li><a href="stanca.avi?SAMI=video_sami.smi "
type="application/x-mpplayer2">Windows Media</a></li>
</ul>
```

Purtroppo però, per questa soluzione, se si utilizzano tre diversi documenti con estensione .smi (SMIL per QuickTime e Real, SAMI per Windows Media) il supporto dei browser è nullo ed avvierà il contenuto multimediale utilizzando il programma utente per cui l'ambiente operativo ha configurato l'esecuzione della tipologia di documento riconosciuta come ".smi".

Se il committente invece richiede l'inserimento di oggetti multimediali all'interno della pagina web è bene preannunciarli che non esiste la possibilità di rendere i contenuti fruibili da diversi browser anche operativi nello stesso sistema operativo: il supporto per i sottotitoli con i tre sistemi è ottimizzato per ambiente Microsoft Windows mentre per gli altri ambienti operativi risulta ottimizzato per il formato QuickTime e Real (ambiente MAC) oppure scarsamente implementato (ambiente Linux).

Innanzitutto è da chiarire che la garanzia di apertura del corretto plugin avviene solamente in ambiente Microsoft Windows in cui l'elemento <object> richiede l'attributo CLSID: tramite questo attributo il browser identifica l'applicazione richiesta dallo sviluppatore al browser per eseguire il contenuto multimediale. La versione per altri browser, rappresentata con i commenti condizionali di IE (che consentono di riprodurre del codice se corrisponde o meno ad una determinata versione di Internet Explorer) utilizza l'attributo "type" e quindi affida la visualizzazione all'applicazione predefinita dall'utente all'interno del browser per la gestione di tale tipologia di documento. Al fine di rispettare il requisito 3, è necessario inoltre prevedere un equivalente testuale per l'oggetto e in questo caso essendo presenti dei sottotitoli è richiesta la disponibilità in formato testuale. Come abbiamo visto all'inizio dell'analisi di questo punto di controllo, con MAGpie è possibile esportare il captioning anche in formato testuale: in questo modo possiamo inserire un collegamento al fine di ottenere la trascrizione dei sottotitoli in formato testuale oppure è necessario fornire nella stessa modalità una descrizione (riassunto) dei contenuti del filmato.

Vediamo innanzitutto un codice cross-browser per Microsoft Windows Media.

```
<p>
<object id="MP" width="320" height="240"
classid="CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6">
<param name="URL" value="stanca.wmv" />
<param name="SAMIFileName" value="stanca.smi" />
<param name="captioningID" value="cc" />
<p>Intervento del Ministro per l'Innovazione...</p>
<!--[if !IE]> <-->
<object type="application/x-mplayer2" id="MP2" width="320"
height="240">
<param name="showcontrols" value="-1" />
<param name="SAMIfilename" value="video_sami.smi" />
<param name="src" value="stanca.avi" />
<param name="captioningID" value="cc" />
<p>Intervento del Ministro per l'Innovazione...</p>
</object>
<!--> <![endif]-->
</object>
</p>
<p><a href="trascrizione.htm">Trascrizione dei sottotitoli</a></p>

<div id="cc">
</div>
```

In questo modo all'interno dell'elemento <div> con id="cc" saranno visualizzati i sottotitoli. Windows Media Player è l'unico applicativo che consente la possibilità di utilizzare i sottotitoli all'interno di un elemento della pagina HTML e che consente quindi di poter gestire tramite fogli di stile la visualizzazione e la gestione dei caratteri. Secondo Microsoft, il controllo multimediale di Windows Media è supportato dalle versioni 4.7, 6.2 e 7.0 di Netscape Navigator, di cui le versioni 6.2 e 7.0 richiedono inoltre la Java 2 Runtime Engine (J2RE) versione 1.3.x o successiva. Da test effettuati ciò corrisponde al vero ma limitatamente alla visualizzazione del video: per quanto riguarda i sottotitoli vengono rappresentati solamente in Internet Explorer. Inoltre è da segnalare un "baco" relativo alle versioni alternative ad Internet Explorer: è necessario indicare il percorso completo del video e del documento SAMI altrimenti il plug-in non riesce ad identificarne il percorso e quindi ad eseguirne il contenuto.

Passiamo quindi ad un esempio di codice cross-browser per QuickTime.

```
<p>
<object id="QT" classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"
```

```

width="320" height="280">
<param name="src" value="video_quicktime.smi" />
<param name="autoplay" value="false" />
<param name="controller" value="true" />
<p>Intervento del Ministro per l'Innovazione...</p>
<!--[if !IE]> <-->
<object id="QT" data="video_quicktime.smi"
type="video/quicktime" width="320" height="280">
<param name="autoplay" value="0" />
<param name="controller" value="-1" />
<p>Intervento del Ministro per l'Innovazione...</p>
</object>
<!--> <![endif]-->
</object>
</p>
<p><a href="trascrizione.htm">Trascrizione dei sottotitoli</a></p>

```

Dai test effettuati in ambiente Windows risulta che la visualizzazione dei sottotitoli internamente al video avviene sia per Microsoft Internet Explorer che per i browser della famiglia Mozilla. Con Opera, invece, se il plug-in non viene identificato è necessario installare l'apposito plugin disponibile nel sito di Apple.

Terminiamo quindi con l'esempio di applicazione del codice per Real Player. Ciò che ho maggiormente apprezzato di Real Player è la possibilità di ingrandire i caratteri del testo dei sottotitoli: ingrandendo la finestra di visualizzazione i caratteri vengono ridimensionati in modo proporzionale. In questo caso quindi consiglio la visualizzazione direttamente nel programma Real Player del filmato per aumentare la fruibilità dei testi dei sottotitoli che non risultano ridimensionabili nell'esecuzione come oggetto all'interno della pagina.

Per garantire la gestione dell'esecuzione del contenuto i lettori multimediali di Real all'interno di Internet Explorer necessitano della creazione di due oggetti:

- un oggetto contenente il video;
- almeno un oggetto contenente i controlli di navigazione.

