

# **Manuale**

---

## **Linee guida per l'Accessibilità basate sulla legge 4/2004**

**Versione 2.0 - dicembre 2005**

---

*A cura di: CINECA, Interactive Marketing, ItalianTeam*

## Sommario

Cos'è l'Accessibilità .....	2
Il rapporto tra Accessibilità e Usabilità .....	4
Normativa di riferimento .....	5
Modalità di applicazione della legge 4/2004 .....	6
Un manuale: perché e per chi.....	8
Guida alla lettura.....	9
Linguaggio, DTD e intestazione del documento (requisito n.1).....	10
DA HTML A XHTML .....	10
Gli standard del web .....	14
Intestazione di documento: l'elemento <!DOCTYPE>.....	14
Intestazione di documento: i META tag .....	15
Intestazione di documento: il titolo della pagina .....	15
Contenuto, Struttura, Presentazione .....	16
Liste .....	17
I pop-up.....	18
Frame (requisito n.2).....	20
Inline Frames .....	22
Immagini e animazioni (requisito n.3).....	24
Figure e diagrammi .....	26
Colore (requisito n.4 e n.6) .....	28
Elementi in movimento e lampeggianti (requisito n.5) .....	30
Immagini sensibili (requisito n.7 e n.8) .....	31
Tabelle di dati (requisito n.9 e n.10) .....	34
Fogli di stile (requisito n.11).....	39
Dimensionamento dei caratteri (requisito n.12).....	44
Tabelle per l'impaginazione (requisito n.13) .....	45
Form (requisito n.14) .....	48
Script, applet o altri oggetti di programmazione (requisito n.15, n.16, n.17 e n.20) ....	51
Javascript, DHTML .....	52
Object e Applet .....	54
Plug-in, documenti non HTML.....	56
Flash MX .....	57
File Microsoft Excel .....	59
Creare file Adobe PDF accessibili .....	60
Multimedia (requisito n.18) .....	62
Link ipertestuali (requisito n.19).....	63
Skipping navigation .....	65
Navigazione da tastiera (requisito n.21) .....	66
Attributo accesskey .....	66
Creare una pagina alternativa (requisito n.22).....	68
Verificare il lavoro, ovvero la validazione .....	69
Glossario minimo.....	70
Bibliografia .....	72
Sul web.....	73

## Cos'è l'Accessibilità

Rendere accessibile qualcosa vuol dire renderla disponibile **senza esclusioni**.

Rendere accessibile una pagina web significa quindi progettare in maniera che sia consentito a tutti l'accesso completo alle informazioni (struttura e contenuto).

Significa creare siti dove le modalità di navigazione e tutti gli elementi interattivi eventualmente presenti siano fruibili dagli utenti indipendentemente dalle loro disabilità, indipendentemente dalla tecnologia che essi utilizzano e indipendentemente dal contesto in cui operano mentre accedono al sito.

Moltissimi sono gli utenti che trovano ostacoli nell'uso del web quando non è progettato in maniera accessibile. Seppur in maniera differente, il problema riguarda diversi tipi di disabilità: visiva, motoria, cognitiva, legata all'età (ad esempio, anziani e bambini)...

Ma vanno presi in considerazione anche tutti coloro che si servono di hardware e di programmi poco aggiornati (computer con processori lenti o con poca memoria, monitor in bianco e nero, vecchie versioni di browser...) o di strumenti alternativi più avanzati, come ad esempio tecnologie di assistenza digitale e telefoni cellulari.

Per tutti questi tipi di utenti, la principale cura dell'accessibilità consiste nel progettare pagine indipendenti dalla periferica, non ottimizzate per nessun sistema operativo e per nessun browser in particolare: ovvero pagine con codice standard valido, altamente flessibili nella struttura, non vincolate ad una data larghezza orizzontale e leggibili anche da chi non è in grado di visualizzare i colori.

Esistono tecnologie assistive, in grado cioè di facilitare il superamento di alcuni ostacoli. Ad esempio, gli screen reader (usati in particolare dai non vedenti) che traducono in sonoro il contenuto (struttura, testo e immagini) di una pagina, o particolari hardware progettati per chi ha problemi motori.

Perché il web sia accessibile, quindi, deve utilizzare un linguaggio riconoscibile anche da questi strumenti, proporre strutture e contenuti semplici, esporli in maniera chiara, eliminando dalla pagina ogni "barriera telematica".

L'accessibilità punta alla valorizzazione delle differenze: differenze nelle capacità individuali e tecnologiche, nei modi di fruizione dell'informazione online, nelle tecnologie e negli strumenti di accesso.

Ecco qualche esempio, per comprendere meglio: caratteri troppo piccoli rendono difficoltosa la lettura per chi ha facoltà visive ridotte; possono creare problemi (ad esempio ai daltonici) l'accostamento di taluni colori (come rosso e verde) o la mancanza di contrasto tra colori del testo e dello sfondo; testi troppo lunghi ostacolano chi ha difficoltà motorie; un linguaggio complesso o troppo tecnico discrimina chi non ha le conoscenze adeguate...

Adattarsi agli standard del W3C significa contribuire al raggiungimento di quella "progettazione universale" che, considerato il ruolo e l'importanza che ricopre il web allo stato attuale, è ormai indispensabile.

Rendere una pagina web accessibile non significa lavorare di più, ma in maniera diversa. E non significa nemmeno limitare le possibilità espressive: i limiti di progettazione imposti dall'accessibilità riguardano infatti le modalità di attuazione delle diverse soluzioni tecniche, non il contenuto o l'estetica dei documenti.

Nel 1999 il **W3C** (World Wide Web Consortium) ha pubblicato il documento "**Web Content Accessibility Guidelines 1.0 - W3C Recommendation 1999**" contenente le linee guida della Web Accessibility Initiative (**WAI**) che dettano gli standard per l'accessibilità delle pagine web e che oggi, a livello mondiale, rappresentano il principale punto di riferimento per gli sviluppatori del web in tema di accessibilità.

La versione 2.0 della raccomandazione è attualmente un working draft.

Nella realizzazione dei portali, il CINECA segue fin dalla fase di progettazione le indicazioni in materia di accessibilità fornite dal consorzio **W3C** (<http://www.w3c.org/>) e dalla normativa italiana.

Il CINECA focalizza l'attenzione sul tema dell'accessibilità già a partire dal 2001, anche grazie alla normativa che a partire da quell'anno viene emanata per assicurare l'accessibilità dei siti web e delle applicazioni della pubblica amministrazione.

In particolare la recente **legge n. 4/2004** "Disposizioni per favorire l'accesso dei soggetti disabili agli strumenti informatici", cosiddetta "legge Stanca" definisce gli obblighi per l'accessibilità nella realizzazione e modifica dei siti Internet per le pubbliche amministrazioni.

### **Approfondimenti**

- "*Chi trae beneficio da siti e pagine web accessibili*" a cura di Michele Diodati  
[http://www.diodati.org/scritti/2004/guida/ele\\_acc07.asp](http://www.diodati.org/scritti/2004/guida/ele_acc07.asp)
- "*Lettori di schermo e browser testuali*" pubblicato sul sito Pubbliaccesso  
[http://www.pubbliaccesso.it/biblioteca/recensioni/prodotti/lettori\\_browser.htm](http://www.pubbliaccesso.it/biblioteca/recensioni/prodotti/lettori_browser.htm)

## Il rapporto tra Accessibilità e Usabilità

L'**usabilità** mira a fornire informazioni e servizi web nel modo più efficiente e chiaro possibile, unendo alla semplicità d'uso, la chiarezza comunicativa, nonché la velocità ed efficacia nella navigazione e consultazione dei contenuti.

La definizione di usabilità data nello standard ISO 9241-11 (Ergonomic requirements for office work with visual display terminals - Guidance on usability) è «il grado in cui un prodotto può essere usato da specifici utenti per raggiungere specifici obiettivi con **efficacia**, **efficienza** e **soddisfazione** in uno **specifico contesto d'uso**».

Progettare un sito secondo criteri di usabilità significa pertanto fare in modo che i suoi contenuti siano facilmente reperibili e fruibili dai suoi utenti. Un sito web può contenere tantissime risorse utili, ma se queste risorse non sono organizzate secondo le aspettative e gli effettivi bisogni dei destinatari, possono risultare addirittura introvabili e, di fatto, inutilizzate. Un progetto di usabilità evita questa ed altre spiacevoli evenienze e fa sì che **l'utente sia gratificato** dall'esperienza di navigazione, che ne tragga beneficio e che ripeta l'esperienza.

Usabilità e accessibilità sono strettamente legate insieme e l'una completa l'altra, infatti per soddisfare i requisiti, è necessario che la progettazione di un sito sia “centrata sull'utente”, caratteristica fondamentale per realizzare un sito usabile.

Anche il Regolamento di attuazione della legge “Stanca” 4/2004 sull'accessibilità, recita all'art. 2 (<http://www.pubbliaccesso.gov.it/normative/regolamento.htm>):

Sono accessibili i servizi realizzati tramite sistemi informatici che presentano i seguenti requisiti:

- a) **accessibilità** al contenuto del servizio da parte dell'utente;
- b) **fruibilità** delle informazioni offerte, caratterizzata anche da:
  - 1) **facilità e semplicità d'uso**, assicurando, fra l'altro, che le azioni da compiere per ottenere servizi e informazioni siano sempre uniformi tra loro;
  - 2) **efficienza nell'uso**, assicurando, fra l'altro, la separazione tra contenuto, presentazione e modalità di funzionamento delle interfacce, nonché la possibilità di rendere disponibile l'informazione attraverso differenti canali sensoriali;
  - 3) **efficacia nell'uso e rispondenza alle esigenze dell'utente**, assicurando, fra l'altro, che le azioni da compiere per ottenere in modo corretto servizi e informazioni siano indipendenti dal dispositivo utilizzato per l'accesso;
  - 4) **soddisfazione nell'uso**, assicurando, fra l'altro, l'accesso al servizio e all'informazione senza ingiustificati disagi o vincoli per l'utente;

### Approfondimenti

- Studio sulla “[Metodologia per la valutazione dell'accessibilità e dell'usabilità dei siti pubblici](#)” a cura del gruppo di lavoro "Metodologia" della Segreteria Tecnico-Scientifica della Commissione ICT disabili del CNIPA.
- [Webusabile.it](http://Webusabile.it)

## Normativa di riferimento

La **legge n. 4/2004** "Disposizioni per favorire l'accesso dei soggetti disabili agli strumenti informatici" ([http://www.pubbliaccesso.it/normative/legge\\_20040109\\_n4.htm](http://www.pubbliaccesso.it/normative/legge_20040109_n4.htm)), cosiddetta "legge Stanca", definisce gli obblighi per l'accessibilità nella realizzazione e modifica dei siti Internet per le pubbliche amministrazioni.

In merito alla stessa legge sono stati pubblicati:

- **DPR 1 marzo 2005, n. 75** - Regolamento di attuazione della Legge n.4/2004, per favorire l'accesso dei soggetti disabili agli strumenti informatici, pubblicato sulla Gazzetta Ufficiale n.101 del 3 maggio 2005 (<http://www.pubbliaccesso.it/normative/regolamento.htm>). Il DPR detta i criteri ed i principi operativi ed organizzativi generali per l'accessibilità, ovvero le modalità con cui può essere richiesta la valutazione, i criteri per l'eventuale partecipazione del richiedente ai costi dell'operazione, il logo con cui è reso manifesto il possesso del requisito dell'accessibilità, le modalità con cui può essere verificato il permanere del requisito stesso.
- **Decreto Ministeriale 8 luglio 2005**, pubblicato sulla Gazzetta Ufficiale n. 183 dell'8 agosto 2005 (<http://www.pubbliaccesso.it/normative/DM080705.htm>)  
Stabilisce:
  - a) le linee guida recanti i requisiti tecnici e i diversi livelli per l'accessibilità;
  - b) le metodologie tecniche per la verifica dell'accessibilità dei siti INTERNET, nonché i programmi di valutazione assistita utilizzabili a tale fine.

### Altre direttive e circolari:

#### **Direttiva del 30 maggio 2002 del Ministro per l'Innovazione e le Tecnologie**

[http://www.innovazione.gov.it/ita/documenti/direttiva\\_portale.shtml](http://www.innovazione.gov.it/ita/documenti/direttiva_portale.shtml)

#### **Circolare AIPA del 6 settembre 2001, n. AIPA/CR/32**

*"Criteri e strumenti per migliorare l'accessibilità dei siti web e delle applicazioni informatiche a persone disabili"*.

[http://www.cnipa.gov.it/site/contentfiles/00127800/127899\\_CR%2032\\_2001.pdf](http://www.cnipa.gov.it/site/contentfiles/00127800/127899_CR%2032_2001.pdf)

#### **Circolare del Dipartimento della Funzione Pubblica del 13 marzo 2001, n. 3/2001**

*"Linee guida per l'organizzazione, l'usabilità e l'accessibilità dei siti web delle pubbliche amministrazioni"*.

[http://www.pubbliaccesso.it/normative/circolare\\_funzione\\_pubblica\\_20010313.htm](http://www.pubbliaccesso.it/normative/circolare_funzione_pubblica_20010313.htm)

#### **Libro Bianco: Tecnologie per la disabilità**

A cura della Commissione interministeriale per l'uso e lo sviluppo delle tecnologie in favore delle categorie deboli del Ministro per l'Innovazione e le Tecnologie.

<http://www.innovazione.gov.it/librobianco/>

#### **Comunicazione della Commissione Europea del 28 settembre 2001**

*"eEurope 2002: accessibilità e contenuto dei siti Internet delle amministrazioni pubbliche"*.

<http://register.consilium.eu.int/pdf/it/01/st12/12265i1.pdf>

## Modalità di applicazione della legge 4/2004

### I soggetti interessati alla legge ([L.4/2004, Art. 3](#))

La legge 4/2004 si applica alle **pubbliche amministrazioni** di cui al comma 2 dell'articolo 1 del decreto legislativo 30 marzo 2001, n. 165, e successive modificazioni, agli enti pubblici economici, alle aziende private concessionarie di servizi pubblici, alle aziende municipalizzate regionali, agli enti di assistenza e di riabilitazione pubblici, alle aziende di trasporto e di telecomunicazione a prevalente partecipazione di capitale pubblico e alle aziende appaltatrici di servizi informatici.

Le disposizioni della presente legge in ordine agli obblighi per l'accessibilità non si applicano ai sistemi informatici destinati ad essere fruiti da gruppi di utenti dei quali, per disposizione di legge, non possono fare parte persone disabili. (es. Guardia di Finanza)

### A cosa si applica la legge

- i siti Internet
- le applicazioni basate su tecnologie Internet
- informazioni o servizi forniti mediante applicazioni Internet rese disponibili su reti Intranet o su supporti, come CD-ROM, DVD, utilizzabili anche in caso di personal computer non collegato alla rete
- materiale formativo e didattico utilizzato nelle scuole di ogni ordine e grado
- personal computer di tipo desktop e portatili
- ambiente operativo, applicazioni e prodotti a scaffale (applicazioni preconfezionate da utilizzarsi anche senza sviluppare appositi programmi di adattamento)

### Il primo livello di accessibilità

I siti Internet, le applicazioni web-based e tutto ciò che rientra nel campo di applicazione della legge, deve raggiungere il **primo livello di accessibilità**, accertato previo esito positivo della verifica tecnica che riscontra la conformità ai 22 requisiti tecnici elencati nell' Allegato A del Decreto Ministeriale 8 luglio 2005 (<http://www.pubbliaccesso.it/normative/DM080705-A.htm>).

### La verifica tecnica

La verifica tecnica è una verifica basata su parametri tecnici e oggettivi, stabiliti nel DM 8 luglio 2005 all' Allegato A (<http://www.pubbliaccesso.it/normative/DM080705-A.htm>).

La verifica tecnica fa ricorso a strumenti automatici, semiautomatici e alle conoscenze di una persona esperta di accessibilità.

La valutazione si conclude con un rapporto conclusivo dove si attesta la conformità con i 22 requisiti tecnici tramite una "checklist", nella quale si riporta punto per punto l'esito dei test effettuati.

## Il bollino dell'accessibilità

L'esito positivo della verifica tecnica consente di esporre sui siti e applicazioni Internet il logo attestante il possesso del requisito di primo livello di accessibilità, cosiddetto "bollino" dell'accessibilità. A tal fine, il responsabile dell'accessibilità informatica presso la pubblica amministrazione (una persona da individuare tra il personale appartenente alla qualifica dirigenziale), dovrà seguire le procedure indicate in una apposita sezione del sito del Centro nazionale per l'informatica nella pubblica amministrazione (CNIPA) per richiedere l'autorizzazione ad esporre il logo.

Tale responsabile dovrà preventivamente registrarsi nella sezione del sito del CNIPA per accedere alla quale riceverà, su richiesta, un nome utente e una password. La procedura prevederà quindi il caricamento di un modello, opportunamente compilato tramite form.

A conferma dell'avvenuta segnalazione, il CNIPA procederà al rilascio di un codice univoco che, inserito nel collegamento ipertestuale associato al logo utilizzato nel sito o applicazione interessati, permetterà anche al pubblico di consultare la segnalazione di conformità così come pervenuta al CNIPA e le informazioni relative.

## Il secondo livello di accessibilità

Il secondo livello di accessibilità riguarda la qualità delle informazioni fornite e dei servizi erogati dal sito Web e si articola in primo, secondo e terzo livello di qualità; tali livelli di qualità sono accertati con la verifica soggettiva attraverso i criteri di valutazione di cui all'Allegato B del DM 8 luglio 2005 (<http://www.pubbliaccesso.it/normative/DM080705-B.htm>), applicando la metodologia ivi indicata.

Il superamento del secondo livello di accessibilità **non è obbligatorio**; le attività di verifica soggettiva sono svolte da esperti in fattori umani, da esperti di interazione con le persone disabili e dal gruppo di valutazione, cioè un gruppo rappresentativo di utenti, del quale fanno parte anche disabili che utilizzano diverse tecnologie assistite.

## Approfondimenti

**Pubbliaccesso** - Sito sull'accessibilità a cura del CNIPA  
<http://www.pubbliaccesso.it>

## Un manuale: perché e per chi

Il manuale che proponiamo è il risultato di un approfondito lavoro di analisi e studio per rendere accessibili le pagine web progettate da CINECA e dai suoi partner, in particolare per siti e portali delle pubbliche amministrazioni.

Il CINECA si avvale della consulenza di ASPHI (Associazione per lo Sviluppo di Progetti Informatici per gli Handicappati). Con ASPHI è stato fatto un lavoro di verifica, selezione e approfondimento delle **linee guida del WAI** per giungere alla definizione delle regole e delle indicazioni di procedura proposte nel manuale "Linee guida per l'accessibilità di siti e portali web" versione 1.1 - maggio 2004.

In seguito all'emanazione della legge 4/2004, il manuale è stato aggiornato tenendo conto delle linee guida e metodologia indicate nell' **Allegato A del DM 8 luglio 2005** (<http://www.pubbliaccesso.it/normative/DM080705-A.htm>) recanti i requisiti tecnici per raggiungere il primo livello di accessibilità.

### A chi ci rivolgiamo

Il manuale è dedicato agli sviluppatori di pagine web e portali, realizzati con la tecnologia **BackStage Director** (BS-D) del CINECA, realizzati con altri strumenti di **Web Content Management** o di **Web Publishing**.

BS-D è uno strumento di Web Content Management, ovvero uno strumento in cui contenitore e contenuti sono realizzati in maniera indipendente per permettere l'implementazione e l'aggiornamento di questi ultimi anche a chi non ha specifiche competenze informatiche. Due le tipologie di operatori previste da BS-D: gli scenografi e i redattori. Gli Scenografi si occupano di realizzare la struttura, sono quindi web designer e grafici. Ai Redattori non sono richieste invece competenze informatiche (saper navigare in Internet e conoscere un qualsiasi programma di videoscrittura), possono essere diversi, con diverse competenze, e far capo a più Redazioni (ciascuna delle quali cura una o più sezioni del sito). I Redattori si occupano dei contenuti, della loro produzione, del loro inserimento e aggiornamento.

### L'idea e gli obiettivi

L'idea di redigere una **guida pratica per sviluppatori di pagine web** è nata dopo la pubblicazione del manuale "*Linee guida per l'accessibilità del sito web del Ministero della Giustizia*" prodotto durante lo svolgimento delle attività che hanno reso accessibile al livello "A" delle linee guida WAI, il sito del Ministero della Giustizia.

Dopo la versione 1.1 che ha prodotto un manuale generalizzato, rivolto a tutti gli sviluppatori di pagine web, si è giunti alla redazione di questo manuale versione 2.0, aggiornato secondo le linee guida della legge 4/2004.

Il manuale fornisce indicazioni concrete su come procedere di fronte ai singoli casi ed è un documento unico, a cui tutti gli sviluppatori possono fare riferimento. Oltre a risolvere questioni pratiche, un documento di tal genere garantisce uniformità a siti complessi e gestiti da più redazioni.

Nella parte tecnica sono indicati per ciascuna tipologia il corrispondente requisito tecnico della legge 4/2004, le regole generali a cui far riferimento, le istruzioni su come procedere concretamente, i riferimenti alle linee guida del WAI.

## Guida alla lettura

La parte di questo manuale dedicata alle caratteristiche tecniche è stata progettata in maniera da essere il più possibile chiara e di facile consultazione per redattori, scenografi BS-D, web designer, grafici e in genere sviluppatori di pagine web.

Si presenta divisa in capitoli - in alcuni casi ulteriormente organizzati in più parti - che contengono le direttive tecniche **classificate secondo i 22 requisiti tecnici della legge 4/2004 e per tipologia** (immagini, tabelle, navigazione...).

Ogni capitolo propone:

- Il **requisito tecnico** con il numero d'ordine, l'enunciato, il riferimento ai punti di controllo delle Web Content Accessibility Guidelines - versione 1.0 (WCAG 1.0) del W3C-WAI, nonché il riferimento agli standard definiti nella Sezione 508 del "Rehabilitation Act" (i punti di controllo del W3C-WAI e gli standard della Sezione 508 eventualmente richiamati nei singoli requisiti, sono da intendersi soltanto come elementi di riferimento, al fine di consentire un più facile riscontro con gli standard già impiegati e per facilitare l'utilizzo degli strumenti informatici di valutazione della accessibilità attualmente disponibili sul mercato).
- Una breve **descrizione della tipologia trattata**: cos'è, a cosa serve...
- **La regola generale**, ovvero la spiegazione di come rendere accessibile l'elemento.
- **Come procedere**, ovvero le indicazioni su come procedere in concreto per rendere accessibile l'elemento.
- Diversi **esempi**.
- Alcuni *source* dei file di esempio.
- Un riquadro **BackStage Director**, che illustra l'eventuale strumento di ausilio BS-D relativo all'elemento trattato nel capitolo.
- Le istruzioni operative per la **verifica del lavoro** relativi all'elemento trattato nel capitolo.
- Gli **approfondimenti**, ovvero riferimenti a risorse web relativi all'elemento trattato nel capitolo.

È importante sottolineare che le **indicazioni** della sezione "**Come procedere**" sono **prioritarie** rispetto a quelle indicate nei punti di controllo WAI. Poiché a volte esse non coincidono perfettamente, in collaborazione con **ASPFI**, è stato fatto un lavoro di verifica, selezione e approfondimento delle indicazioni WAI (in particolare di livello 1 e di livello 2) per giungere alla definizione delle regole e delle indicazioni di procedura che proponiamo in questo manuale.

Infine, è presente un **glossario minimo** per la terminologia tecnica.

## Linguaggio, DTD e intestazione del documento

(requisito n.1)

### Requisito n.1 -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:
  - 1) 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;
  - 2) evitare la generazione di nuove finestre; ove ciò non fosse possibile, avvisare esplicitamente l'utente del cambiamento del focus;
  - 3) 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:

(Priorità 2)

[3.1](#) Quando esiste un linguaggio di marcatori adatto, per veicolare informazione usare un marcatore piuttosto che le immagini.

[3.2](#) Creare documenti che facciano riferimento a grammatiche formali pubblicate.

[3.5](#) Usare elementi di intestazione per veicolare la struttura del documento e usarli in modo conforme alle specifiche.

[3.6](#) Marcare le liste ed elencare le voci della lista in modo appropriato.

[3.7](#) Marcare le citazioni. Non usare marcatura che definisca citazioni per ottenere effetti di formattazione come il rientro.

[11.1](#) Usare le tecnologie W3C quando sono disponibili e sono appropriate per un certo compito e usare le versioni più recenti quando sono supportate.

[11.2](#) Evitare le caratteristiche delle tecnologie W3C che sono disapprovate.

**Riferimenti Sec. 508:** *Non presente*

## DA HTML A XHTML

Come è noto, le pagine web sono realizzate sulla base del linguaggio **HTML** (HyperText Markup Language), cosiddetto linguaggio di marcatura, perché consente in modo molto semplice di "marcare" i contenuti, la struttura e la presentazione di una pagina web.

Le indicazioni vengono fornite attraverso degli appositi marcatori, detti **tag**, che hanno differenti nomi a seconda della loro funzione. I tag (detti anche **elementi**) possono essere dotati di proprietà associate (**attributi**), in grado di avere dei valori (ad esempio, <table border="1">).

Le definizioni e le regole di applicazione di tutti gli elementi e gli attributi HTML utilizzati in una pagina web sono scritte nella **DTD**, acronimo di Document Type Definition, cioè "definizione del tipo di documento". Ogni pagina HTML valida deve cominciare con la dichiarazione del tipo di documento: questa dichiarazione dice al browser quale è, e dove si trova la DTD.

Il W3C, World Wide Web Consortium, l'organizzazione che fra le varie attività si occupa di standardizzare e sviluppare la sintassi del linguaggio HTML, per la versione **HTML 4.01**, rilasciata il 24 dicembre 1999, ha prodotto 3 versioni di DTD: **Frameset**, **Transitional** e **Strict**.

Dopo la 4.01, il W3C non ha più rilasciato nuove versioni, ma ha compiuto un'opera di ridefinizione facendo evolvere l'HTML in **XHTML** (Extensible HyperText Markup Language), portando cioè l'HTML nella famiglia **XML** (Extensible Markup Language).

XHTML è il linguaggio attualmente raccomandato dal W3C per la realizzazione di pagine web. È compatibile con i browser e i software che supportano HTML 4.0 e contemporaneamente si integra con gli standard della famiglia XML.

Il **vocabolario** (i tag) XHTML è essenzialmente lo stesso di HTML 4.01, ma la sintassi è più rigorosa di HTML, perché basata sulle regole di XML. Ad esempio, XHTML richiede che a tutti i tag iniziali corrisponda un tag finale e che i valori degli attributi siano specificati tra apici. XHTML si basa quindi su XML e ne condivide regole di base e potenzialità, rispondendo così a due esigenze fondamentali:

- portare HTML nella famiglia XML con i benefici che ciò comporta in termini di estensibilità e rigore sintattico: il documento XHTML è un documento XML;
- mantenere la compatibilità con i browser che supportano HTML 4.0.

### Quali sono i vantaggi di XHTML rispetto ad HTML?

L'utilizzo di XHTML comporta notevoli benefici, primo fra tutti **vantaggi di gestione**, infatti un codice pulito e ben strutturato significa pagine più leggere, più gestibili, più facili da mantenere, quindi una diminuzione del lavoro di produzione.

Molti siti sono scritti in un HTML non valido, scritto male, che viene **comunque visualizzato dal browser web**, anche se non rispetta le regole delle raccomandazioni W3C. La conseguenza è un codice poco pulito, pesante, meno gestibile, meno portabile. Grazie a un sano codice strutturato i documenti scritti in XHTML e validati sono naturalmente **più accessibili**: la validazione aiuta infatti lo sviluppatore ad usare elementi e attributi in favore dell'accessibilità.

Gli **screen reader** e i browser alternativi, come quelli **vocali** o **testuali** gestiscono meglio una pagina con un codice pulito e ben strutturato.

I documenti scritti in XHTML sono più facilmente "**portabili**", cioè hanno la capacità di essere visualizzati e implementati efficacemente su diversi sistemi: PC, PDA, cellulari WAP/GPRS, WebTV. Potendo contare su un linguaggio essenziale, centrato sulla struttura del documento (titolo, intestazione, paragrafo, lista per gli argomenti...), si può pensare di produrre un unico flusso di dati in formato XHTML e un foglio di stile (si veda il capitolo "**Fogli di stile**" per approfondimenti)

adatto alle caratteristiche del dispositivo, per esempio con un layout “ridotto” se si tratta di un cellulare.

Sulla portabilità poggia l'enfasi con cui aziende del calibro di Nokia guardano a XHTML. È interessante il documento pubblicato sul sito <http://www.nokia.com/nokia/0,,32912,00.html> dal titolo "Next Generation Mobile Browsing" dove viene illustrata la tecnologia su cui si basano i servizi mobili della Nokia, fondati sull'integrazione tra XHTML e fogli di stile, con il supporto delle necessarie tecnologie sul lato server.

Infine, ma non meno importante, dal momento che XHTML è un documento XML, diventa **estensibile**. Ciò significa che sarà facile incorporare in un documento parti scritte in uno dei tanti linguaggi della famiglia XML.

Le **versioni di XHTML** pubblicate come raccomandazioni del W3C sono tre:

- **XHTML 1.0** <http://www.w3.org/TR/xhtml1/> del 26 gennaio 2000  
Si basa sulle tre DTD già definite per HTML 4.0: Strict, Transitional, Frameset
- **XHTML 1.1** <http://www.w3.org/TR/xhtml-modularization/> del 10 aprile 2001  
Basata sulla **DTD Strict** della versione 1.0. Rappresenta la prima formulazione del **concetto di modularità**. Gli elementi sono raggruppati in una serie di moduli indipendenti, che possono essere **implementati** o **esclusi** secondo le necessità.
- **XHTML Basic** <http://www.w3.org/TR/xhtml-basic/> del 19 dicembre 2000, versione “ridotta” del linguaggio. Specificatamente pensata per **dispositivi mobili** (PDA, cellulari), contiene solo gli elementi che si adattano a questi dispositivi (ad esempio, esclude i frame).

Le **DTD XHTML 1.0** sono tre: **Strict, Transitional, Frameset**.

- **XHTML 1.0 Strict**  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
È centrata esclusivamente sulla **struttura** del documento. Elimina diversi elementi ed esclude tutti gli attributi che definiscono **la presentazione**. Per questo scopo vanno usati i CSS.
- **XHTML 1.0 Transitional**  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
Basata sull'omologa DTD di HTML 4.0 . Supporta tutti gli elementi e gli attributi di presentazione di HTML 4.0.
- **XHTML 1.0 Frameset**  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">  
Va usata quando si utilizzano i frame.

## Descrizione delle tre DTD

Per siti di nuova costruzione è si deve utilizzare **XHTML 1.0 Strict**, infatti la DTD Strict è il **miglior punto di partenza** per il raggiungimento di un'elevata accessibilità. È l'unica che può assicurare la maggiore indipendenza dei contenuti della pagina dai suoi aspetti di presentazione, e quindi la possibilità di accedere debitamente ai contenuti per il maggior numero possibile di programmi utente.

Con XHTML 1.0 Strict si ottiene una **totale separazione fra struttura e contenuto** da un lato e presentazione dall'altro. Se si inseriscono elementi non supportati (ad esempio font, larghezza per le celle di tabelle, align... ) il documento non è valido. La validazione aiuta lo sviluppatore ad usare elementi e attributi in favore dell'accessibilità (ad esempio l'attributo **alt** per le immagini è obbligatorio).

XHTML dovrebbe essere utilizzato esclusivamente per i contenuti e la struttura, eliminando gli elementi e gli attributi di presentazione, mentre il foglio di stile dovrebbe gestire la presentazione.

Per **elemento di presentazione** si intende un elemento che specifica la presentazione del documento (ad esempio <b>, <font>, <center>). Analogamente, un attributo che specifica un modo di presentare un certo contenuto è un attributo di presentazione (ad esempio align, height, bgcolor, size, face, color e simili).

Anche una pagina conforme ad HTML 4.01 Strict può essere altrettanto accessibile della stessa pagina conforme a XHTML 1.0 Strict, però il nostro consiglio è quello di utilizzare la DTD Strict di XHTML, in quanto, come già scritto precedentemente, il linguaggio XHTML aggiunge all'HTML benefici di gestione, di portabilità e di estensibilità propri del linguaggio XML.

Produrre una pagina web in XHTML 1.0 Strict significa fare un investimento per il futuro: consente infatti di entrare nel mondo XML con tutti i benefici che ci si aspetta, assicurandosi la compatibilità con gli **user agent** (browser grafici, browser testuali, browser vocali, screen reader, cellulari...) attuali e futuri e di non perdere la compatibilità con i vecchi browser.

La DTD **Transitional**, invece, va utilizzata quando desideriamo comporre pagine compatibili con i browser grafici più datati, forniti di scarso o nullo supporto per i fogli di stile. In sostanza, dobbiamo ricorrere alla DTD transitoria quando vogliamo che le nostre pagine web siano viste più o meno allo stesso modo su Internet Explorer dalla versione 3 alla 6 e su Netscape dalla versione 3 alla 7. Ciò però significa mescolare ai contenuti della pagina, in modo inestricabile, una quantità di codice avente il solo scopo di definire la presentazione visuale dei contenuti.

Mentre la DTD **Frameset** andrebbe usata con parsimonia, non perché non sia possibile, ricorrendo a certi accorgimenti, realizzare dei siti a frame in certa misura accessibili, ma perché è possibile ottenere un'accessibilità senz'altro maggiore non utilizzando del tutto i frame.

## Approfondimenti

- *Diodati.org*: Siti ad elevata accessibilità <http://www.diodati.org/scritti/2004/guida/index.asp>
- *W3C site*: XHTML - <http://www.w3.org/xhtml/>
- *Html.it*: Guida XHTML in italiano - <http://www.html.it/xhtml/>
- *Zvon.org*: XHTML 1.0 reference - <http://www.zvon.org/xxl/xhtmlReference/Output/index.html>
- *W3Schools Online Web Tutorials* - <http://www.w3schools.com/xhtml/default.asp>
- *XHTML.ORG*: Risorse selezionate su XHTML - <http://www.xhtml.org>

## Gli standard del web

Una chiave per sviluppare pagine accessibili consiste nel rispettare gli **standard web** nella rappresentazione del contenuto e nella formattazione.

Per progettare nel rispetto degli standard web è necessario definire struttura, contenuto, formattazione e funzionamento utilizzando **un codice valido** (HTML, XHTML, CSS, DOM, ECMAScript – JavaScript, ecc.), ottenendo in tal modo garanzie di accessibilità e di compatibilità con i programmi utente (*user agent*) attuali e futuri e con le tecnologie assistive.

### Lo strumento BS-D

Per effettuare un riscontro della rispondenza alla sua definizione formale del linguaggio a marcatori utilizzato, BackStage-Director mette a disposizione uno strumento trasparente al redattore in fase di salvataggio dell'oggetto.

Nel caso sia rilevato un errore, lo strumento di verifica di BS-D propone la correzione del codice: il redattore può decidere se accettare la correzione (in quel caso è sicuro di ottenere codice valido) oppure può salvare comunque l'oggetto senza correggerlo (in quel caso sicuramente il codice NON sarà valido).

Lo strumento BS-D di verifica del codice si basa sui seguenti strumenti di verifica automatici, reperibili liberamente su Internet:

- Markup Validation Service (<http://validator.w3.org/>)
- HTML Tidy (<http://infohound.net/tidy/>)

## Intestazione di documento: l'elemento <!DOCTYPE>

Seguendo le indicazioni del Consorzio W3C, l'elemento <!DOCTYPE> deve essere il primo elemento ad aprire il documento (X)HTML.

L'elemento <!DOCTYPE> ha il compito di fornire informazioni al server web e al browser che ospita la pagina sul tipo di documento visualizzato, in particolare definisce quale versione di (X)HTML si sta utilizzando e la relativa DTD a cui è conforme.

Il <!DOCTYPE> consente ai browser e ad altri tools di processare correttamente il documento, ad esempio i sistemi automatici di validazione come il [Markup Validator](#) del W3C consentono di individuare eventuali errori di sintassi (X)HTML e validarli.

L'assenza o la presenza ed il tipo di <!DOCTYPE> determinano il modo in cui è visualizzata una pagina e i problemi che possono sorgere in fase di realizzazione di una pagina possono dipendere da questo piccolo pezzo di codice che precede gli elementi (X)HTML.

### Come procedere

- Includere sempre all'inizio di un documento una dichiarazione sul tipo di documento che rimandi a una DTD pubblicata.
- La lista completa dei <!DOCTYPE> che si possono utilizzare è curata dal W3C ed è disponibile all'indirizzo <http://www.w3.org/QA/2002/04/valid-dtd-list.html>.
- Validare la sintassi (X)HTML tramite il sistema automatico di validazione raccomandato dal W3C (<http://validator.w3.org/>).

**Esempio**

```
<!DOCTYPE> completo per l'XHTML Strict versione 1.0:

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

**Approfondimenti**

- *A List Apart: Fixing Your Site With The Right DOCTYPE* - <http://www.alistapart.com/articles/doctype/>

**Intestazione di documento: i META tag**

I **META tag** sono stringhe di codice che figurano in testa al documento e che vengono lette per prime dai motori di ricerca.

I META tag forniscono meta-informazione alla pagina web, come ad esempio le **keywords** (parole chiave) che descrivono sinteticamente il contenuto di una pagina web: l'indicazione dell'autore della pagina, il titolo che apparirà alla fine della ricerca...

**Come procedere**

- Fornire i META tag per aggiungere informazione di tipo semantico alla pagina web.

**Approfondimenti**

- *Diodati.org: Specifiche HTML 4.01 - I metadati* - <http://www.diodati.org/w3c/html401/struct/global.html#h-7.4.4>

**Intestazione di documento: il titolo della pagina**

L'elemento **<title>** fornisce il titolo alla pagina, che solitamente viene visualizzato dai browser nell'intestazione di pagina e viene letto dallo screen reader prima di procedere alla lettura del contenuto della pagina.

**Come procedere**

- Assicurare un titolo significativo e conciso per ogni pagina tramite l'elemento **<title>**.

## Contenuto, Struttura, Presentazione

Capire la differenza tra contenuto, struttura e presentazione di una pagina web è fondamentale per costruire siti accessibili.

### Contenuto

Per contenuto si intendono testo, immagini, suoni, filmati e animazioni.

### Struttura

La struttura rappresenta l'organizzazione logica del contenuto (ad esempio capitoli, titoli, sottotitoli).

### Presentazione

La presentazione implica la formattazione del carattere, la posizione degli oggetti e le scelte di layout.

Quando si realizza un sito web bisogna dividere la presentazione dalla struttura logica: per dare un significato logico a titoli e sottotitoli di una pagina non basta evidenziarli con un carattere più grande, ma occorre impostarli con i **marcatori** (X)HTML organizzando la struttura del documento in base agli Headers `<h1> <h2>...<h6>`.

Scegliere una struttura chiara e coerente per le informazioni sulle pagine web, utilizzando i codici dedicati disponibili (elemento marcatore strutturale) e separando i codici di contenuto e stile, in modo che le tecnologie assistive possano navigare efficientemente attraverso contenuto e struttura.

### Regola generale

Utilizzare titoli, liste e una struttura coerente, facile da comprendere e riconoscere. Per l'impaginazione fare ricorso ai fogli di stile.

### Come procedere

- Organizzare i documenti molto lunghi in modo che possano essere facilmente navigabili con o senza le tecnologie assistive.
- Assicurarsi che i documenti siano chiari e semplici.

### Approfondimenti

- *Diodati.org*: Valorizzare la struttura logica del documento [http://www.diodati.org/scritti/2004/guida/ele\\_acc21.asp](http://www.diodati.org/scritti/2004/guida/ele_acc21.asp)
- *Diodati.org* - "Un esempio da non seguire: formattare utilizzando esclusivamente elementi e attributi di presentazione dell'HTML" [http://www.diodati.org/scritti/2004/guida/ele\\_acc10.asp](http://www.diodati.org/scritti/2004/guida/ele_acc10.asp)

## Liste

Le **liste** rappresentano uno dei modi più diffusi per organizzare informazioni all'interno di siti web. Una delle loro caratteristiche principali è sicuramente quella di fornire un quadro chiaro e sintetico dell'argomento trattato.

Le liste possono essere ordinate, non ordinate, o di definizioni

È preferibile gestire gli stili delle liste attraverso i fogli di stile: questi consentono un miglior controllo sulla formattazione delle liste (ad esempio, per quanto riguarda il tipo di numerazione, l'uso di immagini per marcare le singole voci, il rientro e l'allineamento, ecc.).



### Regola generale

Marcare le liste utilizzando gli elementi `<ol>` (per le liste ordinate), `<ul>` (per le liste non ordinate) o `<dl>` (per le liste di definizioni); elencare in modo appropriato le voci delle liste ordinate e non ordinate per mezzo dell'elemento `<li>` e le voci delle liste di definizione attraverso gli elementi `<dt>` e `<dd>`, dove `<dt>` indica il termine della lista e `<dd>` la sua definizione.

### Come procedere

- Non utilizzare simboli come “-” (meno) o “\_” (underscore) per elencare le voci di una lista.
- Se si utilizzano le immagini per elencare le voci di una lista, prevedere l'attributo **alt** con la dicitura "punto elenco".

### Lo strumento BS-D

Dalla "Barra di formattazione", selezionare il simbolo  per fare una lista non ordinata e il simbolo  per fare una lista ordinata.



### Approfondimenti

- *Tobias Schmidt*: Styling lists with CSS - <http://www.royal-ts.de/mtarchives/000805.php>.
- *W3Schools*: CSS List Properties - [http://www.w3schools.com/css/css\\_list.asp](http://www.w3schools.com/css/css_list.asp).
- *Eric B. Bednarz*: Manipulating Margin and Padding of Lists With CSS - <http://devnull.tagsoup.com/list/>.

## I pop-up

### Problemi e punti deboli dei pop-up

- Cambiare la finestra attiva oppure aprire nuove finestre può disorientare l'utente, dato che il focus viene trasferito ad una pagina diversa da quella originaria.
- Il cambio di focus senza il permesso dell'utente può essere considerato troppo invasivo e "infastidire" l'utente. È bene sapere che la maggior parte dei browser ha una funzionalità denominata "**pop-up killer**", che serve a controllare le finestre di pop-up, cioè è in grado di forzare il browser a non accettarli e impedirne di conseguenza la comparsa iperinvasiva sul desktop (pensiamo ad esempio alla pubblicità su Internet).
- Aprire una nuova finestra del browser, significa anche cominciare una nuova navigazione e di conseguenza il bottone "Indietro" del browser inizialmente non è attivo: se l'utente usa un monitor molto piccolo, potrebbe non accorgersi dell'apertura di una nuova finestra sopra quella di default e provare a selezionare "Indietro" per tornare sulla home page, con conseguente confusione.
- Gli utenti che utilizzano screen reader o browser vocali possono avere difficoltà nel riconoscere l'apertura di una nuova finestra: [Jaws](#) annuncia "**nuova finestra del browser**" quando un link si apre in una nuova finestra, ma questo sfugge facilmente essendo incastrato tra la lettura del testo del link e la nuova pagina. [Home Page Reader](#) ha invece una soluzione migliore: suona in modo diverso ogni volta che apre una nuova finestra. E [Window Eyes](#), un altro popolare screen reader, non fornisce alcuna indicazione.
- E, comunque, il bottone "Indietro" non è più funzionante. Se all'utente sfugge l'annuncio del "nuova finestra del browser", non può dare un'occhiata alla barra delle applicazioni attive (**taskbar**) e verificare che ha due finestre del browser aperte. Dovrà leggere l'intera lista delle finestre aperte grazie, ad esempio, alla scorciatoia di tastiera specifica di **Jaws Insert+F10** o con la normale combinazione **Alt+Tab**.
- Se Javascript è disabilitato o se il browser non lo supporta, il pop-up gestito con Javascript non viene visualizzato.
- I motori di ricerca ignorano i link alle finestre di pop-up (gli elementi degli script infatti vengono sempre ignorati).

### Come procedere

*Quando il Pop-up è attivato in modo automatico all'apertura del sito*

- Utilizzare l'elemento `<noscript>` per fornire esattamente lo stesso tipo di contenuto e di funzionalità anche per gli utenti non provvisti di interprete javascript. Ad esempio:

```
<script type="text/javascript"> function openwindow()
{.....}
</script>
<noscript> Versione alternativa per utenti che hanno JavaScript
disabilitato <a href="news.htm" target="_blank">News</a> (il link si apre
in una nuova pagina del browser)</noscript>
```

*Quando il Pop-up è attivato in modo esplicito dall'utente*

- Informare l'utente che la selezione di un collegamento attiva l'apertura di una nuova finestra: si consiglia di specificarlo nell'attributo **title** con una frase del tipo "la pagina si apre in una nuova finestra".

- Non utilizzare codice Javascript che richieda l'uso del mouse, ovvero si deve essere in grado di aprire un pop-up sia con il mouse che con la tastiera. La soluzione tecnica è la seguente: tra i gestori di eventi previsti nelle specifiche, troviamo anche **onkeypress**, evento che viene attivato con la pressione di un tasto nel momento in cui un determinato elemento riceve il focus. Nel caso dello script del pop-up basterà aggiungere al codice una nuova istruzione:

```
<a href="popup_esempio.html" onclick="MM_openBrWindow('popup_esempio.html',
'nuovaFinestra','width=400,height=400'); return false;"
onkeypress="MM_openBrWindow('popup_esempio.html',
'nuovaFinestra','width=400,height=400'); return false;" title="Esempio di
pop-up accessibile - Link interno - Nuova finestra">Apri il pop-up</a>.
```

### *Caratteristiche che deve avere la finestra di pop-up*

- Non deve avere nessun attributo (X)HTML che impedisca all'utente di ridimensionare la finestra; inoltre deve avere le barre di scorrimento. All'atto dell'apertura di una finestra pop-up con Javascript, lo sviluppatore ha l'opportunità di non far visualizzare la barra degli strumenti, le barre di scorrimento, la barra di stato e così via. Ma ciascuno di questi elementi facilita l'accessibilità per specifiche tipologie di disabili. Quindi si dovrebbe essere assai cauti anche nell'eliminare gli strumenti standard previsti dal browser. Ed ancora, è anche possibile per lo sviluppatore impedire il ridimensionamento della finestra, il che può rappresentare un problema per gli utenti che hanno bisogno di ingrandire lo schermo; essi saranno obbligati a visualizzare il contenuto ingrandito attraverso una piccola finestra, piuttosto che a schermo pieno.
- Per aiutare l'utente che utilizza uno screen reader a orientarsi nella navigazione e quindi riconoscere l'apertura di una nuova finestra, si consiglia di esplicitare con un messaggio scritto nella pagina, che si tratta di una nuova finestra del browser. Il procedimento può essere simile a quello già utilizzato per lo "Skipping Navigation", ovvero l'utilizzo di una funzione del CSS che permette di nascondere testo ai browser grafici, ma non a quelli vocali. Ad esempio:

In testa, nel sorgente (X)HTML del pop-up, inserire il seguente codice (il messaggio è indicativo):

```
<div class="nascosto">
Ti trovi in una nuova finestra del browser. In questa pagina sono
elencate tutte le news ecc.. ecc... Per ritornare alla home page
e proseguire nella navigazione, chiudi questa finestra del
browser.</div>
```

Nel CSS, il div "nascosto" è così impostato:

```
div.nascosto { display: none; }
```

Nulla vieta di rendere esplicito a tutti con un messaggio "in chiaro", l'apertura di una nuova finestra, senza utilizzare un procedimento solo per gli utenti di *screen reader*.

- La navigazione all'interno del pop-up non dovrebbe rimandare alla home page del sito, né rendere disponibile la navigazione dei menù. L'ideale sarebbe riuscire a mantenere "chiusa" la navigazione all'interno dell'argomento trattato nel pop-up.
- Infine, il contenuto del pop-up deve essere accessibile.

### **Approfondimenti**

- JavaScript and Dynamic HTML Accessibility  
<http://www.trace.wisc.edu/world/java/jscript.htm>

## Frame (requisito n.2)

**Requisito n.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:

- a) 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;
- b) 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; +
- c) 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:

(Priorità 1)

[12.1](#) Dare un titolo a ogni frame per facilitare l'identificazione del frame e la navigazione.

(Priorità 2)

[12.2](#) Descrivere lo scopo dei frame e il modo in cui essi interagiscono se non è evidente dai titoli dei frame da soli..

**Riferimenti Sec. 508:** 1194.22 (i)

### Descrizione

Una configurazione di pagina può essere suddivisa in **cornici** (frame), il contenuto delle quali può essere aggiornato separatamente mediante l'interazione dell'utente. Nella codificazione di layout della pagina, aggiungere un nome significativo per ciascun frame. In tal modo le tecnologie assistive possono fornire informazioni che permettono all'utente di capire il rapporto fra i frame e il loro contenuto. La presenza di frame rende più difficile la navigazione in avanti e indietro nella cronologia.

Chi utilizza un browser testuale o uno screen reader, visitando una pagina che si basa su frame, deve scegliere quale aprire da una lista dei frame disponibili. Se gli sviluppatori di pagine web forniscono titoli significativi ai frame, allora l'utente può facilmente identificare quello che vuole aprire.

Se da un lato l'utilizzo dei frame risulta comodo per alcune funzionalità di navigazione (come mantenere un menu nel frame superiore sempre visibile), dall'altro lato possono causare problemi seri di usabilità. Ad esempio, alcuni browser non riescono a stampare la pagina basata su frame in modo appropriato, il salvataggio dell'indirizzo di una pagina come bookmark può risultare difficoltosa, spesso capita che i motori di ricerca indicizzino pagine interne dei frame che non consentono di ritornare alla gestione dei menu, la navigazione del sito risulta in generale più difficoltosa, specialmente per gli utenti che fanno uso di tecnologie assistive.

### Regola generale

- Quando possibile, **non** utilizzare i frame.
- Se si utilizzano i frame, assegnare un titolo a ciascun frame, utilizzando l'elemento `<title>`, per facilitare l'identificazione dello stesso.

- Utilizzare l'elemento **<noframes>** per specificare un contenuto alternativo per i browser che non supportano i frame o che sono configurati per non mostrarli. Questi browser riprodurranno solo il contenuto dell'elemento **<noframes>**.

### Come procedere

- Per facilitare l'identificazione e la navigazione nei frame è necessario dare un titolo significativo a ogni frame, tramite l'elemento **<title>**. Inoltre è necessario assegnare un nome significativo a ciascun elemento **<frame>** incluso nel documento FRAMESET, tramite gli attributi **name** e **title**.

### Esempio:

#### Codice HTML del documento frameset.html

```
<html>
<head>
<title>Un esempio di pagina basata su frame</title></head>
<frameset cols="20%, 80%">
  <frameset rows="100, 200">
    <frame src="logo.html" title="Logo dell'azienda" name="Logo azienda">
    <frame src="nav.html" title="Menù di navigazione" name="menù navigazione">
  </frameset>
  <frame src="contenuti.html" title="Contenuti principali dell'azienda"
name="contenuti principali">
</frameset>
```

#### Codice HTML del frame contenuti.html

```
<html>
<head>
<title> Contenuti principali dell'azienda </title>
...
```

- Per rendere accessibile il contenuto di un frame, il sorgente dovrebbe essere sempre un file HTML. Se il sorgente di un frame è un'immagine (ad esempio logo.gif), non c'è la possibilità di specificare l'attributo **alt** e quindi non risulta accessibile.

Nell'esempio seguente, il contenuto del frame logo.html è accessibile, perché si tratta di un documento HTML che contiene un elemento **<img>** con l'attributo **alt**.

### Esempio:

#### Codice HTML del frame logo.html

```
<html>
<head>
...
<img src= "logo.gif" alt= "nome dell'azienda">
</html>
```

- Utilizzare sempre l'elemento **<noframes>**, per fornire un contenuto equivalente agli user agents che non supportano i frame o che sono configurati per non visualizzarli. Nell'esempio seguente, l'elemento **<noframes>** fornisce un testo descrittivo con i puntatori alle pagine di riferimento dei frame. L'elemento **<noframes>**, quindi, non dovrebbe essere usato solo per avvisare l'utente di installare un browser compatibile con i frame, ma per puntare alle pagine equivalenti dei frame.

**Codice HTML del documento frameset.html**

```

<html><head><title>Un esempio di pagina basata su frame</title></head>
<frameset cols="20%, 80%">
  <frameset rows="100, 200">
    <frame src="logo.html" title="Logo dell'azienda" name="Logo azienda">
    <frame src="nav.html" title="Menù di navigazione" name="menù di
navigazione">
  </frameset>
  <frame src="contenuti.html" title="Contenuti principali dell'azienda"
name="contenuti principali">
</frameset>
<noframes>
<p>Questo documento contiene tre frame:
<ul>
<li>Il primo frame è per il logo e nome dell'azienda <a href="logo.html">Logo
Azienda </A></li>
<li>Il secondo frame è usato per il menù di navigazione <a
href="navigation.html">Menù di navigazione</A></li>
<li>Il terzo frame è usato per i contenuti principali, ovvero dove appare il
contenuto del link scelto dal menù di navigazione <a
href="maincontent.html">Contenuti principali</a></li>
</ul></p></noframes></frameset>

```

**Inline Frames**

È un tipo di frame integrato da solo, ovvero **senza frameset** in una pagina, allo scopo di richiamarne un'altra all'interno di questa. Permette quindi di creare un frame in un documento che non utilizza la struttura a frame.

Tale elemento non deve necessariamente essere inserito al bordo di una pagina, come avviene per i frame tradizionali, ma in qualsiasi punto di un documento. È possibile creare vere e proprie finestre all'interno della pagina, con contenuti indipendenti dal resto del documento. Mentre i frame tradizionali vengono usati per dividere l'intera finestra del browser in sottofinestre, l'elemento **<iframe>** appare interno alla presentazione del documento come parte integrante dello stesso.

L'elemento **<iframe>** non è supportato da Netscape 4.x, ma da tutti gli altri browser a partire da Internet Explorer 4, Opera, Mozilla/Netscape 6 e superiore.

Per i browser che non sono in grado di leggere l'elemento **<iframe>**, è possibile specificare, tra l'apertura e la chiusura del tag, un contenuto alternativo. In realtà questi browser sono "ciechi" a questo tag e quindi leggono direttamente il contenuto al suo interno; sono invece i browser che supportano questa sintassi a ignorare volutamente ciò che viene compreso tra apertura e chiusura del tag.

L'informazione da inserire in linea è indicata dall'attributo **src** dell'elemento **<iframe>**. Il contenuto dell'elemento **<iframe>** dovrebbe essere visualizzato solo dagli interpreti che non supportano frame o sono configurati per non visualizzarli. Ogni elemento **<iframe>** può avere gli attributi **width**, la larghezza e **height**, l'altezza del frame. Inoltre l'autore può inserire altre informazioni relative allo scroller (se automatico), al bordo, il nome dell'**<iframe>** (opzionale).

L'elemento **<iframe>** è, tecnicamente, un elemento "inline"; il suo contenuto può essere testo, può contenere elementi inline o anche elementi blocco; ciò significa che un elemento **<iframe>** può essere contenuto da un elemento blocco e può contenere un elemento blocco; ad esempio è corretto avere:

```
<div><iframe src="foobar"...><p>testo testo testo</p></iframe></div>
```

L'elemento **<iframe>** non è valido per XHTML 1.0 Strict, ma è invece incluso nella lista degli elementi validi di XHTML 1.0 Transitional e Frameset. Gli elementi **<iframe>** non possono essere ridimensionati dall'utente. Essi infatti non accettano l'attributo **noresize** (attributo che, nei frame tradizionali, impedisce il ridimensionamento a video da parte dell'utente), ma si adattano facilmente al contenuto della pagina.

### Come procedere

- Quando si utilizza un "inline frame" con l'elemento **<iframe>**, fornire un titolo significativo tramite l'attributo **title**.

### Esempio 1

Il codice dell'esempio seguente mostra un inline frame circondato da un bordo, nel mezzo del testo:

```
<div>   <iframe src="docuemnto.html" width="400" height="500"
scrolling="auto" frameborder="1">
      [Il vostro interprete non supporta i frame o è attualmente configurato
per non visualizzarli. Tuttavia, potete visitare <a
href="documento.html">il relativo documento.</a>]
</iframe> </div>
```

### Esempio 2

Non tutti i browser supportano tale metodo. L'uso dell'elemento **iframe** potrebbe essere la simulazione di una "tabella scrollabile": la struttura a **iframe** permette di avere come riferimento fisso l'intestazione della tabella e la parte relativa ai dati, invece, scrollabile:

La tabella di seguito riporta i dati relativi agli iscritti della facoltà di Fisica, Università di Bologna, negli anni 1995-2004;

Anno	Maschi	femmine	Totale
1995	1600	240	1840
1996	1685	140	1825

Si hanno due elementi **<iframe>**, uno per l'intestazione della tabella, l'altro per il corpo della tabella:

```
<iframe src="intestazione_tabella.htm" width="100%" height="30"
style="height:1.8em" scrolling="no" marginwidth="0"
marginheight="0"></iframe><br><iframe src="corpo_tabella.htm" width="100%"
height="60" style="height:4em" marginwidth="0" marginheight="0"> </iframe>
```

Il primo **<iframe>** contiene tutte le celle della intestazione, che sono fisse, ciò che varia è il corpo della tabella (secondo **<iframe>**): l'utente ha la possibilità di "scrollare" con la barra di scorrimento verticale tutte le celle della tabella. Tale metodo è utile quando si hanno tabelle caratterizzate da numerose righe.

### Approfondimenti

- *Dorward's Personal Website*: <http://dorward.me.uk/www/frames/>
- *Jakob Nielsen*: <http://www.useit.com/alertbox/9612.html>
- Using inline frames - <http://www.cs.tut.fi/~jkorpela/html/iframe.html>

## Immagini e animazioni (requisito n.3)

**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:

(Priorità 1)

[1.1](#) Fornire un equivalente testuale per ogni elemento non di testo (per esempio, mediante "alt", "longdesc" o contenuto nell'elemento stesso). *Questo comprende:* immagini, rappresentazioni grafiche di testo (compresi i simboli), zone di immagini sensibili, animazioni (ad es. GIF animate), applet e oggetti programmati, arte ASCII, frame, script, immagini usate come richiami per elenchi, spaziatori, bottoni grafici, suoni (azionati con o senza l'intervento dell'utente), file di solo audio, tracce audio di video e video.

[6.2](#) Assicurarsi che gli equivalenti del contenuto dinamico vengano aggiornati quando il contenuto dinamico cambia.

**Riferimenti Sec. 508:** 1194.22 (a)

Per ciascuna immagine o animazione su una pagina web, **fornire un testo alternativo** con l'attributo **alt** del marcatore **IMG**, ad esempio:

```
<IMG SRC="immagine.gif" border="0" alt="testo alternativo">
```

Chi naviga con un browser testuale o con un sintetizzatore vocale non può vedere le immagini: la sua comprensione di una pagina web contenente immagini sarà perciò tanto maggiore quanto più sono validi i testi alternativi eventualmente presenti.


Per scrivere dei buoni testi alternativi, bisogna **riflettere sulla funzione che un'immagine ha nella pagina**: il compito del testo alternativo è svolgere la stessa funzione dell'immagine per chi non può vederla, non fornire un'acritica descrizione di ciò che l'immagine mostra.

### Come procedere

- Fornire testo alternativo a tutte le immagini con l'attributo **alt** degli elementi **<img>** o **<input>**, o attraverso l'attributo **title**.
- Il testo alternativo all'interno dell'attributo **alt** deve essere conciso e significativo, per illustrare in modo preciso la funzione dell'immagine all'interno della pagina.
- Se l'elemento grafico fornisce un collegamento (**link**), fornire un testo alternativo all'interno dell'attributo **alt** per descrivere l'immagine e usare l'attributo **title** per descrivere più chiaramente e accuratamente l'obiettivo del link ipertestuale (evitare espressioni generiche come "**premi qui**" o "**torna indietro**").
- Se l'elemento grafico è puramente decorativo (pallino, spazio bianco, barra, ecc...) usare **alt=""** lasciando una stringa vuota senza spazi fra gli apici.
- Si raccomanda di evitare l'uso di immagini per formattare le pagine, ad esempio immagini utilizzate per definire degli spazi. Fare pertanto ricorso ai fogli di stile.
- Quando l'elemento grafico è una foto, cercare di descriverla in modo sintetico.
- Quando si usano caratteri di testo e/o simboli combinati insieme per creare un'immagine (cosiddetta "**arte ASCII**"), sostituirla con un'immagine fornita di testo alternativo.

**Nota:** Si raccomanda la lettura dell'articolo di Michele Diodati "[Testi alternativi](http://www.diodati.org/scritti/2002/g_alt/index.asp)" ([http://www.diodati.org/scritti/2002/g\\_alt/index.asp](http://www.diodati.org/scritti/2002/g_alt/index.asp)) che si basa sull'analisi svolta da **Alan J. Flavell** su immagini e testi alternativi: l'analisi sfocia nella definizione di quattro categorie generali, sotto cui possono essere ricondotte tutte le immagini presenti su una qualsiasi pagina web e per ognuna di queste categorie risulta appropriato un preciso tipo di testo alternativo.

### Lo strumento BS-D

Dalla "Barra di formattazione", selezionare 'Aggiungi METAFILE' per caricare un'immagine nel directorio e selezionare il simbolo  per salvare un'immagine nella pagina web



### Verifica del lavoro

Verificare che il contenuto informativo e le funzionalità della pagina siano ancora fruibili in caso di disattivazione del caricamento delle immagini.

- Effettuare una verifica manuale simulando la disattivazione delle immagini con uno dei seguenti browser:
  - Internet Explorer: Web Accessibility Toolbar -> images -> Toggle Image/Alt
  - Mozilla Firefox: Tools -> options -> web features -> proprietà web -> load images
  - Netscape: preferences -> privacy&security -> images -> do not load any images
  - Opera 6: file -> preferences -> Multimedia -> Images
  - Opera 7 e 8: tools -> preferences -> web pages -> Images

La Web Accessibility Toolbar è scaricabile all'indirizzo:

<http://www.nils.org.au/ais/web/resources/toolbar/index.html>

- Se necessario effettuare anche una verifica automatica con uno dei seguenti strumenti automatici:
  - Analizzatore di immagini di JuicyStudio ([www.juicystudio.com/services/image.asp](http://www.juicystudio.com/services/image.asp))
  - Servizio **Alt Text Quality** di "Cynthia Says Portal" (<http://www.cynthiasays.com/fulloptions.asp>)

### Approfondimenti

- *WebAIM*: How to Create Accessible Graphics (<http://www.webaim.org/techniques/images/>) .
- *Jukka Korpela*: Simple guidelines on using ALT texts in IMG elements (<http://www.cs.tut.fi/%7Ejkorpe/alt.html>)
- WAI "[Techniques for Web Content Accessibility Guidelines 1.0](http://www.w3.org/WAI/techniques/html1.0/)"
- *A. J. Flavell* (traduzione a cura di Diodati) : Uso dei test ALT all'interno di IMG ([http://www.diodati.org/scritti/2002/g\\_alt/index.asp](http://www.diodati.org/scritti/2002/g_alt/index.asp)) .
- *WebAIM*: How to Create Accessible Graphics (<http://www.webaim.org/techniques/images/>) .
- *Jukka Korpela*: Simple guidelines on using ALT texts in IMG elements (<http://www.cs.tut.fi/%7Ejkorpe/alt.html>)
- WAI "[Techniques for Web Content Accessibility Guidelines 1.0](http://www.w3.org/WAI/techniques/html1.0/)"

## Figure e diagrammi

Per ciascuna figura o diagramma di una pagina, aggiungere un riassunto testuale o inserire una descrizione testuale alternativa all'interno del codice pagina, in modo che queste possano essere sfruttate da un'interfaccia di assistenza.

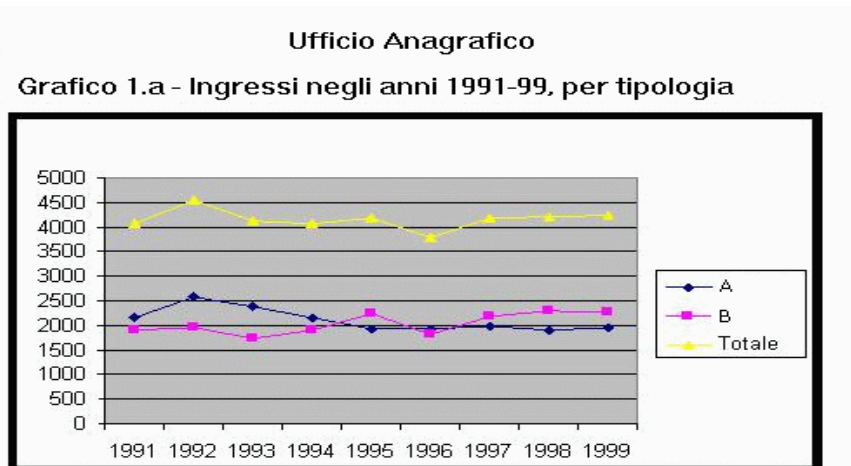
### Regola generale

Descrivere figure e diagrammi all'interno della pagina o utilizzare l'attributo **longdesc** dell'elemento **<img>**.

### Come procedere

- Utilizzare l'attributo **alt** per indicare in modo conciso il titolo della figura.
- Utilizzare l'attributo **longdesc** per fornire la descrizione della figura in una pagina (X)HTML con la seguente sintassi: **LONGDESC="descrizione della figura.htm"**
- Poiché solo gli screen reader di ultima generazione leggono l'attributo **longdesc**, è necessario prevedere un link aggiuntivo a fianco all'immagine, che punta alla stessa pagina contenuta nell'attributo **longdesc**. Tale link aggiuntivo prende il nome di **"D-link"** (description link), poiché per convenzione il nome del link è una **"D"** fra parentesi quadre (**[D]**).
- Il nome del file HTML contenente la descrizione della figura deve essere significativo, poiché lo screen reader lo legge; con ASPHI è stato concordato di adottare diciture come **"descrizione del grafico.htm"**, oppure **"descrizione del diagramma.htm"**, a seconda della tipologia della figura.
- Nel file **"descrizione della figura.htm"** inserire il titolo della figura anche all'interno dell'elemento **<title>**.
- Il testo per descrivere la figura deve essere molto semplice e sintetico: tipicamente la descrizione di un grafico sarà la tabella dei dati linearizzata. Alcuni esempi molto utili e istruzioni su come descrivere figure e diagrammi sono contenute nel documento **"NBA Tape Recording Manual"** (<http://www.w3.org/2000/08/nba-manual/>) pubblicato dal W3C ed estratto da un manuale prodotto dalla National Braille Association.

### Esempi



[D]

*Figura 1 – Esempio: grafico.*

**Codice XHTML:**