```

<p>
<object id="RM0"
classid="clsid:CFCDA03-8BE4-11cf-B84B-0020AFBCCFA"
width="320" height="180">
<param name="src" value="video_real.smi" />
<param name="controls" value="ImageWindow" />
<param name="console" value="Clip1" />
<param name="autostart" value="false" />

```

```

</object>
<br />
<object id="RM1"
classid="clsid:CFCDA03-8BE4-11cf-B84B-0020AFBCCFA"
width="320" height="80">
<param name="src" value="video_real.smi" />
<param name="console" value="Clip1" />
<param name="controls" value="All" />
<param name="AutoStart" value="false" />
<p>Intervento del Ministro per l'Innovazione...</p>
</object>
<!--[if !IE]> <-->
<object id="RM2" data="video_real.smi"
type="application/vnd.rn-realmedia"
width="320" height="80">
<param name="src" value="video_real.smi" />
<param name="console" value="Clip1" />
<param name="controls" value="All" />
<param name="AutoStart" value="false" />
<p>Intervento del Ministro per l'Innovazione...</p>
</object>
<!--> <![endif]-->
</object>
</p>
<p><a href="trascrizione.htm">Trascrizione dei sottotitoli</a></p>

```

Esiste un quarto sistema per la rappresentazione di video: Macromedia Flash. In Macromedia Flash MX è disponibile l'Animation Tool: per i filmati è possibile aggiungere tracce di testo sincronizzandole nella linea temporale, mentre per i contenuti visivi è possibile inserire file audio descrittivi da eseguire su richiesta dell'utente o automaticamente, durante le pause dei sottotitoli. Un altro sistema di pubblicazione consigliato è l'inserimento diretto di documenti multimediali già forniti di titolazione: sarà quindi possibile, per esempio, importare i filmati in formato RealVideo che abbiamo prodotto nei paragrafi precedenti e Flash MX ne riconoscerà le impostazioni SMIL. Infine, per garantire la comprensione del contenuto del filmato Flash è consigliabile creare una schermata informativa da posizionare all'inizio del filmato, in modo che l'utente sappia come orientarsi all'interno del documento.

Una soluzione attualmente disponibile è Hi-Caption SE della HiSoftware. <<http://www.hisoftware.com/hmcflash/>>. Il prodotto è stato ampiamente descritto nel libro "Accessibilità: dalla teoria alla realtà" nel Capitolo 31 al quale rimando per approfondimenti e tecniche di implementazione.



Figura 4.7 Interfaccia di HiCaption in Flash.

Nella lista spagnola di SIDAR.org è stata comunicata la decisione da parte della HiSoftware di rendere gratuito il suo revisore per Flash (AccRepair for Flash) e il visore Hi-Caption Viewer for Flash per contribuire ad incrementare l'accessibilità dei contenuti. Per scaricare il prodotto non è ora necessaria nessuna chiave: una volta scaricato <<http://www.hisoftware.com/downloads/HSFTFlash.zip>> bisognerà solo specificare i codici di licenza che si trovano in un documento incluso nel pacchetto di installazione. L'azienda sta inoltre pensando di offrire in open source questo prodotto con l'appoggio di altre persone: chi desidera appoggiare questa idea può inviare un messaggio a Robert Yonaitis all'indirizzo: [ronaitis@hisoftware.com](mailto:ronaitis@hisoftware.com).

Alle estensioni per Flash MX/8 (le più note sono il già citato Hi-Caption e Captionate) si è aggiunta, nel marzo 2007, l'estensione CCFlash che NCAM offre in relazione a MAGpie, per la generazione di file DFXP.

Esiste naturalmente, in queste soluzioni, un forte vincolo rispetto al software di authoring per la sincronizzazione di file esterni.

Il formato Flash Video (FLV), utilizzato in tutte le piattaforme globali di self-broadcasting, ha il pregio di consentire la massima interoperabilità per gli audiovisivi accessibili diffusi sul Web, un'interoperabilità che gli altri formati audiovisivi per il Web non consentono. Se i filmati sottotitolati reperibili si contano in poche decine, e i filmati audiodescritti sono pressoché inesistenti, una delle cause è individuabile proprio nei gravi problemi di compatibilità crossbrowser e clossplatform dei vari player multimediali per il Web, nella esecuzione dei sottotitoli nei filmati inseriti nelle pagine Web.

Per ovviare a questi problemi, senza vincoli ai vendor per la creazione delle alternative accessibili, all'interno del progetto pilota [www.webmultimediale.org](http://www.webmultimediale.org) - una risorsa dedicata alla multimedialità ideata da Roberto Ellero - è stato sviluppato da Alessio Cartocci, e viene usato in produzione dal gennaio 2007, un player multimediale basato su Flash che presenta caratteristiche di notevole interesse:

- supporto dei formati di sottotitolazione SMIL, DFXP, SAMI, QuickTime txt e Real rt;
- in particolare il **supporto esteso di SMIL con Flash Video** (primo esempio al mondo) consente di predisporre in modo assai semplice, in file esterni sincronizzati, la sottotitolazione multilingue, l'audiodescrizione multilingue e l'interpretazione in Lingua dei segni in un secondo flusso video, all'interno di una finestra ridimensionabile e spostabile per evitare sovrapposizioni col video principale;
- supporto agli screen-reader;
- pieno controllo da tastiera;
- la console consente di attivare e di disattivare le alternative per l'accessibilità.

Un esempio a dimostrazione di queste caratteristiche è disponibile al seguente indirizzo:

<[http://www.webmultimediale.org/SC\\_1.2.5](http://www.webmultimediale.org/SC_1.2.5)>





Figura 4.8 Interfaccia del Flash player di Webmultimediale.org.

La soluzione risponde alle tecniche indicate per il *Success criterion* 1.2.5 delle WCAG 2.0, la nuova versione delle linee guida del W3C per l'accessibilità dei contenuti del Web.

In conclusione riporto un codice cross-browser per la pubblicazione di contenuti Macromedia Flash conformi al requisito. Da test effettuati grazie al supporto di esperti nella lista webaccessibile il risultato ottenuto è tra i migliori ed è stato citato come esempio anche da Joe Clark in una sua raccolta degli esempi di utilizzo dell'elemento object <<http://joelclark.org/access/captioning/bpoc/embed-object.html?WAI>>.

```
<p>
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
width="550" height="400">
<param name="movie" value="demo.swf" />
<p>Descrizione del contenuto</p>
<!--[if !IE]> <-->
<object type="application/x-shockwave-flash"
data="demo.swf" width="550" height="400">
```