```

Ufficio Anagrafico
<p>Grafico 1.a - Ingressi negli anni 1991-99, per
tipologia</p><p>

<a href="descrizioneedelgrafico.html" title="Descrizione
del grafico">[D]</a> </p>

```

**Grafico 1.a - Ufficio anagrafico - Ingressi A e B, negli anni 1991-99, per tipologia**

Le statistiche sottoriportate illustrano gli ingressi di A e B, negli anni 1991-1999

ANNO 1991,	A : 2.100,	B : 976, Totale: 3.076
ANNO 1992,	A : 1.890,	B : 900, Totale: 2.790
ANNO 1993,	A : 1.200,	B : 886, Totale: 2.086
ANNO 1994,	A : 1.990,	B : 796, Totale: 2.786
ANNO 1995,	A : 2.000,	B : 970, Totale: 2.970
ANNO 1996,	A : 2.150,	B : 900, Totale: 3.050
ANNO 1997,	A : 2.050,	B : 950, Totale: 3.000
ANNO 1998,	A : 2.800,	B : 876, Totale: 3.676
ANNO 1999,	A : 2.020,	B : 776, Totale: 2.996

*Figura 2 – Esempio: descrizione testuale del grafico.*

**Codice HTML del file descrizioneedelgrafico.htm :**

```

Grafico 1.a - Ufficio Anagrafico - Ingressi A e B, negli
anni 1991-99, per tipologia.
<p>
Le statistiche sottoriporate illustrano il gli ingressi di
A e B negli anni 1991-1999
<p>
ANNO 1991, A: 2.100, B: 976, Totale: 3.076
<p>
ANNO 1992, A: 1.890, B: 900, Totale: 2.790
<p>
ANNO 1993, A: 1.200, B: 886, Totale: 2.086
<p>
ANNO 1994, A: 1.990, B: 796, Totale: 2.786
<p>
ANNO 1995, A: 2.000, B: 970, Totale: 2.970
<p>
ANNO 1996, A: 2.150, B: 900, Totale: 3.050
<p>
ANNO 1997, A: 2.050, B: 950, Totale: 3.000
<p>
ANNO 1998, A: 2.800, B:876, Totale: 3.676
<p>
ANNO 1999, A: 2.020, B: 776, Totale: 2.996

```

## Colore (requisito n.4 e n.6)

**Requisito n.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:

(Priorità 1)

[2.1](#) Assicurarsi che tutta l'informazione veicolata dal colore sia disponibile anche senza, ad esempio grazie al contesto o ai marcatori.

**Riferimenti Sec. 508:** *1194.22 (c)*

**Requisito n.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:

(Priorità 2)

[2.2](#) Assicurarsi che le combinazioni fra colori dello sfondo e del primo piano forniscano un sufficiente contrasto se visti da qualcuno con deficit percettivi sul colore o se visti su uno schermo in B&N. [Priorità 2 per le immagini, Priorità 3 per il testo].

**Riferimenti Sec. 508:** *non presente*

## Descrizione

- **Non usare colori per trasmettere o richiedere informazioni** a meno che queste non siano chiare anche dal markup o dal testo: ad esempio, frasi del tipo “*seleziona il bottone verde per avviare la procedura*” possono impedire ai daltonici di raggiungere l’informazione; inoltre se nel testo alternativo dell’immagine (attributo **alt**) non è specificato che si tratta di un bottone di colore verde, anche l’utente non vedente che utilizza uno screen reader non recepisce l’informazione e anche chi dispone di un monitor monocromatico non può distinguere i colori.
- **Usare combinazioni di colore per sfondo e primo piano che forniscano sufficiente contrasto** quando vengono visti con monitor monocromatici, o da persone con disturbi di percezione cromatica (ad esempio, font di colore chiaro su sfondo scuro). Il contrasto derivante da una combinazione studiata in modo appropriato consente a ipovedenti o a chi ha difficoltà di riconoscimento dei colori di leggere agevolmente il contenuto della pagina.
- Alcune indicazioni di base per l’uso dei colori:
  - il **rosso** e il **verde** sono i colori più problematici per i daltonici;
  - le **tonalità accese** sono poco riposanti e possono provocare abbagliamento;
  - le **tonalità medie**, soprattutto se combinate tra loro, sono a basso contrasto;
  - è meglio preferire **sfondi chiari e tenui** con testo decisamente scuro, ovvero sfondo decisamente scuro con testo chiaro, ma non troppo sgargiante;
  - il **basso contrasto** tra testo e sfondo dalle tonalità medie normalmente sconsigliato, può essere aumentato mettendo il testo in grassetto.

## Verifica del lavoro

- Per verificare la differenza di luminosità e di colore tra il testo e lo sfondo secondo l'algoritmo di verifica proposto dal W3C, utilizzare lo strumento:
  - **Colour Contrast Analyser** <http://juicystudio.com/services/colourcontrast.php>
- Per controllare un'immagine o una pagina web simulando diversi tipi di deficit visivo (la *protanopia*, la *deuteranopia* e la *tritanopia*), qualora ci sia informazione veicolata dal colore (in particolare rosso, verde, giallo e blu):
  - **Vischeck Colour Blindness Simulation**  
 Controllo di una immagine <http://www.vischeck.com/vischeck/vischeckImage.php>  
 Controllo di una pagina web <http://www.vischeck.com/vischeck/vischeckURL.php>
- Se necessario, utilizzare le simulazioni e gli strumenti della "Web Accessibility Toolbar" con Internet Explorer:
  - -> colour -> Grayscale (per verifica in B/N)
  - -> colour -> Colour used (ottengo la lista dei colori utilizzati nella pagina)
  - -> tools -> simulations (ci sono varie simulazioni di deficit visivo)

## Approfondimenti

- *Diodati.org*: Uso accessibile del colore - [http://www.diodati.org/scritti/2004/guida/ele\\_acc16.asp](http://www.diodati.org/scritti/2004/guida/ele_acc16.asp).
- *Cal Henderson*: Color Vision - <http://www.iamcal.com/toys/colors/> - Mostra lo spettro dei colori come percepiti dai vari tipi di disabilità visiva associata ai colori.
- *Ishihara Test for Color Blindness*: <http://www.toledo-bend.com/colorblind/Ishihara.html> - Contiene una serie di immagini che le persone con disabilità al rosso-verde vedono diversamente o addirittura non vedono del tutto.
- *AWARE Center*: The Color Laboratory - <http://colorlab.wickline.org/colorblind/colorlab/>

## Elementi in movimento e lampeggianti (requisito n.5)

**Requisito n.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: 7.3

#### (Priorità 1)

7.1 Fino a quando gli interpreti non permetteranno agli utenti di controllare lo sfarfallio, evitare di far sfarfallare lo schermo.

#### (Priorità 2)

7.2 Fino a quando gli interpreti non permetteranno agli utenti di controllare il lampeggiamento, evitare di far lampeggiare il contenuto (cioè di cambiare la presentazione a intervalli regolari, come se si accendesse e spengesse).

### Riferimenti Sec. 508: 1194.22 (j)

### Descrizione

Spesso vengono utilizzate, nella realizzazione di pagine web, elementi in movimento o lampeggianti.

Gli utenti con disabilità visive hanno difficoltà ad interagire con i testi in movimento, questo perché, o non riescono a leggerli con velocità tale da comprenderli, o la maggior parte degli *screen reader* non catturano questi tipi di testo.

Inoltre, le persone con disabilità fisiche spesso non riescono a muoversi con velocità o precisione sufficienti ad interagire con oggetti in movimento.

Persone con epilessia fotosensibile possono avere crisi scatenate da sfarfallio oppure da lampeggiamenti nell'intervallo che va da 4 a 59 lampi al secondo (Hertz), con un picco di sensibilità intorno ai 20 lampi al secondo, così come da mutamenti repentini di oscurità e luce (come nel caso di luci intermittenti).

Con le ultime versioni degli *screen reader*, gli utenti con disabilità della vista leggono testi in movimento come quelli generati dagli applets e dai javascripts, ma ne sono totalmente incapaci se navigano in Internet con **strumenti di ausilio più vecchi**.

### Come procedere

- Evitare qualsiasi effetto lampeggiante o aggiornamento dello schermo che causi sfarfallio.
- Il contenuto di una pagina non deve lampeggiare. Le informazioni presenti nella pagina, cioè, non devono cambiare la loro presentazione a intervalli regolari, come se si accendessero e si spegnessero.
- Assicurarsi che gli oggetti in movimento, lampeggianti, scorrevoli o che si autoaggiornano possano essere arrestati temporaneamente o definitivamente.
- Se è assolutamente necessario sottolineare un testo facendolo lampeggiare, allora bisogna usare l'attributo dei CSS: "**text-decoration: blink**". Usando questo attributo l'utente potrà fermare l'effetto disabilitando il foglio di stile.
- È preferibile non usare gli attributi **blink** e **marquee**, perché questi non sono inclusi nelle raccomandazioni del W3C.

## Immagini sensibili

(requisito n.7 e n.8)

**Requisito n. 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:

(Priorità 1)

[9.1](#) Fornire immagini sensibili sul lato client invece di immagini sensibili sul lato server, con l'eccezione dei casi nei quali le zone non possono essere definite con una forma geometrica valida.

(Priorità 3)

[1.5](#) Fino a quando gli interpreti non renderanno disponibili equivalenti testuali per collegamenti di immagini sensibili sul lato client, fornire collegamenti di testo ridondanti per ogni zona attiva di una immagine sensibile sul lato client.

**Riferimenti Sec. 508:** 1194.22 (f)

**Requisito n. 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:

(Priorità 1)

[1.2](#) Fornire ridondanti collegamenti di testo per ogni zona attiva di una immagine sensibile sul lato server.

**Riferimenti Sec. 508:** 1194.22 (e)

## Descrizione

Le **immagini sensibili**, o mappe cliccabili (**image maps**), sono divise in regioni attive (o aree cliccabili), ossia in regioni associate ad azioni (ad esempio recuperare un documento o eseguire un programma). Quando l'utente clicca su una regione attiva, l'azione viene eseguita, attivando il link verso un'altra pagina associato a quella regione.

Solo le mappe **client-side** (cioè quelle interamente contenute nel documento) risultano accessibili, in quanto codificano questa attività disponibile alle tecnologie assistive.

## Regola generale

Utilizzare l'elemento **<map>** per definire la mappa cliccabile, e gli elementi AREA per specificare le regioni attive della mappa (aree cliccabili) e il collegamento ad esse associato.

## Come procedere

- La mappa deve essere **client-side**, cioè interpretata senza alcun intervento da parte del server (usare l'elemento **<map>** per definire la mappa e l'attributo **usemap** di **<img>** per associare la mappa all'immagine).
- Utilizzare l'attributo **alt** per fornire la descrizione della mappa. Il testo deve essere un titolo significativo, perché serve allo screen reader per introdurre l'immagine sensibile.
- Fare ricorso all'attributo **alt** per fornire la descrizione delle singole aree cliccabili, e l'attributo **title** per fornire un testo alternativo di ogni area che presenti un link.

- Laddove ci sia una mappa che contiene informazioni che non portano ad aree cliccabili (ad esempio gli organigrammi), servirsi comunque dell'attributo **alt** per descrivere le aree non cliccabili.

### Esempio



*Figura 1 - Esempio grafico: cartina dell'Italia.*

#### Codice HTML del file di figura 1

```
<div align="center">

</div>


<map name="italia">

< area shape="poly" alt="Lazio"
coords="40,50,33,48,12,54,20,68,43,60,40,50"
href="lazio.htm"></map>
```

#### Note

- Se la mappa immagine contiene informazioni che non portano ad aree cliccabili, utilizzare comunque l'attributo **alt** per descrivere le aree non cliccabili.
- La descrizione del link va apposta nell'attributo **title** di ogni area che presenti un link.

## Lo strumento BS-D

Dalla "Barra di formattazione", selezionare 'Aggiungi METAFILE' per caricare un'immagine mappa ciccabile nel direttorio e selezionare il simbolo  per salvare l'immagine mappa ciccabile nella pagina web



### Verifica del lavoro

Verificare che il contenuto informativo e le funzionalità della pagina siano ancora fruibili in caso di disattivazione del caricamento delle immagini.

Effettuare una verifica manuale simulando la disattivazione delle immagini con uno dei seguenti browser:

- Internet Explorer: Web Accessibility Toolbar -> images -> Toggle Image/Alt
- Mozilla Firefox: Tools -> options -> web features -> proprietà web -> load images
- Netscape: preferences -> privacy&security -> images -> do not load any images
- Opera 6: file -> preferences -> Multimedia -> Images
- Opera 7 e 8: tools -> preferences -> web pages -> Images

La Web Accessibility Toolbar è scaricabile all'indirizzo:

<http://www.nils.org.au/ais/web/resources/toolbar/index.html>

Se necessario effettuare anche una verifica automatica con uno dei seguenti strumenti automatici:

- Analizzatore di immagini di JuicyStudio ([www.juicystudio.com/services/image.asp](http://www.juicystudio.com/services/image.asp))
- Servizio Alt Text Quality di "Cynthia Says Portal" (<http://www.cynthiasays.com/fulloptions.asp>)

### Approfondimenti

- A. J. Flavell (traduzione a cura di Diodati) : Uso dei test ALT all'interno di IMG ([http://www.diodati.org/scritti/2002/g\\_alt/index.asp](http://www.diodati.org/scritti/2002/g_alt/index.asp)) .
- WebAIM: How to Create Accessible Graphics (<http://www.webaim.org/techniques/images/>) .
- Jukka Korpela: Simple guidelines on using ALT texts in IMG elements (<http://www.cs.tut.fi/%7Ejkorpela/html/alt.html>) .

## Tabelle di dati

(requisito n.9 e n.10)

**Requisito n. 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:

(Priorità 1)

[5.1](#) Per tabelle di dati, identificare le intestazioni di righe e colonne.

(Priorità 3)

[5.5](#) Per le tabelle, fornire sommari.

[5.6](#) Fornire abbreviazioni per le etichette di intestazione.

**Riferimenti Sec. 508:** 1194.22 (g)

**Requisito n. 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:

(Priorità 1)

[5.2](#) Per tabelle di 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.

**Riferimenti Sec. 508:** 1194.22 (h)

## Descrizione

Le **tabelle** dovrebbero essere usate per marcare informazioni realmente tabellari, dette appunto "tabelle di dati". Ad esempio, sono tabelle di dati, un calendario, un orario delle lezioni, una rubrica di nomi, indirizzi e numeri telefonici. È necessario assicurarsi che le tabelle abbiano la marcatura necessaria per essere trasformate dai software di tecnologia assistiva.

## Regole generali

Prima di procedere con l'applicazione delle regole di accessibilità per ciascuna tabella di dati, lo sviluppatore deve effettuare un'analisi critica valutando due aspetti fondamentali:

- **semplificare** la tabella dove possibile
- **valutare** la **tipologia** della tabella e, sulla base di questa, seguire le raccomandazioni volte all'accessibilità.

Per quanto riguarda il primo punto, è importante semplificare al massimo le tabelle: evitare righe e/o colonne con dati insignificanti. In presenza di tabelle troppo complesse, lo sviluppatore deve considerare se è opportuno suddividere i dati della tabella in più tabelle distinte.

Nell'esempio proposto in Figura 1, la tabella potrebbe essere semplificata eliminando le prime due colonne e inserendo il rispettivo contenuto come titolo o all'interno della descrizione della stessa tabella.

Librerie	Categoria	Diritti forfettari prestati (per tipologia di libri)				
	LIBRI	N. giorni	A	B	C	D
		Fino a 4	€ 0,77	€ 2,32	€ 5,16	€ 15,49
		Fino a 10	€ 1,55	€ 4,65	€ 6,20	€ 18,59
Fino a 20	€ 3,10	€ 9,30	€ 7,23	€ 21,69		

*Figura 1 - Esempio di tabella che potrebbe essere semplificata.*

Di seguito riportiamo alcune raccomandazioni generali, che consentono di rendere accessibili le tabelle di dati.

### Come procedere

#### Titolo e sintesi della tabella

- Utilizzare l'attributo **summary** dell'elemento `<table>`, che permette di inserire una sintesi del contenuto e della struttura della tabella. È un valido aiuto per la comprensione della tabella da parte di utenti non vedenti.
- Utilizzare l'elemento `<caption>` per inserire un titolo significativo nella tabella di dati. Tale elemento è consentito solo immediatamente dopo l'elemento iniziale `<table>`. Il `<caption>` produce, a livello di impaginazione, il titolo della tabella, ed è localizzato esternamente alla tabella. Un elemento `<table>` può contenere solo ed esclusivamente un elemento `<caption>`. Se nella tabella è presente l'attributo **summary** significativo, l'elemento `<caption>` può essere omesso. In alternativa, è possibile inserire il titolo della tabella come semplice testo, immediatamente prima del `<table>`.
- In presenza di una tabella, costituita da una o n righe e una sola colonna, lo screen reader **Jaws**, versione 4.50, ignora che è una tabella, in quanto legge sequenzialmente il contenuto nelle singole righe; in questo caso, anche l'attributo `summary` viene ignorato dallo screen reader.

#### Struttura della tabella

- Identificare la tabella attraverso l'elemento di apertura `<table>` e quello di chiusura `</table>`, entrambi obbligatori.
- Identificare le celle di dati con l'elemento `<td>` e ciascuna riga con l'elemento `<tr>`.
- Inserire sempre le intestazioni di righe e di colonne, usando l'elemento `<th>` poiché se assenti, lo screen reader interpreta erroneamente la prima riga della tabella come riga di intestazione.

#### Ampiezza della tabella

- Specificare l'ampiezza della tabella con l'attributo **width** dell'elemento `<table>`. Si consiglia di esprimere l'ampiezza della tabella in percentuale anziché in pixel (ad esempio: `table width="80%"`), per questioni di corretta trasformazione alle diverse risoluzioni video.
- È preferibile non inserire l'ampiezza di colonna, in quanto obbliga lo screen reader ad una larghezza raccomandata.



### Presentazione e formattazione della tabella

- Si consiglia di usare i fogli di stile per controllare l'impaginazione e la presentazione dei dati presenti in tabella. Ad esempio, si possono prevedere delle classi per definire gli attributi della tabella (colore, stile e misura del carattere) e la formattazione delle intestazioni delle tabelle.

### Gestione delle celle vuote

- Evitare la presenza di righe e/o colonne completamente bianche, all'interno della tabella. Lo screen reader legge tutte le righe, anche quelle bianche e ciò può creare confusione.
- Identificare le celle vuote con un carattere (ad esempio: il meno "-") o indicare il contenuto in ciascuna cella vuota, se può servire a chiarire meglio la lettura da parte dello screen reader.
- Identificare i dati mancanti nelle celle della tabella con un'abbreviazione (ad esempio: "n.p.", non presente), quando il dato mancante esprime un valore.
- Inserire in testa alla tabella una legenda che spieghi il significato dei caratteri o delle abbreviazioni utilizzate, oppure descrivere la legenda all'interno dell'attributo **summary**.

### Lo strumento BS-D

Dalla "Barra di formattazione", selezionare il simbolo  per fare una tabella semplice con intestazione di colonne e/o di righe e il simbolo  per fare una tabella con più livelli di intestazione di colonne e/o righe.



Proponiamo ora alcune tipologie di tabelle dati con le indicazioni di come procedere per renderle accessibili.

### Esempio di tabella semplice con intestazione di colonne e/o righe

Intestazione prima colonna	Intestazione seconda colonna
Prima cella	Seconda cella

```
<table border="1" cellpadding="2" cellspacing="2" summary="sommario">
  <tr>
    <th scope="col">Intestazione prima colonna</th>
    <th scope="col">Intestazione seconda colonna</th>
  </tr>
  <tr>
    <td>Prima cella</td>
    <td>seconda cella</td>
  </tr>
</table>
```

*Caratteristiche:*

Le intestazioni di colonna sono etichettate con l'attributo **scope**, associato all' elemento **<th>**. Questo attributo specifica l'insieme di celle di dati per le quali la cella d'intestazione corrente fornisce informazioni d'intestazione.

L'attributo **scope** deve avere uno dei seguenti valori:

- **"col"**: la cella corrente fornisce informazioni di intestazione per il resto della colonna che la contiene;
- **"row"**: la cella corrente fornisce informazioni di intestazione per il resto della riga che la contiene.

## Esempio di tabella con più livelli di intestazione di colonne e/o righe

Intestazione prima colonna	Intestazione seconda colonna	Intestazione terza colonna
Prima cella della prima riga	Seconda cella della prima riga	Terza cella della prima riga
Prima cella della seconda riga	Seconda cella della seconda riga	Terza cella della seconda riga

### Codice HTML:

```
<table border="1" cellpadding="2" cellspacing="2" summary="sommario">
  <tr>
    <th id="c1">Intestazione prima colonna</th>
    <th id="c2">Intestazione seconda colonna</th>
    <th id="c3">Intestazione terza colonna</th>
  </tr>
  <tr>
    <th id="r1">Prima cella della prima riga</th>
    <td headers="r1 c2">Seconda cella della prima riga</td>
    <td headers="r1 c3">Terza cella della prima riga</td>
  </tr>
  <tr>
    <th id="r2">Prima cella della seconda riga</th>
    <td headers="r2 c2">Seconda cella della seconda riga</td>
    <td headers="r2 c3">Terza cella della seconda riga</td>
  </tr>
</table>
```

*Caratteristiche:*

Le intestazioni di riga o di colonna sono etichettate con l'attributo **headers** del l'elemento **<th>**. Tale attributo specifica l'elenco di celle d'intestazione che forniscono informazioni d'intestazione per la cella di dati corrente. Il valore di questo attributo è un elenco di nomi di celle separato da spazi. Tali celle vengono prima denominate attraverso l'uso dell'attributo **id**. In particolare, l'attributo **headers** è importante quando le intestazioni sono posizionate irregolarmente rispetto ai dati ai quali si riferiscono.

## Tabella complessa (non riconducibile ad alcuna delle tipologie qui proposte)

Se la tabella non è comunque visibile, fornire un'alternativa equivalente nei contenuti, ad esempio una versione linearizzata testuale.

### Verifica del lavoro

- Assicurarsi che il contenuto e la struttura della tabella risultino chiari anche quando la tabella stessa viene letta cella dopo cella e una riga alla volta, assicurarsi cioè che **la tabella risulti chiara anche quando è linearizzata**.
- Si consigliano alcuni strumenti in grado di verificare l'accessibilità di una tabella di dati:
  - Internet Explorer: [Web Accessibility Toolbar](#) --> Document Structure
    - --> Table borders
    - --> Linearize (remove Tables)
  - **Lynx** (copyrighted by the University of Kansas, distribuito gratuitamente da the GNU General Public License, si veda: <http://lynx.browser.org>), browser testuale, legge le tabelle in modo linearizzato, pertanto simula la lettura da parte di uno *screen reader* (scaricare il browser in locale e procedere con la lettura della pagina contenente la tabella).
  - **Tablin** (un programma sviluppato dal WAI, si veda: <http://www.w3.org/WAI/Resources/Tablin/>), programma disponibile via web, consente allo sviluppatore di sottoporre l'URL della pagina web contenente la tabella e di ottenere l'output della tabella linearizzata.

### Approfondimenti

- *Jim Byrne*: Table Manners - <http://www.mcu.org.uk/articles/tables.html>.
- *WebAIM*: Create tables that transform gracefully - <http://www.webaim.org/techniques/tables/>.
- *Kynn Bartlett*: Understanding Accessible Table Markup - <http://access.idyllmtn.com/tables/>.

## Fogli di stile (requisito n.11)

**Requisito n.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:

(Priorità 1)

[6.1](#) Organizzare i documenti in modo che possano essere letti senza i fogli di stile. Per esempio, quando un documento HTML viene reso senza i fogli di stile associati, deve essere sempre possibile leggere il documento.

(Priorità 2)

[3.3](#) Usare fogli di stile per controllare l'impaginazione e la presentazione.

**Riferimenti Sec. 508:** 1194.22 (d)

### Descrizione

Adottando i **fogli di stile** detti anche **CSS** (acronimo di "Cascading Style Sheets"), gli sviluppatori di pagine (X)HTML potranno esercitare un controllo più accurato sulle pagine ed alleggerire il codice, ottenendo una migliore navigabilità delle pagine, assicurando nel contempo una piena accessibilità ai disabili, comprese le persone con problemi di ipovedenza.

Con i fogli di stile è possibile liberare il testo da vincoli di dimensione: ciò è di fondamentale importanza per gli ipovedenti, poiché possono modificare secondo le loro esigenze la dimensione predefinita dei caratteri. Il nucleo di un CSS è al contempo estremamente semplice e potente. Consente di associare regole stilistiche agli elementi del codice (X)HTML. Queste regole definiscono l'aspetto rappresentativo degli elementi (X)HTML a cui vengono applicati e offrono potenti strumenti di controllo tipografico e di visualizzazione, permettendo di specificare non solo il carattere, ma anche la sua dimensione, l'interlinea e gestire ampie aree o certi elementi specifici del documento con una data regola, fino a utilizzare i CSS per definire complesse "gabbie" grafiche della pagina.

I fogli stile CSS vengono supportati bene dai browser più recenti, ma browser piuttosto datati come Internet Explorer 4 e Netscape 4 hanno invece una pessima interpretazione dei CSS. È importante precisare che il mancato supporto al CSS da parte di **Netscape 4** e Internet **Explorer 4** si limita al solo aspetto grafico, non all'accesso dei contenuti e alla navigazione della pagina e del sito. È bene dunque valutare quanto sia utile e abbia senso limitare le potenzialità dei CSS e l'adozione degli standard al fine di ottenere una presentazione visuale uguale sul maggior numero possibile di browser grafici in uso.

L'esperienza maturata nel campo dell'accessibilità ci insegna che l'obiettivo primario è ottenere la **massima fruibilità** dei **contenuti**, **non** la massima uniformità delle presentazioni. Una pagina è accessibile non perché si vede allo stesso modo su tutti i browser grafici, ma perché i suoi contenuti possono essere navigati con qualsiasi programma utente, anche non visuale.

Tali indicazioni sono anche confermate dal DM 8 luglio 2005, Allegato A che afferma:

- il *contenuto informativo* e *le funzionalità* presenti in una pagina siano gli stessi in varie versioni di diversi browser grafici, in vari sistemi operativi;

- la *presentazione* della pagina (impaginazione, stile e grafica) sia simile nei browser che supportano gli standard di linguaggio XHTML Strict e CSS level 2.

Per risolvere il problema di compatibilità con il browser Netscape 4, si consiglia l'utilizzo del cosiddetto "**metodo @import**" e 2 possibili scelte basate su tale metodo:

**Scelta n.1:** non utilizzare alcun foglio di stile per Netscape 4 (*consigliata*)

*Esempio:*

```
<head>
<!-- Risolve il FOUC -->
<script type="text/javascript"> </script>
<style rel="stylesheet" type="text/css" media="all">
    @import url("css_avanzato.css");
</style>
</head>
```

Netscape 4 non è in grado di interpretare l'elemento **@import**, quindi lo ignora e non carica il CSS. In questo modo la pagina visualizzata tramite Netscape 4 sarà priva di layout, quindi sarà una versione solo testuale, ma il *contenuto informativo* e *le funzionalità* saranno completamente accessibili

N.B. Per il cosiddetto FOUC si veda <http://www.bluerobot.com/web/css/fouc.asp>

**Scelta n.2:** creare un foglio di stile apposito per Netscape 4

```
<head>
<link rel="stylesheet" type="text/css href="css_x_nn4.css" />
<style type="text/css">
    @import url("css_avanzato.css");
</style>
</head>
```

Netscape 4 interpreta correttamente la stringa:

```
<link rel="stylesheet" type="text/css href="css_x_nn4.css" />
```

di conseguenza carica il foglio stile "css\_x\_nn4.css", che deve essere appositamente preparato per Netscape 4 (CSS level 1).

Con i CSS è possibile creare dei **layout personalizzati**; in particolare si possono definire vari fogli di stile CSS per diverse esigenze, associati allo stesso documento (X)HTML.

Gli autori possono specificare un certo numero di fogli di stile mutuamente esclusivi, chiamati **fogli di stile alternativi** e possono indicarne uno preferenziale. Sta all'autore implementare un sistema che consenta all'utente di scegliere dinamicamente lo stile alternativo sostituendolo al primo, ad esempio attraverso un javascript, in grado di intervenire su proprietà e attributi dell'elemento **<link>**.

Gli interpreti devono anche permettere agli utenti di disabilitare interamente i fogli di stile dell'autore, nel qual caso l'interprete non deve applicare alcun foglio di stile, principale o alternativo.

Le tecniche di personalizzazione permettono di creare delle versioni alternative del sito, senza bisogno di creare dello stesso una copia parallela accessibile.

È importante che sia l'utente a scegliere come navigare i contenuti, l'interfaccia infatti dovrebbe essere in grado di adattare i contenuti alle esigenze di ogni singolo utente. L'accessibilità viene intesa come **nuovo criterio progettuale universale**, basato su tecnologie che migliorano la vita di ogni cittadino, semplificano le interazioni uomo-macchina, favoriscono le prestazioni, propongono agli utenti una molteplicità di soluzioni (le esigenze di navigazione di un ipovedente sono diverse da quelle di un utente che utilizza un browser vocale o da quelle di un utente che usa un browser grafico molto avanzato).

Alcuni esempi:

- Versione ad **alta leggibilità**: definisco un foglio di stile CSS con caratteri grandi, colori contrastati (sfondo scuro e testo di colore chiaro), disattivo il menu grafico e attivo un menu basato su testo.
- Versione **solo testo**: definisco un foglio di stile CSS eliminando completamente la grafica e mantenendo solo la struttura e i contenuti.
- **Ridimensionare i caratteri**: l'utente può impostare la grandezza dei caratteri grazie a un foglio di stile CSS dove i caratteri sono impostati con unità di lunghezza relativa.
- Versione **senza/con la struttura di navigazione**: definisco un foglio di stile CSS con la struttura di navigazione disattivata.
- Versione "grafica" **alternativa**: definisco un foglio di stile CSS con una grafica alternativa a quella principale, ad esempio tonalità, sfondi e caratteri con stili diversi.
- Versione per la **stampa**: definisco un foglio di stile specifico per la stampante, ovvero elimino tutto ciò che è superfluo (il menu di navigazione), imposto i colori adatti alla stampa, definisco i caratteri per la stampante (è preferibile usare i caratteri **serif** come il Georgia o il Times). Definisco la dimensione dei caratteri attraverso i punti (**pt**), più adatta per la carta, inoltre posso specificare dimensioni della pagina in millimetri e impostare bordi, margini, larghezze con precisione millimetrica. Infine si possono evidenziare i link e stampare anche l'indirizzo vero e proprio (**URL**).

## Regola generale

Usare fogli di stile per controllare l'impaginazione e la presentazione dei documenti, evitando così l'uso di tabelle di layout. Le istruzioni dei fogli di stile possono essere scritte in un documento separato con estensione **.css** nel quale si introducono in formato testo le indicazioni di stile da applicare a tutte le pagine HTML contenute in un sito web. È importante l'uso della proprietà dei caratteri CSS invece che l'elemento **<font>** per controllare gli stili di caratteri.

## Come procedere

- Usare l'attributo **href** per indicare la locazione del documento di foglio di stile.
- Per rendere un foglio di stile preferenziale o alternativo, impostare l'attributo **rel** rispettivamente al valore "stylesheet" o "alternate stylesheet" e attribuire un nome al foglio di stile per mezzo dell'attributo **title**. Il title permette agli utenti di visionare e scegliere dalla lista di stili alternativi.
- Usare unità relative e non assolute nei valori della proprietà del foglio di stile, "**em**" e misure di percentuale, invece di "pt", "cm", "px" che sono misure assolute. È possibile scegliere di

utilizzare misure assolute, avendo cura di accertarsi che il contenuto espresso non presenti difficoltà di fruizione.

- Nelle pagine che usano un foglio di stile, assicurarsi che i contenuti siano ordinati e strutturati cosicché possano essere letti nell'ordine voluto, anche quando il foglio di stile non viene utilizzato.
- Organizzare i documenti in modo che possano essere letti senza i fogli di stile. Ad esempio, quando un documento (X)HTML viene reso senza i fogli di stile associati, deve essere sempre possibile leggere il documento.

### Esempi