```
<param name="movie" value="demo.swf" />
</object>
<p>Descrizione del contenuto</p>
<!--> <![endif]-->
</object>
</p>
<p><a href="trascrizione.htm">Trascrizione dei sottotitoli</a></p>
```

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la barra dell'accessibilità per:

- verificare la presenza contenuti multimediali;
- verificare la presenza di testi alternativi (requisito 3).
- verificare la presenza sottotitoli per i dialoghi e contenuti audio equivalenti per contenuti video;
- verificare la sincronizzazione tra il contenuto multimediale e le alternative testuali equivalenti;
- verificare la presenza di testi descrittivi / riassunti del contenuto.

Tramite la Barra dell'accessibilità è possibile utilizzare:

### Informazioni

*Identifica documenti multimediali (Nuova finestra).* Visualizza riferimenti ai documenti multimediali presenti nella pagina web.

L'esperto, identificati i contenuti multimediali dovrà verificare quindi la presenza dei documenti SAMI o SMIL e, ove presenti, verificarne il contenuto. Dovrà quindi verificare se il contenuto multimediale è importante per la fruibilità delle informazioni oppure se ha finalità di contorno. Nel primo caso, se i contenuti multimediali sono predisposti per l'esecuzione all'interno della pagina web è necessario verificare la conformità del codice della pagina (Requisito 1) confrontando il codice con gli esempi indicati per il punto di controllo 1.4 e controllando quindi l'effettiva sincronizzazione. In caso di apertura dei contenuti multimediali all'interno dei media player è necessario verificare l'attivazione della visualizzazione dei sottotitoli e la relativa sincronizzazione con il contenuto.

Per attivare i sottotitoli in Windows Media Player 9, nel caso non venga visualizzata la sottotitolazione, è necessario selezionare la voce di menu Play >

Didascalie e sottotitoli > Attivo se disponibile. In Windows Media Player 10, invece, tale impostazione è configurabile da Strumenti - Opzioni - Protezione - Mostra didascalie locali se presenti. Per le versioni 7 e 8 di Real Player è necessario selezionare la voce di menu View > Preferences. Verrà visualizzata una finestra di dialogo che permette di eseguire alcune impostazioni. Selezionando la voce Content e nella parte inferiore della finestra sarà possibile selezionare il pulsante Accessibility. Questa operazione aprirà una nuova finestra di dialogo, nella quale è necessario selezionare le caselle di controllo Use accessibility features when available e Show captions, confermando con un clic sul pulsante OK. Utilizzando RealONE Player è necessario selezionare la voce di menu Strumenti > Preferenze. Selezionando la categoria Contenuto, sarà possibile operare sulle "Impostazioni di accesso facilitato" e selezionare entrambe le caselle di controllo, "Usa sottotitoli supplementari quando disponibili" e "Usa audio descrittivo quando disponibile". Da test effettuati risulta che l'audio descrittivo è disponibile solamente se si utilizza il formato Real Audio.



Figura 4.9 Impostazione delle preferenze.

Infine, utilizzando QuickTime la visualizzazione dei sottotitoli è automatica: identificando il documento SMIL, QuickTime avvierà i sottotitoli sia se eseguito all'interno di una pagina web che come programma indipendente.

Tra i tre sistemi analizzati l'unico a non supportare i contenuti audio equivalenti per le azioni video è Windows Media Player di cui però si possono invece apprezzare le caratteristiche di accessibilità dei sottotitoli.

Tutto quanto esposto riguarda contenuti di particolare importanza per la fruibilità del servizio. Nel caso invece di contenuti non importanti, è necessario verificare se sia presente o meno un riassunto del contenuto multimediale disponibile in formato testuale e che, se fornito tramite linguaggio di marcatura, dovrà rispettare i 22 requisiti web.

Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 19

Enunciato: Rendere chiara la destinazione di ciascun collegamento ipertestuale (link) con testi significativi anche se letti indipendentemente dal proprio contesto oppure associare ai collegamenti testi alternativi che possiedano analoghe caratteristiche esplicative, nonché prevedere meccanismi che consentano di evitare la lettura ripetitiva di sequenze di collegamenti comuni a più pagine.

Riferimenti WCAG 1.0: 13.1, 13.6

Riferimenti Section 508: 1194.22 (o)

### Motivazione

---

La chiarezza della destinazione dei collegamenti è utile per tutte le categorie di utenti ma in particolar modo agli utenti con disabilità di tipo visivo e cognitivo: conoscere se il collegamento apre una nuova pagina web di un sito esterno oppure fornire un avvertimento all'utente se il documento di destinazione diverso dalle normali pagine HTML (per esempio un documento PDF) è senz'altro utile al fine di evitare disorientamento e tempo di caricamento inutile. Per lo stesso motivo è necessario garantire la chiarezza del testo del collegamento ipertestuale e la possibilità di saltare i collegamenti ripetitivi nelle varie pagine (ad esempio, l'intestazione della pagina o la barra di navigazione) a diverse categorie di utenti che utilizzano tecnologie assistive o hanno problemi di mobilità.

### Implementazione

---

Il requisito chiede di garantire l'accesso ai contenuti di una pagina web in modo da garantire la possibilità di saltare gruppi di collegamenti ripetitivi e di identificare chiaramente la destinazione dei collegamenti. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti.

### Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

## Punto di controllo 13.1 (Priorità 2)

### Identificare con chiarezza la destinazione di ogni collegamento

Quando creiamo un collegamento ipertestuale (link) è necessario che il testo del collegamento abbia un senso. In una frase come la seguente: "Il testo del disegno di legge sull'accessibilità." può essere rappresentata con dei collegamenti in modalità diverse:

```
<p><a href="testo.html">Il testo del disegno di legge  
sull'accessibilità.</a></p>
```

In questo modo il link risulterà troppo esteso: un collegamento ipertestuale deve essere sintetico in quanto può confondere la lettura e creare degli effetti sgradevoli alla vista all'interno della pagina.

```
<p>Il testo del disegno di legge sull'accessibilità:  
<a href="testo.html">clicca qui</a> per leggerlo.</p>
```

Questa modalità è assolutamente da evitare: dove si trova il "qui"? se sono un utente che naviga con tastiera come clicco su un determinato collegamento? E se sono un utente non vedente? È pur vero che il mio lettore di schermo potrà segnalarmi il collegamento ma di fatto manca la contestualità tra il testo del collegamento e la sua destinazione, rendendo problematica la comprensione per esempio ad utenti con disabilità cognitive.

```
<p>Il <a href="testo.html">testo</a> del disegno di legge  
sull'accessibilità.</p>
```

Questa modalità è la consigliata in quanto pone il collegamento ipertestuale direttamente sulla parola interessata. Se il testo invece non risulta adeguato per presentare una descrizione della pagina di destinazione, è possibile utilizzare l'attributo title:

```
<p>Il <a href="testo.html"  
title="Gli articoli del disegno di legge">testo</a>  
del disegno di legge sull'accessibilità.</p>
```

Nel caso per esempio si effettui un collegamento a dei file da scaricare, è necessario informare l'utente della dimensione del file per dare un'idea sul tempo che impiegherà a scaricare il tutto.

```
<p>Il <a href="testo.zip"
title="documento compresso .zip da 100 Kb">testo</a> del disegno di
legge sull'accessibilità.</p>
```

Sempre per informare l'utente, è necessario che i collegamenti a pagine che si aprono su nuove finestre siano chiaramente notificati all'utente.

```
<p>Il <a href="testo.html" title="[nuova finestra] Gli articoli del
disegno di legge" target="_blank">testo</a> del disegno di legge
sull'accessibilità.</p>
```

È inoltre importante sottolineare che se due collegamenti puntano a pagine diverse ma utilizzano lo stesso testo è necessario utilizzare l'attributo title per fornire due differenti spiegazioni. È chiaro quindi che l'utente che utilizza tecnologie assistive e salta da un collegamento ad un altro deve poter facilmente identificare il significato del link, significato che può essere compreso solamente da un chiaro testo del collegamento o da un attributo "title" di supporto.

### **Punto di controllo 13.6 (Priorità 3)**

**Raggruppare i collegamenti correlati, identificare i gruppi (per i programmi utente) e, fino a quando i programmi utente non lo consentiranno, fornire un modo per saltare il gruppo.**

Il requisito indica di raggruppare i collegamenti consentendo di saltare collegamenti ripetitivi, come ad esempio la barra di navigazione. È sempre bene posizionare il menu di navigazione all'inizio del codice all'interno della nostra pagina web, magari inserendo anche una barra di navigazione ed i menu contestuali. È altresì vero che gli utenti non vedenti, che utilizzano i lettori di schermo solitamente si spostano all'interno di una pagina saltando da collegamento a collegamento: questo accadrebbe per ogni pagina visitata se non venisse offerta la possibilità di saltare direttamente al contenuto. Se i collegamenti sono raggruppati secondo una logica (per esempio, le barre di navigazione) è possibile definire un collegamento ipertestuale che ne consenta il superamento: questo significa offrire la possibilità anche agli utenti non vedenti di saltare la lettura di una parte del testo, così come qualsiasi utente vedente farebbe nel caso di ripetizione di testi.

Considerando che spesso la prima voce interna a una pagina Web riguarda proprio la barra di navigazione, se raggruppiamo questi collegamenti in una unità logica alcune tecnologie assistive consentono di saltare direttamente il blocco. Per le tecnologie assistive che invece non consentono tale possibilità, è necessario definire un collegamento che permetta di saltare a un'altra area del contenuto.

Nell'esempio seguente i collegamenti vengono raggruppati nell'elemento `<map>` e la prima opzione consente di saltare direttamente all'inizio del contenuto.

```
<map title="Barra di Navigazione" id="BarNav">
<p>
[<a href="#contenuto">Passa al Contenuto</a>]
<a href="/index.html">Pagina Iniziale</a> -
<a href="/prodotti.html">Prodotti</a> -
<a href="/about.html">Informazioni</a> -
<a href="/contatti.html">Contatti</a>
</p>
</map>
<a id="contenuto"></a>
```

È inoltre possibile utilizzare i fogli di stile per nascondere questo collegamento in modo che sia fruibile esclusivamente dai lettori di schermo.

```
<div class="nascosto">
<a href="#contenuto">vai direttamente al contenuto</a>
</div>
```

È sufficiente inserire il seguente selettore di classe nel foglio di stile, in modo da rendere subito operativo il codice sopra riportato rendendolo quindi "visibile" agli utenti che navigano con sistemi di lettura dello schermo.

```
.nascosto {
width:0;
position:absolute;
height:0;
overflow:hidden;
top:-200em;
}
```

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la Barra dell'accessibilità per:

- identificare i collegamenti ipertestuali;
- verificare la leggibilità dei collegamenti ipertestuali e degli eventuali attributi "title";
- verificare la presenza di aree ripetitive tra le varie pagine;



- verificare la presenza di codice di marcatura che consente di saltare le aree ripetitive.

Tramite la barra dell'accessibilità è possibile utilizzare:

### **Informazioni**

*Visualizza titoli.* Consente di porre graficamente accanto ad ogni collegamento ipertestuale il contenuto dell'attributo `title`, al fine di controllarne la corretta affiliazione e la chiarezza (Punto di controllo 13.1).

*Elenco dei collegamenti ipertestuali (Nuova finestra).* Visualizza un elenco dei collegamenti ipertestuali presenti nella pagina corrente con relativo titolo, url ed attributo `title`, ove presente (Punto di controllo 13.1).

Seguendo le indicazioni del punto di controllo 13.6 l'esperto dovrà quindi verificare se nel codice della pagina web sono state inserite delle caratteristiche che consentono di saltare i collegamenti ipertestuali.

Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 20

Enunciato: Nel caso che per la fruizione del servizio erogato in una pagina è previsto un intervallo di tempo predefinito entro il quale eseguire determinate azioni, è necessario avvisare esplicitamente l'utente, indicando il tempo massimo consentito e le alternative per fruire del servizio stesso.

Riferimenti WCAG 1.0: 7.4, 7.5

Riferimenti Section 508: 1194.22 (p)

### Motivazione

---

Gli utenti che utilizzano lettori di schermo oppure utenti con difficoltà di lettura possono riscontrare difficoltà nella fruibilità di contenuti web le cui pagine si autoaggiornano o richiedono un determinato tempo di risposta. È quindi necessario avvertire l'utente su tali condizioni consentendo delle alternative per poter fruire del servizio. Un caso tipico è un test di assunzione tramite tecnologie informatiche: un utente non vedente dotato di lettore di schermo dovrà poter fruire dei contenuti senza che la pagina si autoaggiorni (passando alla domanda successiva) con la stessa frequenza degli utenti che non hanno disabilità.

### Implementazione

---

Il requisito chiede di garantire l'accesso ai contenuti di una pagina web in modo da garantire la possibilità di fruire dei contenuti conoscendo il tempo massimo concesso per la lettura e fornendo la possibilità di soluzioni alternative e/o di personalizzazione del tempo di risposta. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti.

### Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

## Punto di controllo 7.4 (Priorità 2)

**Fino a quando i programmi utente non forniranno la possibilità di bloccare l'autoaggiornamento, non creare pagine che si autoaggiornano periodicamente.**

Prendiamo come esempio il caso di un sito di un ente pubblico in che fornisce contenuti con aggiornamento in tempo reale e che quindi richiedono un frequente aggiornamento dei contenuti. L'esempio classico è dato dai risultati elettorali per l'elezione di un Sindaco, di un presidente di Regione, ecc. Per non dover costringere l'utente ad aggiornare la pagina Web spesso viene utilizzato l'elemento meta `http-equiv="refresh"` per definire automaticamente la frequenza di aggiornamento della pagina.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="it" xml:lang="it">
<head>
<title>Titolo Pagina</title>
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-1" />
<meta http-equiv="Refresh" content="60" />
</head>
...
...
</html>
```