```
<link rel="stylesheet" href="default.css" type="text/css" media="screen"
title="Default" />
<link rel="stylesheet" type="text/css" href="print.css" media="print" />
<link rel="alternate stylesheet" type="text/css" href="alto_contrasto.css"
media="screen" title="Alto contrasto" />
<link rel="alternate stylesheet" type="text/css" href="no_menu.css"
media="screen" title="Menu nascosto" />
<link rel="alternate stylesheet" type="text/css" href="orange.css"
media="screen" title="Orange" />
<link rel="alternate stylesheet" type="text/css" href="solotesto.css"
media="screen" title="Solo testo" />
```

### Fogli di stile con l'attributo media

Attraverso l'utilizzo dei fogli di stile alternativi, si può specificare come il documento deve apparire in diversi **media**: palmare, carta, dispositivo Braille... L'attributo media inserito in una pagina (X)HTML imposta un foglio di stile per uno specifico dispositivo. Associando l'attributo media all'elemento "**style**" o "**link**", l'interprete deve filtrare i fogli di stile che non si applicano al medium corrente.

Alcuni dei **media type** disponibili sono:

- **screen** (per pagine destinate ad essere visualizzate sul monitor del computer);
- **print** (per contenuti da stampare);
- **aural** (per la riproduzione tramite sintetizzatore vocale);
- **Braille** (per la lettura tattile, usata soprattutto dai non vedenti);
- **handheld** (per la riproduzione su computer palmari).

L'elenco completo è contenuto nelle specifiche **CSS2** (<http://www.w3.org/TR/REC-CSS2/media-types>) e comprende i vari tipi di periferica su cui un contenuto web può essere riprodotto.

Tra i fogli di stile alternativi più utilizzati vi è quello auditivo, l'**aural** (aural media styles), per la riproduzione tramite sintetizzatore vocale. Tali fogli di stile auditivi permettono di controllare tempi, suoni, voci, spazio, volume e altre caratteristiche in modo molto elaborato.

L'uso degli **aural media styles** è destinato a modificare le caratteristiche audio delle sezioni ed elementi di una pagina: definiscono quindi l'intensità, il volume, la durata delle pause tra parole o

sezioni del documento, la velocità del parlato e tutte le caratteristiche che possono meglio rendere il suono gradevole.

Queste funzionalità davvero interessanti, enunciate nelle specifiche CSS2 del W3C <http://www.w3.org/TR/REC-CSS2/media.html>, purtroppo ad oggi risultano ancora teoriche, poiché i vari browser attualmente in uso offrono un supporto non uniforme né universale al cambio automatico di foglio di stile a seconda del tipo di media. Le ragioni principali sono lo scarso supporto fornito dai vari programmi utente (sia browser grafici, sia tecnologie assistive), la ridotta conoscenza di questi strumenti autoriali da parte degli sviluppatori e veri e propri limiti di concezione dei sistemi media type previsti dalle specifiche CSS2.

Questo non vuol dire che non è possibile costruire grazie ai CSS versioni alternative di un documento per diversi media, ma che conviene utilizzare una strategia alternativa al cambio automatico dei media types, cioè quella di presentare all'utente dei **collegamenti espliciti** a versioni alternative di uno stesso documento, tarate per la presentazione su media diversi dal tradizionale monitor di computer.

## Approfondimenti

Risorse su Web:

- *Diodati.org*: "L'alternativa accessibile: formattare utilizzando esclusivamente i CSS level 2" [http://www.diodati.org/scritti/2004/guida/ele\\_acc11.asp](http://www.diodati.org/scritti/2004/guida/ele_acc11.asp)
- Csst-discuss mailing list: <http://css-discuss.incutio.com/>
- Meyer web site: CSS information - <http://www.meyerweb.com/eric/css/>
- Constile.org: sito italiano dedicato all'uso avanzato dei fogli di stile <http://www.constile.org/>
- W3C: CSS 2.1 Specification- <http://www.w3.org/TR/CSS21/>
- W3C: CSS1 Specification - <http://www.w3.org/TR/CSS1>
- W3C School: CSS Tutorial - <http://www.w3schools.com/css/default.asp>
- Dorward Online CSS: Resources for getting started with Cascading Style Sheets - <http://dorward.me.uk/www/css/>
- CSS Zengarden: <http://www.csszengarden.com/>
- A list Apart: Alternative Style: Working With Alternate Style Sheets <http://www.alistapart.com/articles/alternate/>

Libri:

- “*CSS Guida completa*” - Gianluca Troiani - Novembre 2005
- “*Lo Zen e il design CSS Nuova luce nell'arte del Web*” - Shea Dave; Holzschlag Molly E. - Luglio 2005
- “*Guida ai CSS. Trucchi, segreti e soluzioni*” - Andrew Rachel - Giugno 2005
- “*CSS guida pocket*” - seconda edizione - Meyer Eric A. - Maggio 2005
- “*Cascading Style Sheets: La guida completa*” - seconda edizione (CSS 2 e CSS 2.1) - Meyer Eric A. - Dicembre 2004
- “*Progettare il Web del futuro - Standard e tecniche per il design*” - Zeldman Jeffrey - Dicembre 2003
- “*Cascading Style Sheet (CSS) - Fogli di stile per il Web*” - Briggs Owen, Champeon Steven, Costello Eric, Patterson Matt - Ottobre 2002

## Dimensionamento dei caratteri

(requisito n.12)

**Requisito n.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:**  
(Priorità 2)

[3.4](#) Usare unità relative e non assolute nei valori degli attributi del linguaggio dei marcatori e i valori della proprietà del foglio di stile.

**Riferimenti Sec. 508:** *non presente*

### Descrizione

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.

### Come procedere

- Usare le proprietà dei caratteri del foglio stile CSS invece che l'elemento `<font>` per controllare gli stili di caratteri.
- Usare **unità relative** e non assolute nei valori della proprietà del foglio di stile, per esempio "em" o misure di percentuale, invece di "pt", o "cm", "px" che sono misure assolute.

### Lo strumento BS-D

Dalla "Barra di formattazione", selezionare i simboli **T T T** per scegliere lo stile più appropriato per "Titolo", "Sottotitolo" e "Titoletto".



### Verifica del lavoro

Effettuare una verifica manuale sui seguenti browser, provando ad aumentare o diminuire la dimensione dei caratteri :

- Netscape: view -> text zoom -> larger/smaller
- Mozilla firefox: view -> text size -> increase/decrease
- Internet Explorer: tenere premuto il tasto CTRL e girare la rotellina del mouse
- Opera: view -> zoom -> cambiare la %

### Approfondimenti

- Usare dimensioni relative per i font [http://www.francocarcillo.it/dive/day\\_26.html](http://www.francocarcillo.it/dive/day_26.html)
- Sane CSS Sizes <http://www.thenoodleincident.com/tutorials/typography/index.html>

## Tabelle per l'impaginazione (requisito n.13)

**Requisito n.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:

(Priorità 2)

[5.3](#) Non usare tabelle per impaginazioni a meno che la tabella non sia comprensibile se letta in modo linearizzato. Altrimenti, se la tabella non risulta leggibile, fornire un'alternativa equivalente (che può essere una versione linearizzata).

[5.4](#) Se per l'impaginazione viene usata una tabella, non usare nessun marcatore di struttura per la formattazione della resa visiva.

**Riferimenti Sec. 508:** *non presente*

### Descrizione

Le tabelle d'impaginazione sono utilizzate per disporre i contenuti sulla pagina in posizioni precise, ad esempio una griglia di righe e colonne i cui contenuti riproducono un'impaginazione basata su testata, tre colonne centrali e pie' di pagina oppure una tabella che contiene, insieme ad elementi grafici e testo, immagini spaziatrici invisibili, utilizzate per distanziare i contenuti visibili sulla pagina.

In linea di massima le tabelle di impaginazione sono sconsigliate, perché il loro utilizzo **rallenta il caricamento della pagina** ed aumenta la **quantità di metainformazioni** che i sintetizzatori vocali leggono all'utente.

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.

Per evitare problemi di comprensione dell'informazione contenuta nella tabella da parte dell'utente che usa strumenti alternativi, è importante che lo sviluppatore sappia come lo screen reader legge il testo inserito in una tabella.

	Colonna 1	Colonna 2
Riga 1	Cella 1.1	Cella 1.2
Riga 2	Cella 2.1	Cella 2.2

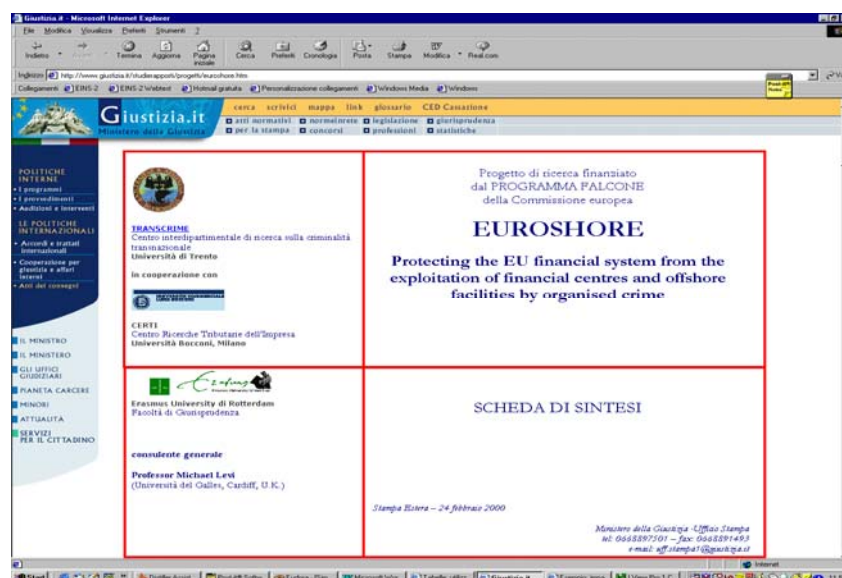
*Figura 1 - Esempio: tabella.*

Ad esempio, nella tabella riportata in Figura 1, lo screen reader inizia col leggere la prima riga, legge l'intero contenuto della cella 1.1, poi l'intero contenuto della cella 1.2. Procede quindi, con lo stesso metodo, alla lettura della seconda riga. La tabella multicolonne viene perciò

rilevata e letta come se i suoi contenuti fossero disposti su una sola riga. Questo metodo di lettura è chiamato **linearizzazione**.

Spesso gli sviluppatori eccedono nell'uso nidificato di tabelle (una dentro l'altra), per ottenere il posizionamento degli elementi nei punti desiderati della pagina HTML (X)HTML. In tal modo vanificano le possibilità di lettura degli screen reader, impedendo di fatto l'accessibilità alla pagina.

L'esempio seguente è riferito a una tabella di impaginazione (Figura 2) che crea problemi di accessibilità poiché il titolo del documento è situato nella prima cella della seconda colonna e i loghi sono distribuiti nelle due celle della prima colonna, separati da marcatori strutturali (<p>, <br>). Il testo non risulta di conseguenza immediatamente comprensibile, poiché lo screen reader inizia a leggere prima le informazioni della prima cella della tabella (relative a enti partecipanti al progetto di ricerca Euroshore) e poi legge il titolo del documento (in Figura 2 la griglia in rosso evidenzia la struttura della tabella).



**Figura 2 - Esempio: tabella di impaginazione e relativa struttura.**

Un altro esempio di utilizzo di una tabella per l'impaginazione ai fini della formattazione visiva sono i moduli fac-simile o schemi (Figura 3), in cui oltre a usare la tabella per l'impaginazione, viene utilizzato il simbolo “\_” (underscore) o “.” (punto) per identificare uno spazio per la compilazione (\_\_\_\_\_ oppure .....): in questo caso lo screen reader interpreta lo "spazio per la compilazione", leggendo tre volte di seguito il relativo simbolo.

Per evitare che lo screen reader abbia problemi di lettura e di interpretazione della pagina XHTML di un modulo fac-simile, conviene convertire il modulo in formato PDF e richiamarlo come link dalla pagina.

È importante che il PDF sia accessibile, cioè completamente leggibile e ben interpretato dallo screen reader: se ad esempio il file PDF deriva da un documento word, è importante che l'autore scriva il documento in formato .doc rispettando alcuni criteri fondamentali, come utilizzare gli stili per strutturare il documento (titolo della pagina, capitoli, sezioni e sottosezioni) oppure evitare l'impaginazione a colonne (per i dettagli consultare il capitolo "**Creare file Adobe PDF accessibili**")

Prima di trasformare i moduli fac-simile dal formato word al formato PDF, è possibile sostituire il simbolo “\_” (underscore) o “.” (punto) con la linea grafica orizzontale di word;

in tal modo “lo spazio di compilazione” appare visivamente come una sequenza di underscore, rendendo meno fastidiosa la lettura da parte dello screen reader.

Figura 3 - Esempio: modulo fac-simile.

### Come procedere

- Quando è possibile, non creare tabelle per l'impaginazione.
- Se fosse necessario creare una tabella di impaginazione, semplificare quanto più possibile la struttura della tabella (una sola colonna multiriga, oppure una sola riga multicolonna).
- Se si utilizza una tabella per l'impaginazione non devono essere usati marcatori strutturali ai fini della formattazione visiva, ad esempio non usare l'elemento `<th>` (table header) con il solo scopo di ottenere che il contenuto di una cella sia visualizzato centrato e in grassetto.
- L'attributo **summary** può essere utilizzato per descrivere l'eventuale funzione della tabella nella struttura della pagina.
- Il dimensionamento delle tabelle deve essere espresso preferibilmente in **percentuale** anziché in pixel per questioni di corretta trasformazione alle diverse risoluzioni video.

### Verifica del lavoro

- Assicurarsi che il contenuto e la struttura della tabella risultino chiari anche quando la tabella stessa viene letta cella dopo cella e una riga alla volta, assicurarsi cioè che **la tabella risulti chiara anche quando è linearizzata**.
- Si consigliano alcuni strumenti in grado di verificare l'accessibilità di una tabella di dati:
  - Internet Explorer: [Web Accessibility Toolbar](#) -> Document Structure
    - -> Table borders
    - -> Linearize (remove Tables)
  - **Lynx** (copyrighted by the University of Kansas, distribuito gratuitamente da the GNU General Public License, si veda: <http://lynx.browser.org>), browser testuale, legge le tabelle in modo linearizzato, pertanto simula la lettura da parte di uno *screen reader* (scaricare il browser in locale e procedere con la lettura della pagina contenente la tabella).
  - **Tablin** (un programma sviluppato dal WAI, si veda: <http://www.w3.org/WAI/Resources/Tablin/>), programma disponibile via web, consente allo sviluppatore di sottoporre l'URL della pagina web contenente la tabella e di ottenere l'output della tabella linearizzata.

### Approfondimenti

- *M.Diodati*: Tabelle usate a scopo d'impaginazione  
[http://www.diodati.org/scritti/2004/guida/ele\\_acc29.asp](http://www.diodati.org/scritti/2004/guida/ele_acc29.asp)
- *M.Diodati*: Come distinguere una tabella di dati da una tabella d'impaginazione.  
Linearizzazione di una tabella [http://www.diodati.org/scritti/2004/guida/ele\\_acc30.asp](http://www.diodati.org/scritti/2004/guida/ele_acc30.asp)

## Form (requisito n.14)

**Requisito n.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:

(Priorità 2)

[10.2](#) Fino a quando gli interpreti non supporteranno esplicite associazioni fra etichette e controlli dei moduli, assicurare, per tutti i controlli dei moduli che hanno etichette associate implicitamente, che l'etichetta sia posizionata correttamente.

[12.4](#) Associare esplicitamente le etichette ai loro controlli.

**Riferimenti Sec. 508:** 1194.22 (n)

### Descrizione

Un modulo HTML (**form**) è la sezione di un documento che presenta contenuto normale, ovvero marcature, elementi speciali detti controlli, o **caselle** (quadratini, pulsanti basculanti, menu, ecc.), ed etichette per tali controlli. Di solito l'utente compila un modulo modificandone i controlli (introducendo del testo, selezionando elementi di menu, ecc.), prima di passare il modulo ad un programma di elaborazione (ad esempio a un server web, a un server di posta elettronica, ecc.).

### Come procedere

- Assicurarsi che le etichette siano posizionate adeguatamente: l'etichetta deve precedere immediatamente la sua casella sulla stessa linea (permettendo più di un casella/etichetta per linea).
- Si consiglia di posizionare l'etichetta **a sinistra** rispetto alla relativa casella nel caso di campo testo, campo textarea e menu a tendina.
- Si consiglia di posizionare l'etichetta **a destra** rispetto alla relativa casella nel caso di **radio button** e **checkbox**.
- Qualora l'etichetta non sia posizionate adeguatamente rispetto alla sua casella (ad esempio quando sia utilizzata una tabella per impaginare la form, dove l'etichetta e la casella siano posizionate in celle non sequenziali), associare esplicitamente le etichette alle loro caselle tramite l'elemento `<label for>` e `<id>`.

#### Esempio 1 (versione corretta)

Cerca:	<input type="text"/>
<i>etichetta</i>	<i>casella</i>

#### Esempio 2 (versione corretta)

Scelta del numero massimo di documenti mostrati:	15 ▼
--	------

**Esempio 3** (versione corretta)

<input type="radio"/> Frase esatta <input type="radio"/> Almeno una parola <input checked="" type="radio"/> Tutte le parole
---

**Esempio 4** (versione non corretta)

Età e professione <input type="checkbox"/> <input type="text"/>
--

**Esempio 5** - utilizzo dell'elemento `<label for>` e `<id>`

**Ricerca Fulltext**

Cerca: [Istruzioni per la ricerca](#)

Frase esatta   
 Almeno una parola   
 Tutte le parole

Opzioni di Ricerca:		
Distanza massima (in n. di caratteri)		<input type="text"/>
Scelta del numero massimo di documenti mostrati:		<input type="text" value="15"/>
Espandi i Termini <input type="checkbox"/>	Cerca i sinonimi <input type="checkbox"/>	Correggi gli errori <input type="checkbox"/>

**Source del codice HTML dell'esempio n. 5 (si riporta solo il codice relativo ai campi della form)**

```

. . .
<label for="cerca">Cerca:</label>
. . .
<input type="text" id="cerca" value="" SIZEsize="30" maxlenght="255">
. . .
Opzioni di Ricerca:
. . .
<label for="proximity">Distanza massima (in n. di caratteri)</label>
. . .
<select name="proximity" id="proximity">
<option>nessuna</option>
<option value="20" >20</option>
<option value="50" >50</option>
<option value="100" >100</option>
</select>
. . .
<label for="soglia">Scelta del numero massimo di documenti mostrati:</label>
. . .
<select name="soglia" id="soglia">
<option value="15" >15</option>
<option value="25" >25</option>
<option value="50" >50</option>

```

```

<option value="100" >100</option>
<option value="200" >200</option>
</select>
. . .
<label for="espandi">Espandi i Termini</label>
<input type="CHECKBOX" name="espansione_termini" id="espandi" value="1">
. . .
<label for="sinonimi">Cerca i sinonimi</label>
. . .
<input type="CHECKBOX" name="sinonimi" id="sinonimi" value="1" >
. . .
<label for="errori">Correggi gli errori
<input type="CHECKBOX" name="errori" id="errori" value="1" >
. . .

```

## Verifica del lavoro

Effettuare la seguente verifica manuale:

- Internet Explorer: [Web Accessibility Toolbar](#) -> fieldset/label (form)

## Approfondimenti

- *Webaccessibile.org*: Come rendere più accessibili i Moduli di interrogazione - <http://www.webaccessibile.org/argomenti/argomento.asp?cat=295>
- *WebAIM*: How to Create Accessible Forms - <http://www.webaim.org/techniques/forms/> Per moduli più complessi, potrebbero essere necessari ulteriori tag, connessi con l'accessibilità, come <legend> e <fieldset>. Questa guida mostra cosa sono e come usarli.
- *W3C HTML 4.01 Specification*: Forms in HTML Documents: The LABEL element <http://www.w3.org/TR/REC-html40/interact/h-17.9.1>

## Script, applet o altri oggetti di programmazione (requisito n.15, n.16, n.17 e n.20)

### Requisito n.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.

### Requisito n.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.

### Requisito n. 17

**Enunciato:** 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.

### Requisito n.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:

#### (Priorità 1)

[6.3](#) Assicurarsi 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.

#### (Priorità 2)

[6.4](#) Per quanto riguarda script e applet, assicurarsi che i gestori di eventi siano indipendenti dai dispositivi di input.

[6.5](#) Assicurarsi che il contenuto dinamico sia accessibile oppure fornire una presentazione o pagina alternativa.

[7.4](#) Fino a quando gli interpreti non forniranno la possibilità di bloccare l'autoaggiornamento, non creare pagine che si autoaggiornano periodicamente. Ad esempio, in HTML, non fare autoaggiornare le pagine con "HTTP-EQUIV=refresh" fino a quando gli interpreti non permetteranno agli utenti di disabilitare questa caratteristica.

[7.5](#) Fino a quando gli interpreti non forniranno la capacità di bloccare l'auto-reindirizzamento, non usare marcatura per reindirizzare le pagine automaticamente. Piuttosto, configurare il server in modo che esegua i reindirizzamenti.

[8.1](#) Fare in modo che elementi di programmi come script e applet siano direttamente accessibili o compatibili con le tecnologie assistive [Priorità 1 se la funzionalità è importante e non presentata altrove, altrimenti Priorità 2.]

[9.2](#) Assicurarsi che ogni elemento che possiede una sua specifica interfaccia possa essere gestito in una modalità indipendente da dispositivo.

[9.3](#) Negli script, specificare gestori di evento logici piuttosto che gestori di evento dipendenti da dispositivo.

**Riferimenti Sec. 508:** 1194.22 (l),1194.22 (m)

## Javascript, DHTML

**Javascript** permette agli sviluppatori di implementare dei piccoli programmi (gli **script**) all'interno del codice (X)HTML in grado di generare effetti dinamici, svolgere azioni semplici come modificare un elemento grafico quando il puntatore del mouse vi passa sopra, eseguire calcoli matematici, controllare la coerenza delle informazioni inserite, ecc...

Javascript è stato sviluppato da Netscape Corporation e, pur avendo una sintassi simile a Java, è un linguaggio di scripting, quindi non necessita della compilazione, ma viene direttamente interpretato dal browser. Per poter funzionare, il codice Javascript deve essere eseguito all'interno di un browser in grado di interpretarlo: quando il browser trova del codice Javascript all'interno di un documento (X)HTML, lo esegue e ne visualizza gli eventuali risultati sulla pagina web. Il computer che legge il codice Javascript deve possedere ed avere attivato un interprete Javascript, in grado di comprendere le istruzioni e di eseguirle.

**DHTML** è una sigla (Dynamic HTML) utilizzata dai fornitori di user agent per indicare l'interazione tra i linguaggi di scripting, il documento caricato in memoria (**DOM**, ovvero Document Object Model) e il foglio di stile CSS. La compatibilità degli effetti dinamici ottenuti con DHTML sui vari user agent, dipende da come i fornitori hanno implementato le indicazioni del W3C riguardo a DOM, CSS e Javascript.

Per ottenere una compatibilità della pagina (X)HTML **cross-platform**, lo sviluppatore dovrà pertanto tener conto delle suddette problematiche.

### Regole generali

- Assicurarsi che i contenuti e le funzionalità forniti dagli script siano garantiti, anche in modo alternativo, in caso il browser o user agent abbia gli script disabilitati
- Assicurarsi che le funzionalità degli script siano accessibili anche dalla tastiera: non sono casi rari quelli di persone non in grado, anche temporaneamente, di sfruttare un mouse per navigare.
- Fornire un contenuto alternativo.

### Come procedere

- Le funzionalità Javascript che realizzano effetti "visuali" che possono essere realizzati con i fogli CSS devono essere sostituito con quelli via CSS.
- Alcune funzionalità Javascript, utili per effetti "visuali", non richiedono tecniche per favorire l'accessibilità, perché considerate non essenziali all'esecuzione di un'applicazione, ma utili al più per l'usabilità dell'applicazione. Ad esempio, gli effetti "mouse-over" per evidenziare un elemento quando il puntatore del mouse vi passa sopra, non sono interpretati dallo screen reader e trascurabili per chi utilizza una tecnologia assistiva, non in grado di visualizzare tali effetti, ma risultano utili in generale per l'usabilità di un'applicazione.
- Non utilizzare un codice Javascript che richieda solo l'uso del mouse. In particolare i gestori di eventi, ovvero le funzionalità dello script che generano un evento ogni qual volta l'utente fa qualcosa (click con il mouse, sposta o trascina il puntatore e così via) dovrebbero avere specificati entrambi gli eventi, ad esempio usare "onmousedown" con "onkeydown", "onclick" con "onkeypress", "onmouseup" con "onkeyup", ecc...

Ad esempio:

```
<a href="esempio.html" onclick="MM_openBrWindow('esempio.html',
'nuovaFinestra','width=400,height=400'); return false;"
onkeypress="MM_openBrWindow('esempio.html',
'nuovaFinestra','width=400,height=400'); return false;" title="Esempio">Apri
l'esempio</a>
```

- Fornire alternative: non tutti gli utenti hanno un browser con Javascript attivato, per cui è indispensabile fornire qualche tipo di alternativa. La maniera più semplice è inserire del contenuto all'interno dell'elemento **<noscript>**; non è sufficiente dire "il tuo browser non supporta Javascript". Si dovrebbe, invece, fornire esattamente lo stesso tipo di contenuto e di funzionalità anche per gli utenti non provvisti di interprete Javascript. Si ricorda che a seconda della versione dell'(X)HTML scelto il codice Javascript all'interno del tag `<script>` deve essere opportunamente trattato. Ad esempio:

```

<script type="text/javascript">
function openwindow()
{
.....
.....
}
</script>
<noscript>
Versione alternativa per utenti che hanno JavaScript disabilitato
<a href="news.htm" target="_blank">News</a> (il link si apre in una nuova
pagina del browser)
</noscript>

```

- Se si utilizzano funzioni Javascript all'interno di collegamenti ipertestuali, la pratica corretta è quella di fornire una destinazione "reale" al collegamento e gestire gli eventi, come onclick e onkeypress.

Ad esempio il link:

```
<a href="http://www.cineca.it/" onClick="apri(this); return false;">
```

permette di accedere alla risorsa contenuta nell'<href anche nel caso in cui il Javascript sia disabilitato, mentre il link:

```
<a href="javascript:apri('http://www.cineca.it/')">
```

non risulta accessibile perché il Javascript potrebbe essere disabilitato o non supportato.

- Laddove possibile, utilizzare la tecnica di scripting lato server, piuttosto che quella lato client.
- Per pagine che si aggiornano automaticamente o con risposta temporizzata, sarebbe meglio trovare una soluzione che elimini tale temporizzazione. Se comunque è necessario mantenerla allora è necessario fornire una seconda copia della pagina nella quale il **refresh** avviene solo dopo la selezione di un link. Sarebbe meglio evitare di creare pagine che si autoaggiornano periodicamente: ad esempio, in (X)HTML, non bisognerebbe fare autoaggiornare le pagine con l'attributo "HTTP-EQUIV=refresh" o usare i marcatori per reindirizzare le pagine automaticamente. È preferibile, in ogni caso, configurare il server in modo che esegua i reindirizzamenti e fornire una pagina alternativa in cui tale aggiornamento non avvenga automaticamente, ma solo mediante la selezione di un link.

## Approfondimenti

- *Trace Center: JavaScript and Dynamic HTML Accessibility*  
<http://www.trace.wisc.edu/world/java/jsript.htm>
- W3C: Document Object Model FAQ - <http://www.w3.org/DOM/faq.html>.

## Object e Applet

L'elemento **<object>** è - secondo le indicazioni del W3C - l'elemento corretto per inserire elementi multimediali in un documento (X)HTML, tanto che nelle specifiche dell'XHTML 2 (l'evoluzione di XHTML 1.0) persino le immagini devono essere inserite tramite questo elemento.

In molti casi l'elemento **<object>** si occupa di attivare un "**plug-in**", cioè un componente aggiuntivo che si integra nel browser, per lo più fornito dal produttore del software multimediale (ad esempio Macromedia Flash o Sun Java Applet), in grado di leggere il file multimediale (qualunque esso sia).

Un'**applet** è un'applicazione Java che viene eseguita **lato client** (sulla macchina utente) e interagisce con il browser web. Le applet Java sono dei veri e propri programmi (scritti in linguaggio Java della Sun Microsystems), che permettono di produrre effetti dinamici o caratteristiche attive su una pagina.

L'elemento **<object>** deve essere elaborato e pertanto compreso dal browser dell'utilizzatore. Poiché le azioni richiamate attraverso l'elemento **<object>** non sempre vengono riprodotte dalle interfacce di assistenza e persino da alcuni user agent, è necessario aggiungere sempre un modo alternativo per veicolare o indicare il contenuto all'interno del codice della pagina. È bene specificare che l'applet può essere riferita all'interno della pagina anche attraverso uno specifico elemento che si chiama **<applet>**, tuttavia l'elemento è *disapprovato* ("deprecated", in inglese) dal W3C in HTML 4.01 e non è conforme alla DTD Strict di XHTML 1.0.

### Regola generale

Fornire una modalità alternativa e assicurarsi che le pagine siano utilizzabili anche quando applet e object sono disabilitati oppure non supportati. Se questo non è possibile, fornire informazione equivalente in una pagina accessibile alternativa.

### Come procedere

- Quando si utilizza l'elemento **<object>**, fornire un equivalente in (X)HTML nel contenuto dell'elemento. In questo modo quando l'elemento **<object>** non è supportato dallo **user agent**, viene comunque visualizzato il contenuto in (X)HTML. Ad esempio:

```
<object classid="java:gravity.class" width="200" height="250"> Quando la gravità agisce su un oggetto, il peso... </object>
```

- Quando si utilizza l'elemento **<applet>**, fornire testo alternativo con l'attributo **alt**, all'interno del contenuto dell'applet. Ad esempio:

```
<applet code="gravita.class" width="200" height="250" alt="Calcolatore di gravità"> . . . </applet>
```

In ogni caso, quando l'applet non è accessibile o non è supportata, fornire una modalità alternativa. Ad esempio:

```
<applet code="spin-globe.class" width="200" height="250" alt="Rotazione del Pianeta Terra"> Questa applet mostra la rotazione del Pianeta Terra ...</applet>
```

*oppure*

```
<applet code="spin-globe.class" alt="Rotazione del Pianeta Terra">...</applet><p>Se questa applet non è supportata dal tuo browser,
```