Nell'esempio riportato, la pagina si aggiorna ogni 60 secondi, presentando in tempo reale l'elenco delle ultime notizie. Il caricamento automatico della pagina può causare problemi ad alcune categorie di utenti: chi ha bisogno di maggior tempo per leggere i contenuti e chi utilizza lettori dello schermo troverà fastidioso dover ricominciare a leggerli. Come ovviare a questo problema? Innanzitutto è necessario avvisare gli utenti di questa funzionalità di autoaggiornamento, in modo che in ogni caso siano preparati all'evento. Nel caso non sia possibile concedere all'utente il controllo di tale evento consentendo una personalizzazione del refresh della pagina, è preferibile fornire un collegamento a una versione alternativa dove questa funzionalità non sia presente. Ipotizzando di usare codice ASP, con un semplice modulo è possibile consentire all'utente di modificare il tempo di riaggiornamento della pagina:

```
<%
'definisce il valore in secondi del refresh
Dim aggiorna
if request.form("aggiorna") = "" then
```

```

aggiorna = 60
elseif isnumeric(request.form("aggiorna")) then
aggiorna = int(request.form("aggiorna"))
else
aggiorna= 60
end if
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="it" xml:lang="it">
<head>
<title>Titolo Pagina</title>
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-1" />
<meta http-equiv="Refresh" content="<%=aggiorna%>" />
</head>
<body>
<form action="default.asp" method="post">
<label for="aggiorna">Aggiornamento:</label>
<input type="text" id="aggiorna"
value="<%=request.form("aggiorna")%>" />
<input type="submit" value="Aggiorna" />
</form>
...
...
</body>
</html>

```

## **Punto di controllo 7.5 (Priorità 2)**

**Fino a quando i programmi utente non forniranno la capacità di bloccare l'autoreindirizzamento, non usare marcature per reindirizzare le pagine automaticamente. Piuttosto, configurare il server in modo che esegua i reindirizzamenti.**

Nel punto di controllo 7.4 il problema di accessibilità riguardava il caricamento a particolari intervalli di tempo. Il punto di controllo 7.5 invece richiede di non utilizzare gli elementi meta per reindirizzare l'utente ad altre pagine per diverse motivazioni. La prima innanzitutto è che tale funzionalità può disorientare i navigatori del vostro sito Web creando problemi di navigazione, come per esempio la difficoltà di ritornare alla pagina precedente. Per quanto riguarda invece gli utenti che utilizzano tecnologie assistive (come per esempio i lettori di schermo) oppure gli utenti che necessitano maggior tempo per leggere i contenuti di una pagina il caricamento automatico di una nuova pagina ad un

determinato intervallo di tempo causa grossi problemi di fruibilità dei contenuti rendendo perciò la pagina inaccessibile per l'attuale requisito.

Un esempio di codice da non utilizzare è il seguente:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="it" xml:lang="it">
<head>
<title>Titolo Pagina</title>
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-1" />
<meta http-equiv="Refresh" content="3; http://www.w3.org/" />
</head>
```

Noterete che, a differenza del punto di controllo 7.4 l'elemento `http-equiv="Refresh"` accanto al tempo di esecuzione (in questo caso tre secondi) riporta l'indirizzo della nuova pagina dove l'utente verrà automaticamente trasferito (se il browser supporta tale funzionalità). Si sarà facilmente compreso che tutto dipende dal supporto tecnologico del programma utente, ma soprattutto dalle impostazioni del server del proprio hosting provider. Pertanto i consigli che si possono dare agli sviluppatori sono i seguenti:

- configurare il server per utilizzare il codice HTTP 301 e relativi codici HTTP 30x;
- utilizzare dei normali collegamenti alla nuova pagina anziché delle funzionalità automatiche di redirect;
- evitare, ove possibile, l'utilizzo di tali tecnologie in quanto la navigazione deve essere sotto il totale controllo dell'utente.

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la barra dell'accessibilità per:

- identificare se la pagina prevede funzionalità di autoaggiornamento o autoreindirizzamento;
- verificare se sono presenti possibilità di configurare i tempi di autoaggiornamento e di autoreindirizzamento.

Tramite la barra dell'accessibilità è possibile utilizzare:

## Informazioni

- Informazioni Metadata (Nuova finestra). Visualizza l'elenco di tutti gli elementi meta ed il relativo contenuto della pagina corrente al fine di individuare la presenza di meta http-refresh (Punto di controllo 7.4 e 7.5). Se sono presenti, controllare che vi siano funzionalità nella pagina che consentono di variare il tempo di reindirizzamento.

Un caso particolare sono le aree riservate: è necessario controllare la disponibilità di funzioni che consentano all'utente di esser informato sulla durata della sessione utente ed eventualmente della presenza di funzionalità per personalizzare la durata della sessione. Questa ultima possibilità garantisce accessibilità ma può causare problemi di sicurezza in alcuni sistemi informatici.

Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 21

Enunciato: Rendere selezionabili e attivabili tramite comandi da tastiera o tecnologie in emulazione di tastiera o tramite sistemi di puntamento diversi dal mouse i collegamenti presenti in una pagina; per facilitare la selezione e l'attivazione dei collegamenti presenti in una pagina è necessario garantire che la distanza verticale di liste di link e la spaziatura orizzontale tra link consecutivi sia di almeno 0,5 em, le distanze orizzontale e verticale tra i pulsanti di un modulo sia di almeno 0,5 em e che le dimensioni dei pulsanti in un modulo siano tali da rendere chiaramente leggibile l'etichetta in essi contenuta.

Riferimenti WCAG 1.0: non presente

Riferimenti Section 508: non presente

### Motivazione

---

Gli utenti con ridotta capacità motoria, che utilizzano comandi da tastiera, tecnologie in emulazione di tastiera o sistemi di puntamento alternativi, possono riscontrare difficoltà nella selezione e nell'attivazione dei collegamenti ipertestuali, e dei pulsanti dei moduli, quando questi sono eccessivamente ravvicinati. Questo vale anche per gli utenti con difficoltà cognitive legate alla lettura, ipovedenti, e, in generale, per tutte le categorie di utenti che sperimentano difficoltà di accesso a testi e ipertesti se non sono presenti idonee distanze tra i singoli elementi. Il requisito richiede di garantire la fruibilità dei collegamenti ipertestuali, e dei pulsanti dei moduli, impostando una distanza ragionevolmente ampia tra di essi, indicata in 0.5em.

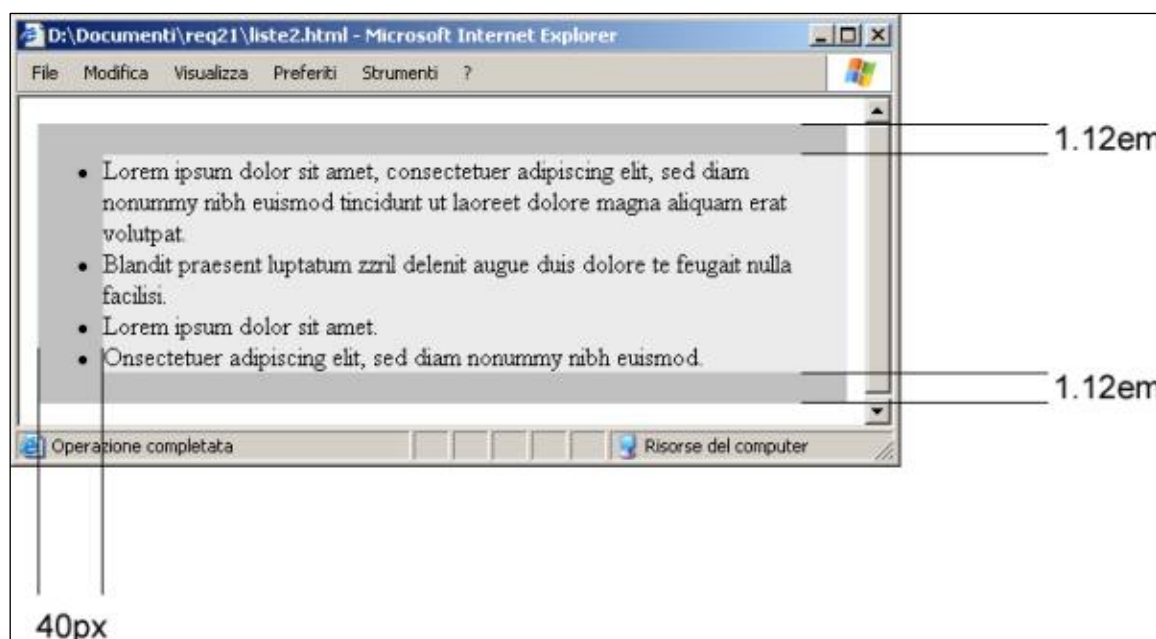
### Implementazione

---

Le liste HTML (marcatori `<ul>`, `<ol>` e `<li>`) rappresentano il metodo corretto per codificare un elenco di collegamenti ipertestuali. Questo vale sia per le liste di link verticali sia per quelle orizzontali. Utilizzando i fogli di stile è possibile modificare l'aspetto di una lista HTML (per esempio eliminando o modificando i glifi e i margini predefiniti, oppure trasformando una lista verticale in una orizzontale) senza con ciò alterare il codice HTML sottostante. In tale modo le liste di link avranno l'aspetto desiderato e saranno fruibili anche nel caso in cui i fogli di stile fossero disabilitati o non supportati (Requisito 11).

L'elemento `<ul>` possiede un margine predefinito superiore e inferiore di 1.12em, un margine destro pari a 0 e il margine sinistro (lo spazio riservato ai glifi) pari a

40px. Queste distanze sono rispettate pressoché da tutti i browser (Mozilla Firefox non applica il margine sinistro, sostituendolo con il padding, e imposta i margini superiore e inferiore a 1em). Non sono previsti margini predefiniti tra le voci di una lista (elementi `<li>`), è infatti il glifo, posto a sinistra delle singole voci, ad indicarne l'inizio, mentre l'interlinea determina lo spazio verticale tra le righe del testo:



**Figura 4.10** Effetto delle spaziature.

Tutti i browser considerano 1.2 come il valore predefinito per l'interlinea del testo (tranne Mozilla Firefox che lo imposta, curiosamente, a 1.25), conformandosi così al limite superiore raccomandato dal W3C per il valore normal della proprietà `line-height` (che può infatti oscillare tra 1.0 e 1.2). Nella sua prima stesura, il requisito indicava per le distanze il valore minimo di 1em. Questo valore era inteso dal legislatore come la misura minima dell'interlinea del testo. 1em è, però, un valore inferiore rispetto a quello predefinito dai browser per l'interlinea. Applicarlo, quindi, significherebbe ridurre la fruibilità delle liste di link verticali anziché incrementarla. Se invece lo ampliassemo, per esempio portandolo a 1.5, otterremmo certamente un maggiore spazio verticale tra le righe del testo, ma non vi sarebbe soluzione di continuità tra le singole voci della lista:



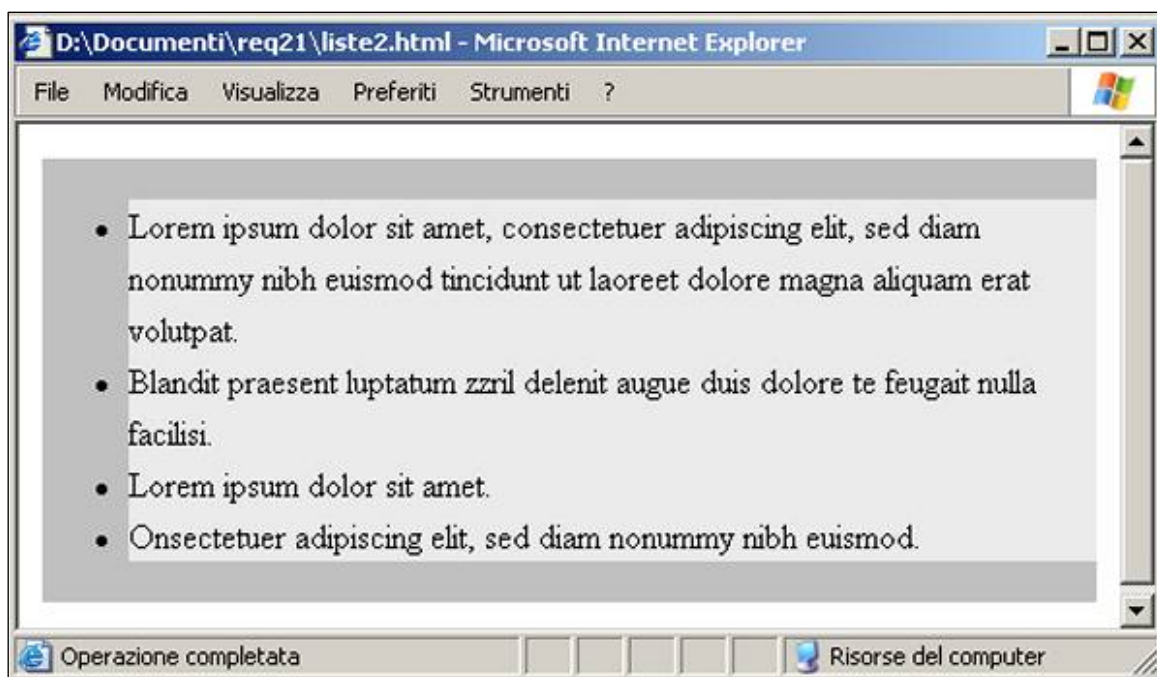
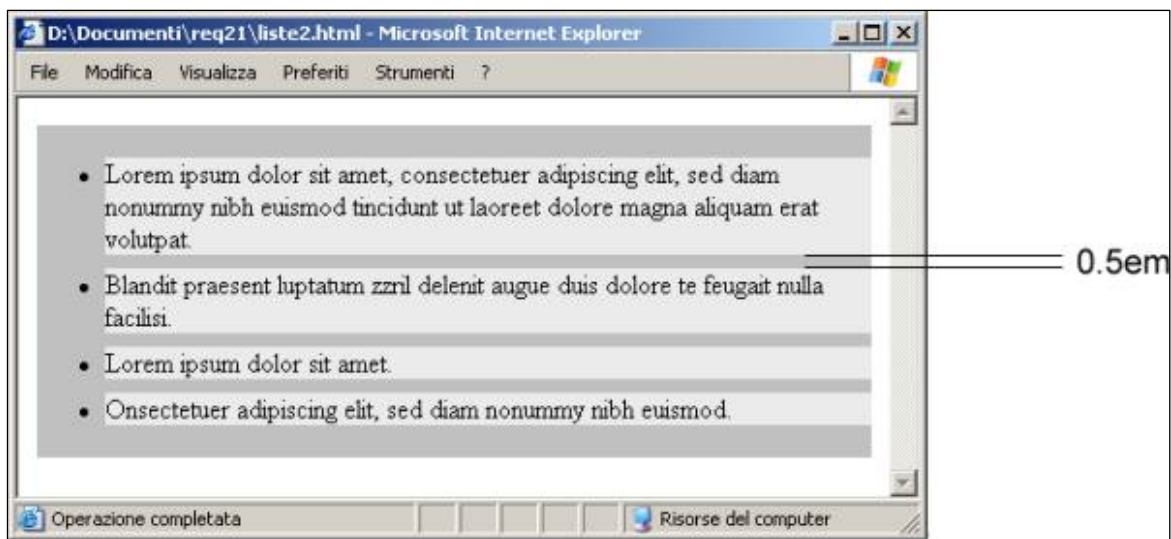


Figura 4.11 Effetto dell'aumento dell'interlinea.

Oltre a ciò, l'interlinea non è evidentemente utilizzabile per distanziare le voci delle liste di link orizzontali, o i pulsanti dei moduli, e un valore minimo di 1em può risultare elevato se associato a proprietà diverse da essa, come per esempio il margine o il padding. Queste riflessioni hanno indotto il legislatore a ridurre il valore minimo a 0.5em non associandolo più all'interlinea del testo. È il margine tra gli elementi, infatti, che va considerato come l'interpretazione corretta dei termini "*distanza*" e "*spaziatura*" presenti nell'enunciato del requisito. Il margine è applicabile sia ad entrambe le tipologie di liste di link, orizzontali e verticali, sia alle distanze tra i pulsanti di un modulo.

Utilizzando il margine, quindi, a prescindere dal valore assegnato all'interlinea del testo (per esempio quello predefinito o un valore superiore), le voci della lista risulteranno sempre adeguatamente distanziate e il testo diviso in blocchi omogenei. Anche se una voce particolarmente lunga si distribuirà su più righe, la separazione logica tra le voci sarà ugualmente garantita:

```
li { margin-top: 0.5em; margin-bottom: 0.5em; }
```



**Figura 4.12** Effetto della regola.

Ricordo che i margini degli elementi `<li>` collassano. Ciò significa che due margini adiacenti non si sommano. Il valore applicato sarà quello più alto tra i due. Nell'esempio precedente i due valori sono identici quindi la misura 'vincente' è esattamente 0.5em.

Chiaramente è possibile impostare esclusivamente il margine superiore, o quello inferiore, come nell'esempio seguente:

```
li { margin-bottom: 0.5em; }
```

Qualora non fosse possibile utilizzare il margine per distanziare le voci delle liste di link, un risultato analogo si ottiene con la proprietà CSS `padding`. Nel caso delle liste verticali, per esempio, è possibile applicare il `padding` superiore e inferiore agli elementi `<li>`. Dato che il `padding` non è soggetto al collassamento, è necessario fare attenzione all'impostazione dei valori. Per ottenere una distanza effettiva di 0.5em tra le voci della lista, il valore del `padding` superiore e inferiore dovrà essere impostato a 0.25em:

```
li { padding-top: 0.25em; padding-bottom: 0.25em; }
```

Anche in questo caso è possibile impostare esclusivamente il `padding` superiore, o quello inferiore, piuttosto che entrambi. In questo caso il valore indicato dovrà essere pari a 0.5em:

```
li { padding-bottom: 0.5em; }
```

In merito alla porzione dell'enunciato che si riferisce alla distanza tra i pulsanti dei moduli: "le distanze orizzontale e verticale tra i pulsanti di un modulo sia di almeno 0,5 em" bisogna aggiungere che, nonostante non sia esplicitamente indicato, è consigliabile considerare questa distanza minima come riferita non solo allo spazio esistente tra i pulsanti, siano essi affiancati orizzontalmente o posti su righe differenti, ma anche alla distanza tra i pulsanti e i campi dei moduli che li precedono o seguono. Non avrebbe molto senso, per gli scopi che si prefigge il requisito, distanziare due pulsanti tra di loro ma non in relazione al campo o all'area di testo (textarea) che eventualmente li precede o li segue.

Infine, la frase *"le dimensioni dei pulsanti in un modulo siano tali da rendere chiaramente leggibile l'etichetta in essi contenuta"* è difficile da implementare, e valutare, a causa della mancanza di un riferimento oggettivo a un valore minimo. Il concetto di chiara leggibilità, com'è facile intuire, è estremamente soggettivo. Il consiglio è di non scendere mai sotto le dimensioni predefinite dai browser per i pulsanti dei moduli. Questa è, in ogni caso, una soluzione approssimativa dato che esiste una certa variabilità nel modo predefinito in cui i vari browser rappresentano i pulsanti. Qualora, quindi, si desiderasse modificare l'aspetto dei pulsanti utilizzando i fogli di stile non si dovranno impostare altezza, larghezza e padding inferiori a quelli predefiniti. Inoltre, per mantenere la conformità al Requisito 6, sarà necessario assicurarsi che il contrasto tra il colore dell'etichetta e lo sfondo del pulsante sia sufficiente.

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà utilizzare strumenti di supporto come la barra dell'accessibilità per:

- Identificare i moduli (form) e i raggruppamenti di collegamenti ipertestuali.
- Identificare gli elementi e le classi associate ad essi.
- identificare se nel foglio di stile vi sono regole che riducono le distanze minime richieste dal requisito.

Tramite la Barra dell'accessibilità è possibile utilizzare:

### **CSS**

*Visualizza gli stili applicati (Nuova finestra):* Visualizza gli stili associati al contenuto posizionato al di sotto del puntatore del mouse.

*Visualizza il foglio di stile (Nuova finestra):* Visualizza il contenuto dei fogli di stile richiamati dalla pagina corrente. A termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)

## Requisito 22

Enunciato: Per le pagine di siti esistenti che non possano rispettare i suelencati requisiti (pagine non accessibili), in sede di prima applicazione, fornire il collegamento a una pagina conforme a tali requisiti, recante informazioni e funzionalità equivalenti a quelle della pagina non accessibile ed aggiornata con la stessa frequenza, evitando la creazione di pagine di solo testo; il collegamento alla pagina conforme deve essere proposto in modo evidente all'inizio della pagina non accessibile.

Riferimenti WCAG 1.0: 11.4

Riferimenti Section 508: 1194.22 (k)

### Motivazione

---

Questo requisito consente di creare delle pagine alternative solo in fase di prima applicazione e nel caso in cui, per una determinata pagina, non sia possibile generare contenuti accessibili. È bene ricordare inoltre che la pagina "solo testo" non significa testo senza formattazione ma una pagina HTML senza elementi di formattazione.

Possiamo considerarlo di fatto un "requisito fantasma", dato che la sua applicazione riguarda esclusivamente i contenuti esistenti: per i nuovi siti internet tale requisito non è applicabile.

### Implementazione

---

Il requisito chiede di garantire l'accesso ai contenuti di una pagina web esistente inaccessibile fornendo una pagina parallela con medesimi contenuti e funzionalità e rispettosa dei 21 requisiti, in modo da garantire la possibilità di fruire dei contenuti. Lo sviluppatore dovrà quindi seguire delle semplici raccomandazioni che si ispirano alle WCAG 1.0 al fine di garantire l'accesso ai contenuti alle diverse categorie di utenti.

### Riferimenti WCAG 1.0

---

Di seguito sono riportati i riferimenti ai punti di controllo delle WCAG 1.0 al fine di fornire allo sviluppatore le informazioni per implementare contenuti conformi al requisito.

## **Punto di controllo 11.4 (Priorità 1)**

**Se, nonostante ogni sforzo, non è possibile creare una pagina accessibile, fornire un collegamento a una pagina alternativa che si riferisca alle tecnologie W3C, che sia accessibile, che contenga informazioni (o funzionalità) equivalenti e sia aggiornata con la stessa frequenza della pagina originale inaccessibile.**

Gli sviluppatori dovrebbero ricorrere a pagine alternative solo quando le altre soluzioni falliscono perché le pagine alternative sono in genere meno aggiornate delle pagine principali. È bene chiarire innanzitutto che pagina alternativa non significa pagina visualizzata con diverso foglio di stile ma di una pagina con stesse informazioni e funzionalità della pagina inaccessibile.

Ipotizziamo un sito di un ente pubblico dove ogni ufficio inserisce dei contenuti in modo autonomo: se 100 operatori inseriscono contenuti non accessibili che richiedono una versione parallela, il costo e le ore lavorative saranno quantomeno raddoppiate e sempre con il dubbio che non tutte le pagine alternative siano aggiornate.

Una pagina non aggiornata può essere frustrante quanto una pagina inaccessibile visto che, in entrambi i casi, l'informazione presentata nella pagina originale non è disponibile. La generazione automatica di pagine alternative può portare a aggiornamenti più frequenti, ma gli sviluppatori devono comunque fare attenzione ad assicurare che le pagine generate abbiano sempre senso, e che gli utenti siano in grado di navigare in un sito seguendo i collegamenti delle pagine primarie, di quelle alternative, o di entrambe (magari posizionando su ogni pagina il collegamento alla versione alternativa). Alcuni esperti di accessibilità consigliano di iniziare la conversione dei vecchi siti tramite versioni alternative solo testo, magari con qualche foglio di stile dedicato a particolari tipi di disabilità.

Personalmente seguendo le indicazioni del W3C sconsiglio tale pratica che rende pesante l'aggiornamento e che di fatto può risultare dannosa all'accessibilità di un sito Web in quanto si utilizza tale modalità di ripiego per "tenere la coscienza pulita" da eventuali incompetenze degli sviluppatori o dei gestori dei contenuti. Prima di ricorrere a una pagina alternativa, riesaminare il progetto della pagina originale: è probabile che rendendola accessibile essa risulti migliore per tutti gli utenti.

Il W3C non parla mai di sito alternativo ma di pagina alternativa: chi crea siti alternativi non rispetta le WCAG 1.0 e nemmeno i principi di sviluppo per l'accesso universale. Inoltre la soluzione "solo testo", spesso utilizzata anche da

siti istituzionali, non è conforme a questa linea guida in quanto non si tratta di pagina equivalente che utilizza le tecnologie W3C ma si tratta dell'ispirazione alla normativa americana Section 508, legge non in vigore nel nostro paese.

Possiamo quindi dare la seguente chiave di lettura al punto di controllo:

- Si parla di pagina, non di sito. Questo è particolarmente importante, in quanto le dichiarazioni di conformità alle WCAG devono essere effettuate per singole pagine.
- "Nonostante ogni sforzo..." significa che lo sviluppatore, dopo aver valutato l'uso di tutte le possibili soluzioni tecniche presentate nelle WCAG per rendere accessibile un contenuto, alza le braccia e si arrende. Se ciò capita per un intero sito web, con creazione di wai.nomesito.it, il committente dovrebbe verosimilmente chiedersi se ha scelto il fornitore adatto.
- Si parla di pagina alternativa "che usi le tecnologie W3C, sia accessibile, contenga informazioni (o funzionalità) e equivalenti". Questo non sembra riscontrabile nei cosiddetti siti paralleli, in quanto molto spesso riportano esclusivamente alcuni contenuti di tipo testuale che non sono "una pagina accessibile che usa le tecnologie W3C".
- Si parla di aggiornabilità della pagina (non del sito).

Riassumendo: il sito alternativo è sviluppato da persone che nonostante ogni sforzo di applicare le linee guida non sono stati capaci (forse per incompetenza) di creare nemmeno una pagina accessibile e hanno ripiegato su una versione solo testo ad alto contrasto per accontentare una fetta del mercato degli utenti con disabilità (in questo caso, limitatamente alla disabilità visiva).

Quindi, chi nonostante tutto decide di creare la "porta di servizio" (con tutti i limiti che abbiamo già descritto) non può dirsi in linea con gli standard del Web, non potrà dire che sta seguendo le WCAG – e neppure la legge 04/2004.

La soluzione più accessibile, nel caso si desideri creare la pagina alternativa per un contenuto non accessibile è semplice e logica: la pagina principale dovrà essere la pagina accessibile con un collegamento alla pagina non accessibile, ossia come già oggi avviene per i siti Web che propongono introduzioni animate con tecnologia Flash dove all'ingresso della pagina Web viene fornita la possibilità di visualizzare la pagina contenente la tecnologia Flash.

È importante ricordare inoltre che il contenuto della pagina alternativa deve comunque rispettare tutti i requisiti dal numero 1 al numero 21.

## Verifica del requisito

---

Il valutatore, al fine di verificare l'applicazione del requisito dovrà:

- Identificare dei collegamenti nella pagina che consentono l'accesso a versioni parallele del tipo "Versione accessibile", "Versione WAI", ecc.
- Analizzare tale pagina rispetto i 21 requisiti presentati in precedenza.

Al termine della verifica dei suddetti punti l'esperto potrà quindi dichiarare la conformità al requisito.

[Torna a Elenco dei requisiti](#)