```
vai alla seguente pagina che propone una descrizione testuale<a
href="spin-globe.html"> Descrizione della rotazione del Pianeta Terra
</a>.</p>
```

- Le pagine Web non dovrebbero contenere elementi in movimento. Quando una pagina include contenuto in movimento, bisogna fornire un meccanismo all'interno di uno script o di una applet per permettere agli utenti di bloccare il movimento o gli aggiornamenti. Se si usano i fogli di stile insieme con gli script per creare il movimento, gli utenti potranno disabilitare oppure tenere sotto controllo gli effetti con maggiore facilità.

## Approfondimenti

- Desktop Java Accessibility - <http://java.sun.com/products/jfc/accessibility/index.jsp>
- *IBM Guidelines*: "IBM Guidelines for Writing Accessible Applications Using 100% Pure Java." <http://www-306.ibm.com/able/guidelines/java/accessjava.html>
- Java Accessibility and Usability Work <http://trace.wisc.edu/world/java/java.htm>

## Verifica del lavoro

Verificare che i contenuti e le funzionalità della pagina siano ancora fruibili, anche in modalità diverse, in caso di disattivazione di script ed altri oggetti di programmazione e che la pagina sia navigabile con il solo uso della tastiera e l'impiego di una normale abilità.

- Qualora la pagina web presenti javascript o Active X, effettuare una verifica manuale provando a disabilitare **javascript e Active X** su alcuni dei seguenti browser:
  - Internet Explorer: Web accessibility toolbar -> IE options -> toggle javascript, toggle Active X oppure Tools -> Internet options -> Security -> Internet -> Custom Level -> Scripting
  - Netscape: Edit > Preferences > Advanced > Script & Plug-ins
  - Mozilla Firefox: view -> Tools > Options > Web features > Enable Javascript
  - Opera 6: file > Preferences > Multimedia -> Enable Javascript
  - Opera 7 e 8: Tools > Preferences > Advanced -> Enable Javascript
- Qualora la pagina web presenti applet Java, effettuare una verifica manuale provando a disabilitare **Java** su alcuni dei seguenti browser:
  - Internet Explorer: Tools -> Internet options -> Security -> Internet -> Custom Level -> Scripting.
  - Netscape: Edit > Preferences > Advanced > Script & Plug-ins
  - Mozilla Firefox: view -> Tools > Options > Web features > Enable Java
  - Opera 6: file > Preferences > Multimedia -> Enable Java
  - Opera 7 e 8: Tools > Preferences > Advanced -> Enable Java
- La verifica dei plug-ins va effettuata se sono presenti file **Flash o altri oggetti di programmazione** che richiedano particolari interpreti. Effettuare una verifica manuale provando a disabilitare i plug-ins su alcuni dei seguenti browser:
  - Internet Explorer: Web accessibility toolbar -> tools -> simulations -> disable plugins oppure Tools -> Internet options -> Security -> Internet -> Custom Level -> Scripting.
  - Netscape: Edit > Preferences > Advanced > Script & Plug-ins
  - Mozilla Firefox: view -> Tools > Options > Web features > Enable Plug-ins
  - Opera 6: file > Preferences > Multimedia -> Enable Plug-ins

- Opera 7 e 8: Tools > Preferences > Advanced -> Enable Plug-ins

## Plug-in, documenti non HTML

Spesso sorge l'esigenza di fornire contenuti web in formato proprietario che non vengono visualizzati direttamente dal browser, ma da opportune applicazioni aggiuntive (**plug-in**). Si può trattare di documenti in formato immagine TIFF, di contenuti multimediali (Real Player, Windows Media Player), animazioni Flash o Shockwave, PDF, ma anche di documenti Word, Excel, WordPerfect, ecc.

### Come procedere

- Fornire il link da cui scaricare l'eventuale plug-in, se possibile quello della versione in lingua del sito o della parte di sito interessata. A riguardo, bisogna tenere presente che di alcuni prodotti potrebbero esistere specifiche versioni accessibili differenziate da quella standard, come è il caso di Adobe Acrobat Reader 5.05.
- Assicurarsi che gli stessi plug-in siano a loro volta accessibili. Si tenga pertanto presente che i pacchetti della suite Office e in generale i prodotti più diffusi lo sono.
- Si tenga in considerazione che certi formati non sono supportati in tutte le piattaforme; laddove possibile, preferire formati non proprietari (ad esempio RTF).
- È comunque vivamente consigliato, obbligatorio se si ha il sospetto o la certezza che non sia fruibile da tutti, fornire una versione in (X)HTML del documento in modo da garantirne la leggibilità nell'ambito del browser.
- I plug-in con contenuto interattivo devono essere attivati e controllati anche dalla tastiera e non solo con l'utilizzo del mouse.

### Esempio

Alcuni siti web hanno pagine che presentano un elenco di documenti (ad esempio la rassegna stampa), disponibili soltanto in formato TIFF. Tale formato è fruibile unicamente dagli utenti provvisti di un browser grafico (Internet Explorer, Netscape, ...) e del plug-in opportuno.

Il formato TIFF non è accessibile ad utenti che utilizzano altri navigatori web (ad esempio gli screen reader).

Una soluzione per superare l'attuale limitazione consiste nel prevedere, per ciascun documento con estensione TIFF, una pagina (X)HTML contenente solo il testo. In tal modo, ciascun documento sarebbe così disponibile in formato TIFF e in formato (X)HTML.

Quindi, il link al documento presente nella pagina di un sito si presenterebbe come segue:

"Titolo del documento" [ <a href="#">formato TIFF</a> KB ...] [ <a href="#">formato (X)HTML</a> ]
---

anziche'

" <a href="#">Titolo del documento</a> ", dove il link punta direttamente al formato TIFF.
--

Inoltre i file TIFF sono solitamente pesanti, pertanto è consigliato indicare a fianco del link il peso in KByte.

## Flash MX

Flash Mx permette di rendere accessibili i contenuti dei filmati. Il modello tecnologico su cui si basa Flash MX è quello di **Microsoft Active Accessibility** supportato da Internet Explorer. Active Accessibility permette agli sviluppatori di software di creare programmi maggiormente compatibili con i supporti per l'accesso facilitato come gli screen reader, mentre gli sviluppatori di tali supporti possono creare strumenti più affidabili e sofisticati.

Attualmente è possibile sviluppare siti accessibili in Flash in ambiente Mac, ma non è possibile testarli con lo screen reader perché al momento non esiste un supporto simile al modello Microsoft per sistemi operativi Mac OS. La possibilità di interpretare i contenuti accessibili dei filmati Flash parte dalla versione 4.50 dello screen reader Jaws.

Si possono applicare due tipi di accessibilità ad un filmato:

- l'**accessibilità di default** fornisce la descrizione della scena nel suo insieme che, se viene abilitata per l'intero file, permette agli screen readers di leggere tutte le aree di testo contenute nel file, facendo in modo che i testi inseriti nei pulsanti o posizionati vicino ad una loro istanza vengano associati a questi.
- l'**accessibilità personalizzata**, ovvero definibile per ogni elemento grafico del nostro file, come ad esempio per ogni clip filmato, pulsante e grafico.

Queste funzioni si applicano tramite i due tipi di pannello **Accessibilità** in Flash, che si aprono dal menu **Finestra** oppure cliccando sull'iconcina del **Pannello proprietà**.

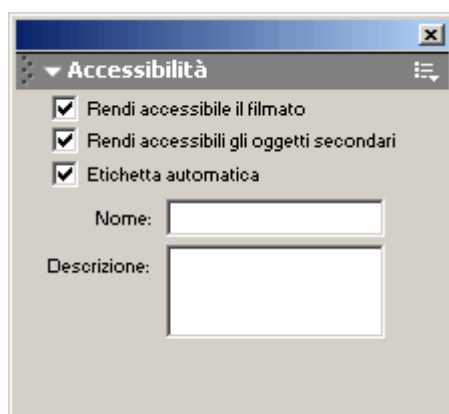
È possibile rendere accessibile la parte testuale di un filmato Flash utilizzando il testo in modalità dinamica (Figura 1): questo tipo di testo viene letto dallo screen reader in qualunque livello esso si trovi, al contrario del testo statico e di quello di input che invece viene ignorato.



Tramite invece il pannello di Figura 2, che si apre con lo stage selezionato, l'opzione **Rendi accessibile il filmato** (Make Movie Accessibile) permette di leggere tutte le aree di testo.

È inoltre possibile rendere accessibile il contenuto di un intero filmato assegnandogli un nome e fornendo una descrizione quanto più dettagliata possibile, in modo da rendere efficiente la lettura degli screen readers. Si può attivare l'opzione etichetta automatica (auto label), che fornisce

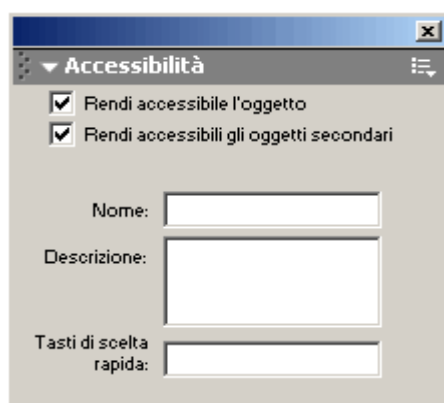
istruzioni al Flash Player così da associare ad un simbolo i campi di testo che lo toccano o che sono vicini ad esso.



(fig.2)

Per descrivere un oggetto del filmato (grafico, clip filmato o bottone), occorre selezionare l'oggetto e aprire il pannello accessibilità mostrato nella Figura 3 per darne descrizione tramite l'operazione di riempimento dei campi, ossia associando ad ogni simbolo sullo stage un nome, una descrizione estesa e uno shortcut di attivazione.

Infatti per gli oggetti come **clip**, **grafici** o **bottoni** è possibile creare una **scorciatoia** da tastiera inserendo la combinazione di tasti di scelta rapida da utilizzare (Figura 3). Attualmente solo una versione beta di Windows Eyes supporta questa funzionalità che sarà comunque presto introdotta negli screen readers più diffusi.



(fig.3)

È importante verificare che l'utente sia in grado di navigare il sito anche solo usando i comandi da tastiera come avviene in altre applicazioni.

Flash MX ha a disposizione anche alcuni comandi di **ActionScript** per gestire al meglio i contenuti accessibili. Il metodo `Accessibility.isActive()` indica se è attualmente attivo uno screen reader. L'uso di questo metodo è consigliato se si desidera che il filmato abbia un diverso funzionamento quando viene aperto in presenza di uno screen reader. È strettamente collegato alla proprietà `System.capabilities.hasAccessibility`, che restituisce un valore booleano che indica se il dispositivo supporta la comunicazione tra Flash Player e gli strumenti di accesso.

### Consigli per gli sviluppatori

### **Animazioni**

Le animazioni sono sicuramente il **punto di forza** di Flash, ma potrebbero complicare la navigazione agli utenti disabili. È perciò importante cercare di evitare delle animazioni inutili come quelle di pulsanti, testo statico e campi di testo statico e di input nei filmati perché, ogni qual volta il Player visualizza un'animazione, lo screen reader riceve il comando di leggere nuovamente i contenuti della pagina (funzionalità che si può disattivare con lo short cut **Alt + Shift + M**).

Non bisogna inoltre utilizzare informazioni che rimangono per un tempo troppo breve sullo schermo perché lo screen reader potrebbe non fare in tempo a leggerle.

In più è necessario verificare la temporizzazione in quanto il cambio troppo rapido del video a determinate frequenze potrebbe creare problemi alle persone epilettiche.

### **Effetti sonori e file audio**

Si consiglia di evitare l'inserimento di file audio o di effetti sonori non strettamente necessari in un filmato, in quanto potrebbero interferire con lo screen reader.

### **Approfondimenti**

- *Macromedia*: Accessibility <http://www.macromedia.com/macromedia/accessibility/>

## **File Microsoft Excel**

I formati che non sono del W3C, come i file Microsoft Excel, richiedono di essere visti con applicazioni autonome.

I file Excel visualizzati con il programma **Microsoft Excel** o con il **viewer Microsoft Excel** sono **accessibili**, in quanto lo screen reader è in grado di leggerli. Mentre i file Excel visualizzati all'interno del **browser non** sono **accessibili**, poiché lo screen reader non è in grado di leggerli correttamente.

### **Come procedere**

- Assicurarsi che la tabella dei dati del foglio elettronico Excel rispetti alcuni criteri importanti di formattazione, in particolare:
  - eliminare le **righe** e/o **colonne** completamente **vuote** quando non servono;
  - **identificare** le **celle vuote** con un **carattere** (ad esempio il meno '-' ) o indicare il contenuto in ciascuna cella vuota;
  - identificare i dati mancanti nelle celle con un'abbreviazione (ad esempio "**n.p.**", non presente), quando il dato mancante esprime un valore;
  - **inserire** in fondo alla tabella una **legenda** che spieghi il significato dei caratteri o delle abbreviazioni utilizzate.
- Attribuire nomi significativi ai file Excel, in modo da consentire agli utenti che salvano i file in locale, di individuare facilmente i documenti di proprio interesse (ad esempio un file Excel che contiene i "Dati relativi al superamento degli esami presso l'Università degli Studi di Bologna, A.A. 2002/2003" potrebbe avere come nome "superamentoesami\_bo\_2002\_2003.xls").

- Indicare con una nota sul sito come procedere per l'apertura del file (salvarlo in locale e aprirlo con il programma o il viewer Microsoft Excel oppure configurare le opzioni del browser per aprire il file con il programma Microsoft Excel).

## Creare file Adobe PDF accessibili

I diversi tentativi di **accessibilità** fatti sui file Adobe PDF ci hanno portato a concludere che tali file **non sono accessibili**, a meno che il contenuto non sia costituito da **solo testo**.

Il progetto WAI del W3C non delinea nessuna linea guida precisa relativa ai documenti PDF, se non suggerire di fornire una versione alternativa del contenuto che sia accessibile. Il sito dell'Adobe (si veda <http://www.adobe.com>) contiene invece documentazione su alcuni problemi legati all'accessibilità di tali documenti.

Ad esempio, è stata rilasciata la **versione 5.05 di Adobe Acrobat Reader**, che risolve alcune questioni legate all'accessibilità del documento PDF prevedendo nuove funzioni per la lettura del file. È importante, quindi, che l'utente che usa un software di tecnologia assistiva utilizzi la versione Acrobat Reader 5.05 per la gestione dei file Adobe PDF: a questo proposito si raccomanda di inserire sul sito una nota che spieghi come scaricare tale versione e le modalità di lettura offerte dal viewer.

I vari tentativi fatti per valutare l'accessibilità dei file Adobe PDF hanno condotto alle seguenti considerazioni.

- In presenza di file Adobe PDF, convertiti da file Excel, è preferibile utilizzare direttamente la versione originale in Excel: lo screen reader non evidenzia nessun problema nella lettura di un file Excel.
- In presenza di file Adobe PDF piuttosto complessi (che presentano tabelle, grafici, testo incolonnato) è possibile affiancare al file una versione alternativa in linguaggio HTML accessibile.
- Se il file PDF deriva da un documento word, è importante che l'autore scriva il documento in formato .doc rispettando alcuni criteri fondamentali. In particolare:
  - **inserire** nel documento **aiuti di navigazione** e aiuti **organizzativi**, come ad esempio tabelle di contenuto;
  - **specificare** la **lingua** in cui il documento è scritto;
  - **utilizzare** gli **stili** per strutturare il documento: **titolo** della pagina, **capitoli**, **sezioni** e **sottosezioni**. Ciò facilita la navigazione del documento, soprattutto se è troppo lungo. Inoltre, la presenza di una struttura logica è indice di documenti accessibili;
  - **creare** i documenti in **formato .doc** in Word 2000 piuttosto che in Word '97; i documenti realizzati in Word '97 possono essere successivamente aperti e salvati in Word 2000, formato che permette di realizzare documenti Adobe PDF etichettati, che offrono un maggior livello di accessibilità;
  - **non utilizzare** il tasto **enter** per creare spazi tra paragrafi, ma le proprietà di Word, cioè: dopo aver evidenziato i paragrafi da separare l'autore deve selezionare **Formato>Paragrafo**; nella sezione **Spaziatura** immettere la spaziatura desiderata nella casella prima o dopo;

- per creare **testo a due o più colonne** utilizzare i comandi di Word: selezionare **Formato>Colonne** e scegliere il numero di colonne volute; non utilizzare il tasto **Tab** per creare lo stesso effetto;
- per creare tabelle, utilizzare il comando **Inserisci Tabella** o il comando **Disegna tabella** dal menu **Tabella**. Evitare di inserire tabelle già presenti in altri documenti;
- per inserire un'immagine, un grafico o un oggetto, utilizzare il comando **Immagine** dal menu **Inserisci** e selezionare la scelta che interessa. Corredare di testo alternativo tutte le immagini: posizionarsi sull'oggetto, cliccare col tasto destro del mouse, selezionare la voce **Formato immagine>web** ed inserire il testo alternativo.

Tutti questi accorgimenti permetteranno una migliore riproduzione dei contenuti.

### **Approfondimenti**

- Adobe Acrobat Accessibility Techniques <http://www.webaim.org/techniques/acrobat/>
- Make Accessible Plug-In  
<http://www.adobe.com/support/downloads/detail.jsp?hexID=88de>
- *Adobe.com*: How to Create Accessible Adobe PDF Files Booklet -  
[http://www.adobe.com/products/acrobat/access\\_booklet.html](http://www.adobe.com/products/acrobat/access_booklet.html)

## Multimedia

### (requisito n.18)

**Requisito n. 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:

##### (Priorità 1)

[1.3](#) Fino a quando gli interpreti non potranno leggere automaticamente ad alta voce l'equivalente testuale di un filmato, fornire una descrizione audio delle informazioni essenziali del filmato di una presentazione multimediale.

[1.4](#) Per ogni presentazione multimediale temporizzata (ad esempio un film o un'animazione), sincronizzare alternative equivalenti (ad esempio didascalie o descrizioni parlate del filmato) con la presentazione.

**Riferimenti Sec. 508:** 1194.22 (b)

#### Descrizione

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.

In questo modo coloro che non possono vedere o udire agevolmente il contenuto multimediale possono comunque avere accesso allo stesso messaggio.

#### Come procedere

- Per file audio **stand alone**, fornire una trascrizione testuale di tutte le parole dette o cantate e di tutti i suoni significativi.
- Per l'audio associato al video, provvedere a una trascrizione testuale (dei dialoghi e suoni necessari alla comprensione del contesto) sincronizzata con il video (sottotitoli e didascalie).
- Per le piccole animazioni come le **gif animate** fornire il testo alternativo.
- Per i filmati fornire descrizioni audio sincronizzate con il video e armonizzate con l'audio originale.

#### Approfondimenti

- NCAM: The National Center for Accessible Media - <http://ncam.wgbh.org/>
- Rich Media Home - <http://ncam.wgbh.org/richmedia/index.php> - a growing collection of resources for developers and users interested in ways to make rich media accessible to people with disabilities.
- "Creating Captions for Rich Media" - <http://ncam.wgbh.org/richmedia/tutorials/captioning.html>

## Link ipertestuali (requisito n.19)

**Requisito n.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:

(Priorità 2)

[13.1](#) Identificare con chiarezza l'obiettivo di ogni collegamento.

(Priorità 3)

[13.6](#) Raggruppare i collegamenti correlati, identificare i gruppi (per gli interpreti) e, fino a quando gli interpreti non lo fanno, fornire un modo per saltare il gruppo.

**Riferimenti Sec. 508:** 1194.22 (o)

### Descrizione

Per ciascun link ipertestuale di una pagina, scegliere un testo cliccabile significativo che mantenga il proprio senso logico anche fuori del contesto della pagina web.

### Come procedere

- Usare parole o brevi frasi di chiaro e univoco significato anche fuori del contesto, evitando espressioni generiche come "**premi qui**" o "**torna indietro**", ad esempio utilizzare enunciati tipo "**Torna all'indice analitico**".
- Si consiglia di utilizzare l'attributo **title** solo nei casi in cui è necessario descrivere più chiaramente e accuratamente l'obiettivo del link ipertestuale o per aggiungere informazione al testo utilizzato nel link ipertestuale: la descrizione estesa deve aiutare a capire il contenuto della pagina che il link apre, senza necessariamente riportare il testo del link. L'informazione presente nel title appare come un breve messaggio quando il dispositivo di puntamento si ferma sul link (cosiddetto "tooltip") e viene letta a voce dallo screen reader, quindi non conviene specificare un "title" uguale al testo del link, perchè risulta un'informazione ridondante.

Esempio:

```
<a href="/pubblicazioni/index.htm" title="Rivista istituzionale,
periodici di settore"> Pubblicazioni </a>
```

- Quando si utilizzano simboli di testo per rappresentare un link, ad esempio l'uso di ">>>" per indicare una continuazione del testo, è importante utilizzare l'attributo **title** per descriverne il significato (ad esempio l'enunciato "**continua all'interno**", oppure supponendo di trattare il testo completo di una novità del sito, ricorrere a un enunciato più esplicito, come "**News n.1 : testo completo**").

Esempio:

```
<a href="/news/news1.htm" title="News n.1: testo completo"> >>> </a>
```

- Quando il link ipertestuale apre una nuova finestra del browser, si consiglia di specificarlo nell'attributo **title** con una frase del tipo "**la pagina si apre in una nuova finestra**": la moltiplicazione delle finestre a volte può essere fonte di confusione per l'utente che usa uno screen reader, dato che lo strumento lavora in modo seriale (cioè su una sola finestra per volta).

Quando il link ipertestuale conduce al di fuori del portale, si consiglia di specificarlo nell'attributo **title** con una frase del tipo "**link a sito esterno - ...**" .

Esempio:

```
<a href="http://www.giustizia.it" title="Link a sito esterno - Ministero della Giustizia"> Ministero della Giustizia </a>
```

- Quando il link ipertestuale attiva **applicativi proprietari** (ad esempio Acrobat Reader, Excel, Viewer... ) si consiglia di darne indicazioni nell'attributo **title**.

Esempio:

```
<a href="doc.html" title="Documento in formato HTML">Il documento è disponibile in HTML</A>,
<a href="doc.pdf" title="Documento in formato PDF">PDF</a>,
<a href="doc.exe" title="File eseguibile">EXE</a>
```

- Non inserire il **title** nei link ad ancore interne, a meno che non sia necessario assegnare un valore aggiunto significativo.


## Immagine cliccabili

- Considerare l'immagine cliccabile come un link testuale, quindi utilizzare l'attributo **title** così come indicato per i link ipertestuali ricordandosi di inserire l'attributo title nel tag A HREF.

Esempio:

```
<a href="/csd/area/ecm_(form_dist).htm" title="Vai alla scheda del Progetto ECM"> 
```

## Lo strumento BS-D

Dalla "Barra di formattazione", selezionare il simbolo  per creare un link ipertestuale.



## Approfondimenti

- *Jakob Nielsen* - Using Link Titles to Help Users Predict Where They Are Going: <http://www.useit.com/alertbox/980111.html>
- Guidelines for Visualizing Links: <http://www.useit.com/alertbox/20040510.html>

## Skipping navigation

Si raccomanda di utilizzare un procedimento per permettere all'utente che utilizza uno screen reader o un *browser vocale* di saltare la barra o le barre di navigazione (**skipping navigation**) e proseguire nella lettura dei contenuti principali della pagina.

Il WAI non fornisce linee guida precise su questo argomento, ma l'uso di un meccanismo che permetta all'utente di scegliere se leggere tutto il contenuto della barra di navigazione o saltare direttamente al contenuto principale della pagina è fortemente gradito (ASPFI ha suggerito di utilizzarlo) e risulta essere ampiamente adottato dai principali siti che hanno curato l'accessibilità. Di seguito viene illustrato un procedimento per saltare la barra di navigazione.

### Procedimento 1

Inserire prima della barra di navigazione la seguente sintassi:

```
<div class="notifica">
<a id="top" name="top" href="#contenuto">vai direttamente al contenuto e
salta la barra di navigazione</a>
</div>
```

Inserire prima del contenuto principale della pagina la seguente sintassi:

```
<a name="contenuto">
```

Nel foglio di stile il div notifica è così impostato:

```
div.notifica { display: none; }
```

La dicitura “vai direttamente al contenuto e salta la barra di navigazione” viene letta dallo screen reader, ma non viene visualizzata sulla pagina web.

E' bene specificare che il procedimento 1, che utilizza la tecnica CSS del **{display: none;}** per non mostrare l'elemento ai browser grafici, non funziona perfettamente su tutti gli screen reader o browser vocali, cioè in alcuni casi l'elemento **non viene letto**.

La non completa copertura di tutti gli screen reader o browser vocali (dai quali va comunque escluso JAWS, utilizzato con le versioni di Internet Explorer 6.x e 5.x, sul quale il metodo funziona), è l'unico neo della soluzione.

Il procedimento 1 è teoricamente il più corretto, ma al fornitore del sito spetta comunque la decisione di utilizzare o meno questo metodo, ben sapendo che potrebbe non funzionare su alcuni screen reader.

### Procedimento 2

Inserire prima della barra di navigazione la seguente sintassi:

```
<a href="#news"></a>
```

Inserire prima del contenuto principale della pagina la seguente sintassi:

```
<a name="news">
```

*Nota:* saltonav.gif è una gif "trasparente" (stesso sfondo dell'intestazione del sito).

Il procedimento 2, che utilizza un'immagine (gif) trasparente con la dicitura 'salto barra di navigazione' specificata nell'attributo **alt**, quindi letta dallo *screen reader*, è un rimedio completamente funzionale, ma resta comunque un rimedio. Teoricamente non è consigliabile, ma dal punto di vista pratico offre garanzie di funzionamento da parte degli screen reader e browser vocali, pressoché totali. Ancora una volta al fornitore del sito la decisione.

## Navigazione da tastiera

(requisito n.21)

**Requisito n. 21 - Enunciato:** Rendere selezionabili e attivabili tramite comandi da tastiere 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 Sec. 508:** *non presente*

### Attributo accesskey

La funzione dell'attributo **accesskey** permette di raggiungere l'obiettivo del link in modo più veloce attraverso una combinazione di tasti (in ambiente Windows **Alt**+valore dell'**accesskey** e **Invio**, in ambiente Macintosh **Ctrl** +valore dell'**accesskey** e **Invio**).

Tale funzione viene utilizzata principalmente nei link più importanti della pagina e solo quando si ritiene sia utile per favorire un accesso facilitato alle risorse del sito. L'inserimento di questo attributo non è in contrasto con la presenza dell'attributo **title**.

L'attributo **accesskey** è stato introdotto con l'HTML 4.0 ed è supportato dai browser MS Internet Explorer 4 (o superiore) e da Netscape 6 (o superiore).

#### Come procedere

- Inserire l'attributo **accesskey** all'interno di `<a href="..." accesskey="valore">`.
- Il valore dell'attributo **accesskey** corrisponde ad un tasto della tastiera, può assumere un valore numerico o rappresentare una lettera dell'alfabeto. I test effettuati consigliano l'uso di valori numerici, in quanto creano meno problemi nelle diverse applicazioni.
- È possibile inserire contemporaneamente il tasto di scelta rapida **accesskey** e l'attributo **title** all'interno dell'elemento `<a href="...">`.
- Inserire sul sito nella voce "Guida alla navigazione" una sezione relativa alle Accesskey, in modo da far conoscere al navigatore lo schema per la navigazione agevolata.

Un esempio potrebbe essere:

Usa questo schema e la combinazione **Alt+numero elencato+Invio** in ambiente Windows , e la combinazione **Ctrl+numero elencato+Invio** in ambiente Macintosh :

- **1:** Home - Ti riporta nella prima pagina del sito.
- **2:** Cerca - Ti porta direttamente al modulo di ricerca.
- **3:** Mappa - Ti porta direttamente alla mappa del sito.
- ecc ...

**Esempio:**

```
<a accesskey="1" href="/index.htm">  
</a>
```

**Approfondimenti**

- *Jukka Korpela*: [Improving accessibility with accesskey in HTML forms and links](http://www.cs.tut.fi/%7Ejkorpela/forms/accesskey.html) - <http://www.cs.tut.fi/%7Ejkorpela/forms/accesskey.html>.
- *Paul Bohman*: [Access keys, IE6](http://www.webaim.org/discussion/mail_message.php?id=800) - [http://www.webaim.org/discussion/mail\\_message.php?id=800](http://www.webaim.org/discussion/mail_message.php?id=800).

**I punti di controllo del WAI****(Priorità 3)**

**9.5** Fornite scorciatoie da tastiera per i collegamenti importanti (compresi quelli nelle immagini sensibili sul lato client), per i controlli dei moduli e per i gruppi di controlli dei moduli.

## Creare una pagina alternativa (requisito n.22)

**Requisito n. 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:

(Priorità 1)

[11.4](#) Se, nonostante ogni sforzo, non si può creare una pagina accessibile, fornire un collegamento a una pagina alternativa che usi le tecnologie W3C, sia accessibile, contenga informazioni (o funzionalità) equivalenti e sia aggiornata con la stessa frequenza della pagina (originale) inaccessibile.

**Riferimenti Sec. 508:** 1194.22 (k)

Se comunque **ogni tentativo** precedentemente descritto **non produce un documento accessibile** - ovvero se non si può evitare di usare una tecnologia non W3C, o se tale tecnologia è usata in modo non accessibile - occorre **fornire un link** a una **pagina alternativa** che:

- usi le tecnologie W3C;
- sia accessibile;
- offra informazioni equivalenti;
- venga aggiornata con la stessa frequenza della pagina originale.

## Verificare il lavoro, ovvero la validazione

È molto importante alla fine del lavoro verificarne la **qualità** e la **riuscita** utilizzando strumenti di validazione automatici (disponibili gratuitamente in Internet) e visualizzando la pagina prodotta su un browser testuale.

Si consiglia di verificare l'accessibilità sia con **validatori automatici**, sia con **strumenti semiautomatici**, sia con una **revisione diretta**. I metodi automatizzati sono di solito rapidi e convenienti ma non riescono a identificare tutti i problemi dell'accessibilità. La revisione di uno sviluppatore (autore o meno del lavoro) può aiutare ad assicurare la chiarezza di linguaggio e la facilità di navigazione.

I software però non risolvono tutti i problemi di accessibilità, come ad esempio il significato pieno del testo dei collegamenti e l'applicabilità di un equivalente testuale.

Si rimanda al documento "*Metodologia di valutazione dell'Accessibilità basata sulla legge 4/2004*" Versione 1.0 - ottobre 2005

## Glossario minimo

**Accessibile** - Si dice che un documento, una pagina web o un sito sono accessibili quando non contengono “barriere telematiche”, ovvero non presentano ostacoli che impediscono la loro fruibilità da parte di tutti.

**Applet** - Un programma inserito in una pagina web.

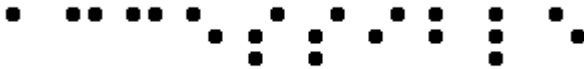
**ASCII** - (American Standard Computer Information Interchange). Codice standard usato per lo scambio di informazioni fra sistemi di elaborazione, sistemi di comunicazione e apparecchiature associate.

**Arte ASCII** - Insieme di caratteri di testo o simboli, combinati per creare un'immagine. Ad esempio :-) esprime un sorriso.

**Attributo** - Gli elementi possono avere proprietà associate, chiamate attributi, che possono avere valori (predefiniti, oppure impostati da sviluppatori o script). Le coppie attributo/valore appaiono prima del ">" finale del tag di inizio di un elemento. Può apparire un qualunque numero di coppie (legali) attributo valore, separate da spazi, nel tag di inizio di un elemento. Le coppie possono comparire in un ordine qualsiasi.

**Braille** - Il Braille è un sistema di lettura mediante i polpastrelli, per non vedenti. Utilizza sei punti in rilievo in diversa sequenza per rappresentare lettere e numeri.

La parola "accessibile" in Braille si traduce così:



**Display Braille** - È comunemente indicato come un "display Braille dinamico", solleva o abbassa sequenze di punti a comando da un dispositivo elettronico, di solito un computer. Il risultato è una linea Braille che può cambiare di momento in momento. Gli attuali display Braille hanno una dimensione che varia da una cella (sei o otto punti) fino a una linea di 80 celle; la maggior parte ha tra le dodici e le venti celle per linea.

**Browser** - Software con cui si esplora Internet e con cui si visualizzano pagine web. *To browse*, in inglese, significa “sfogliare”.

**Elemento** - Sinonimo di *Marcatore*.

**Foglio di stile** - Insieme di specifiche riguardanti la presentazione di un documento.

**Immagine sensibile** - Immagine suddivisa in più parti, ciascuna delle quali viene associata ad un'azione. Passando col mouse su una zona attiva, l'interprete calcola in quale zona si è verificato il click ed esegue l'azione ad essa corrispondente, cioè segue il collegamento associato a quella zona.

**Ingranditore di schermo** - Programma che ingrandisce i caratteri dello schermo del video permettendo di leggere ingranditi i dati del video del personal computer.

**Interprete** - Sinonimo di *user agent*.

**Letttore di schermo** - Programma che legge il contenuto dello schermo a voce alta. I lettori di schermo vengono usati principalmente da non vedenti, di solito sono in grado di leggere solo il testo stampato (scritto) e non quello disegnato.

**Linguaggio naturale** - Vengono così definiti i linguaggi umani parlati e scritti, o i linguaggi che si esprimono attraverso segni, come il linguaggio americano dei segni, il Braille...

**Marcatore** - Nel linguaggio HTML è un codice, un'istruzione che viene riportata tra il segno di maggiore e minore (ad esempio <title>).

**Multimedia** - Programmi che consentono di utilizzare contemporaneamente più canali sensoriali per trasmettere "messaggi". I migliori permettono di visionare brani animati e filmati e di riprodurre suoni in qualità hi-fi.

**Screen reader** - Programma di rilettura dello schermo attraverso sintesi vocale e/o display Braille.

**Sintesi vocale** - Tecnica che consente di sintetizzare la voce umana tramite un calcolatore e di usarla come output in sostituzione dei sistemi tradizionali quali monitor, stampanti...

**Sintetizzatore vocale** - Dispositivo in grado di riprodurre con voce artificiale qualsiasi testo memorizzato in un personal computer: può essere basato su una scheda da inserire all'interno del computer o su apparecchiature esterne collegabili attraverso la porta seriale e la parallela.

**Sviluppatore** - Colui che crea pagine web o progetta siti.

**Tabella linearizzata** - Un processo per rendere la tabella in modo lineare, cioè i contenuti delle celle si presentano come una serie di paragrafi, uno dopo l'altro. I paragrafi seguono lo stesso ordine delle celle della tabella d'origine. Le celle vanno lette di seguito e includono elementi strutturali (che creano paragrafi, titoli, liste, ecc.) in modo che la pagina conservi il senso dopo la linearizzazione.

**Tecnologia assistiva** - Software o hardware progettato specificamente come ausilio allo svolgimento di alcune attività. Nell'area dell'accessibilità del web, le più comuni tecnologie assistive basate su software includono lettori di schermo, ingranditori di schermo, sintetizzatori vocali e software di riconoscimento della voce che operano congiuntamente a browser con desktop grafico. Le tecnologie assistive di tipo hardware includono tastiere alternative e dispositivi di puntamento.

**Tecnologia W3C** - Si dice di specifiche e tecniche raccomandate e certificate dal Consorzio W3.

**Testo del collegamento** - Contenuto di un collegamento reso in maniera testuale.

**User Agent** - Software per l'accesso al contenuto web, compresi browser grafici per desktop, browser testuali, browser vocali, cellulari, lettori multimediali, plug-in e alcuni software di tecnologia assistiva usati congiuntamente a browser, ad esempio lettori di schermo, ingranditori di schermo e programmi per il riconoscimento della voce.

**WAI** - Il WAI (Web Accessibility Initiative) contiene le linee guida che dettano gli standard per l'accessibilità delle pagine web. Le linee guida del WAI sono oggi, a livello mondiale, il principale punto di riferimento per gli sviluppatori del web in tema di accessibilità.

## Bibliografia

### W3C - Web Accessibility Initiative (WAI)

<http://www.w3.org/WAI/>

### “Linee guida per l'accessibilità ai contenuti del Web 1.0” a cura del WAI

<http://www.aib.it/aib/cwai/WAI-trad.htm> (traduzione in italiano)

<http://www.w3.org/TR/WCAG10/> (originale, in inglese)

### Lista dei punti di controllo per le Linee guida per l'accessibilità ai contenuti del Web 1.0

a cura del WAI

<http://www.aib.it/aib/cwai/checkpoint-list-trad.htm> (traduzione in italiano)

<http://www.w3.org/TR/WCAG10/checkpoint-list.html> (originale, in inglese)

### HTML Techniques for Web Content Accessibility Guidelines 1.0

a cura del WAI <http://www.w3.org/TR/WCAG10-HTML-TECHS/>

### “Constructing Accessible Web Sites”

Autore: Jim Thatcher - Edizioni Glasshaus

### “Accessibilità di siti web - Problematiche reali e soluzioni tecniche”

a cura di Laura Burzagli e Paolo Graziani

CNR - IROE Istituto di Ricerca sulle Onde Elettromagnetiche

<http://www.ifac.cnr.it/smid/accesso/accesso.htm>

### HTML.it/ guida introduttiva all'accessibilità a cura di Angela Molteni

<http://www.html.it/accessibilita/>

### “Criteri e strumenti per migliorare l'accessibilità dei siti web e delle applicazioni informatiche a persone disabili”

Circolare AIPA del 6 settembre 2001, n. AIPA/CR/32

[http://www.cnipa.gov.it/site/contentfiles/00127800/127899\\_CR%2032\\_2001.pdf](http://www.cnipa.gov.it/site/contentfiles/00127800/127899_CR%2032_2001.pdf)

### “eEurope 2000: accessibilità e contenuto dei siti Internet delle amministrazioni pubbliche”

Comunicazione della Commissione Europea del 28 settembre 2001

<http://register.consilium.eu.int/pdf/it/01/st12/12265i1.pdf>

### W3C HyperText Markup Language (HTML)

<http://www.w3.org/MarkUp/>

### Cascading Style Sheets

<http://www.w3.org/Style/CSS/>

## Sul web

Ecco, organizzato per temi, un elenco di interessanti riferimenti per la progettazione di pagine web: normativa italiana ed europea, standard internazionali di riferimento, guide in italiano e in inglese, oltre ad alcune risorse utili.

### Standard W3C per l'accessibilità

#### **WAI (Web Accessibility Initiative)**

[Web Accessibility Initiative \(WAI\)](#)

#### **Linee guida per l'accessibilità ai contenuti del web**

Raccomandazione del W3C del 5 maggio 1999.

<http://www.w3.org/TR/WCAG10/> (*originale, in inglese*)

<http://www.aib.it/aib/cwai/WAI-trad.htm> (*traduzione in italiano*)

#### **Web Content Accessibility Guidelines 2.0**

W3C Working Draft 23 November 2005

<http://www.w3.org/TR/WCAG20/>

#### **“Quick Tips”**

Breve promemoria e introduzione ai concetti chiave sull'accessibilità web.

<http://www.w3.org/WAI/References/QuickTips/qt.it.htm>

#### **Lista dei punti di controllo per le Linee guida per l'accessibilità ai contenuti del web 1.0**

<http://www.w3.org/TR/WCAG10/checkpoint-list.html>

(*originale, in inglese*)

<http://www.aib.it/aib/cwai/checkpoint-list-trad.htm>

(*traduzione in italiano*)

#### **“Techniques for Web Content Accessibility Guidelines”**

<http://www.w3.org/TR/WAI-WEBCONTENT/wai-pageauth-tech.html>

#### **Web Content Accessibility Guidelines 2.0**

W3C Working Draft 29 April 2003.

<http://www.w3.org/TR/WCAG20/>

#### **XHTML - W3C site**

<http://www.w3.org/xhtml/>

Riferimento ufficiale del W3C

## Guide

*In italiano*

**Diodati.org: Siti ad elevata accessibilità**

<http://www.diodati.org/scritti/2004/guida/index.asp>

**L'accessibilità dei siti web - a cura di Pubbliaccesso.gov.it**

[http://www.pubbliaccesso.it/biblioteca/manualistica/accessibilita\\_siti/index.htm](http://www.pubbliaccesso.it/biblioteca/manualistica/accessibilita_siti/index.htm)

**Bazzmann.Com - Guide - Accessibilità**

<http://www.bazzmann.com/accessibilita/guide/index.php>

**CNR - IROE Istituto di Ricerca sulle Onde Elettromagnetiche - Accessibilità di siti web**

Problematiche reali e soluzioni tecniche, a cura di Laura Burzagli e Paolo Graziani.

<http://etabeta.iroe.fi.cnr.it/accesso/accesso.htm>

**HTML.it/accessibilità**

Guida introduttiva all'accessibilità curata da Angela Molteni.

<http://www.html.it/accessibilita/>

**Html.it - Guida XHTML in italiano**

<http://www.html.it/xhtml/>

**Accessibilità delle risorse web**

<http://www.ecn.org/xs2web/guida.htm>

**Adobe**

Descrizione delle modalità per rendere accessibili i file in PDF.

<http://access.adobe.com>

*In inglese*

**Guidelines for Building an Accessible Web Site**

<http://www.ispn.gcal.ac.uk/accesites/AccessGuide.html>

**Accessibility and Information Society**

*a cura della Commissione Europea.*

[http://europa.eu.int/information\\_society/topics/citizens/accessibility/index\\_en.htm](http://europa.eu.int/information_society/topics/citizens/accessibility/index_en.htm)

**Zvon.org - XHTML 1.0 reference**

<http://www.zvon.org/xxl/xhtmlReference/Output/index.html>

Una 'reference' molto utile per chi deve scrivere pagine xhtml , perche' contiene la lista degli elementi e attributi suddivisi per DTD Strict , Transitional, Frameset.

**W3Schools Online Web Tutorials**

<http://www.w3schools.com/xhtml/default.asp>

**XHTML.ORG**

<http://www.xhtml.org>

Risorse selezionate su XHTML

**Altre risorse**

*In italiano*

**Normativa italiana** su accessibilità e usabilità

Sito Pubbliaccesso a cura del CNIPA

<http://www.pubbliaccesso.it>

**Webaccessibile.org**

Ultime novità e ultimi interventi pubblicati dai collaboratori di webaccessibile.org, eventi che riguardano l'accessibilità e la promozione delle raccomandazioni del W3C.

<http://www.webaccessibile.org>

**WebUsabile**

<http://www.webusabile.it>

**Stradeblu.it**

Sito che affronta alcune questioni riguardanti la progettazione e la realizzazione di pagine web che rispondano a determinate caratteristiche, in particolare i concetti di usabilità e accessibilità dei siti web.

<http://www.stradeblu.it>

**xs2web**

Contiene una lista completa di link interessanti a livello internazionale dei documenti utili sul tema del web accessibile.

<http://www.ecn.org/xs2web/>

**Fucinaweb**

<http://www.fucinaweb.com/>

*In inglese*

**Web Standards Project (WaSP)**

<http://www.webstandards.org/>

**Il sito di Jeffrey Zeldman** (*guru del web design*)

<http://www.zeldman.com>

**IBM Accessibility guidelines**

<http://http://www.ibm.com/able>