

Università di Bologna
Corso di Laurea Specialistica in Scienze di Internet
A.A. 2007 – 2008

Relazione del corso di Tecnologie Multimodali

**Validatori e Validazione di Tabelle HTML secondo i
requisiti di WCAG e Legge Stanca**

Cristina Donati
Stefania Gabrielli

Introduzione

Lo scopo di questo documento è quello di sintetizzare i risultati di un'analisi svolta su validatori web dell'accessibilità e raccontare il percorso che ha portato alla redazione di un proprio validatore specifico per eseguire l'analisi sulle tabelle.

Nel corso del nostro operato abbiamo, ricercato validatori che già si occupassero del nostro problema, come vedrete nel prossimo capitolo, selezionato due di questi ed eseguito una serie di analisi.

Dopo la fase di testing dell'esistente ha preso forma la parte di progettazione e implementazione del nostro validatore, di cui vi parliamo al capitolo 2.

1. Analisi dell'esistente

1.1 Ricercando Validatori Esistenti

Dovendo scegliere due validatori specifici da analizzare rivolti alle tabelle, e dovendo seguire il comportamento di *Achecker* nella totalità delle sue funzioni, per prima cosa abbiamo riassunto in forma tabellare i 22 requisiti della legge Stanca, comparandoli ai punti delle WCAG cui il regolamento attuativo fa direttamente riferimento. Risulta in questo modo più semplice appurare se un validatore segue un determinato requisito, essendo la legge Stanca una prerogativa italiana, mentre i validatori solitamente si occupano di soddisfare le WCAG.

La tabella di comparazione si trova a fine documento come Allegato A.

Partendo dalla lista dei validatori presente sul sito del W3C, <http://www.w3.org/WAI/ER/tools/complete>, abbiamo cercato di scartare quelli rivolti alle tabelle e altri strumenti utili al fine della valutazione. La scelta è stata ristretta ai seguenti:

- [Acc - an Accessibility Evaluator](#): plugin per Firefox in grado di valutare tabelle di layout annidate, fornire un link al contenuto principale, suddividere la pagina per aree di layout, valutare la consistenza di una pagina in comparazione alla precedente ed effettuare test base sulla scalabilità;
- [com tab](#): plugin per Firefox in grado di identificare gli header di tabelle annidate complesse e di fornire una versione linearizzata della tabella potendo scegliere e identificare selezionando graficamente i collegamenti fra campi;
- [EvalAccess](#): validatore via web di pagine singole, gruppi di pagine, siti interi. Prende in considerazione tutti e tre i livelli di priorità delle WCAG fornendo un report suddiviso in livelli di priorità ed indicando quali sono le linee di codice relative all'errore e quale punto delle WCAG non è rispettato;
- [Hera](#): validatore via web in php, multi lingua, valida pagine singole rispetto a tutti e tre i livelli di priorità delle WCAG;

- [Table Inspector Firefox Extension](#): validatore via web, valida pezzi di codice o pagine html prendendo le tabelle e identificandole in data e associando gli headers;
- [Tablin](#): validatore di tabelle del W3C via web in java, che restituisce l'output linearizzato della tabella;
- [Torquemada](#): validatore via web dell'accessibilità secondo le WCAG, fornisce in output la pagina testata in visualizzazione grafica e codice, e sopra include un report di quali errori sono stati riscontrati in quali linee di codice.

Atri plugin/validatori utili come supporto all'analisi su tabelle:

- [Fangs: The Firefox Screen Reader Emulator](#): plugin per Firefox che data una pagina HTML restituisce in forma testuale l'output generato da uno screen reader.

Sulla base di quanto sopra, abbiamo appurato che non era possibile scegliere uno strumento rivolto alle tabelle senza prima testarli tutti, tramite una pagina di prova appositamente scritta violando i criteri delle WCAG a differenti livelli di priorità. Il codice sorgente è in calce come Allegato B. La nostra ricerca era rivolta a scoprire validatori che non solo riportino gli errori e i punti violati, ma che magari forniscano anche possibili soluzioni. Scoprire come altri prima di noi hanno valutato il problema tabelle, trattato dalla legge Stanca ai requisiti 9,10 e 13 e corrispondente ai checkpoint 5.1, 5.2, 5.3, 5.4, 5.5, 5.6 delle WCAG 1.0.

Acc non si è rivelato molto utile. Testato sulla nostra pagina di prova ha riportato semplicemente un warning relativo alla mancata definizione della lingua principale del documento. Nessun riferimento ai punti delle WCAG, nessun warning relativo a header e summary mancanti e altro ancora.

Com Tab è utile nel cercare di creare uno strumento di supporto per la linearizzazione delle tabelle e per la correzione del codice. Fornisce graficamente la possibilità di associare i campi dati della tabella ai rispettivi header. Non attua però una vera e propria validazione conforme alle WCAG, si limita a valutare campi dati e campi intestazione, collegarli e leggere gli header.

EvalAccess è un validatore completo. Con il nostro file di prova ovviamente si è limitato a fornire errori relativi alla tabella e allo stile incorporato e non legato a un css. E' possibile dargli in ingresso l'url di una pagina singola, un intero gruppo di pagine (scegliendo il livello di profondità a partire dal path e il numero di pagine da considerare, al massimo ne valuta 15 nello stesso momento) oppure direttamente un pezzo di codice sorgente. Offre anche la possibilità di scelta di quali livelli di priorità valutare, possiamo ad esempio escludere tutti i check point del livello di priorità 3 delle WCAG.

Hera è un validatore via web in php che valuta globalmente una pagina in riferimento a tutti i check point delle WCAG. Il codice sorgente è direttamente scaricabile dal sito, e l'interfaccia è tradotta in diverse lingue, fra cui l'italiano oltre all'ovvio inglese. Prende in input una singola pagina web, e fornisce un riepilogo tabellare da cui è possibile accedere ai requisiti soddisfatti, quelli violati e quelli da controllare manualmente relativi ad ogni livello di priorità. E' anche

possibile esportare il report in formato XHTML, PDF, RDF. Con il nostro codice di prova ci ha restituito l'elenco di tutti gli errori relativi alle tabelle e anche quelli generici.

Table Inspector Firefox Extension in realtà non è un'estensione di Firefox, ma funziona tramite una pagina web. Dato un url restituisce una validazione in cui in realtà vengono identificate le tabelle come data ed evidenziati gli header. Permette secondo diversi algoritmi di associare i dati ai relativi header.

Tablin fornisce la lettura linearizzata delle tabelle via web. Strumento realizzato dal W3C, permette di impostare diversi parametri di interpretazione dei dati nelle tabelle.

Torquemada validatore completo via web che nell'output fornisce in cima l'elenco degli errori e sotto la visualizzazione della pagina con accanto il relativo codice sorgente. Come *EvalAccess* e *Hera*, ha riscontrato tutti gli errori delle tabelle rispetto alle WCAG e la solita mancanza di specifica della lingua che riscontrava anche Acc.

In base a queste analisi prima di definire quali validatori è più utile considerare, è bene soffermarci a pensare a cosa vorremmo che il nostro validatore della Stanca facesse.

Se, infatti, pensiamo che la sua unica funzione dovrà essere report di errori, con suggerimenti alla correzione tramite riferimenti generali o esterni, sarà più interessante valutare validatori come *EvalAccess* e *Hera*. Se pensiamo che sia utile fornire all'utente uno strumento che dimostri anche come la tabella viene letta in forma linearizzata è importante considerare *Tablin*, e, infine, se pensiamo sia utile dare all'utente uno strumento intuitivo per rimediare alla mancata attribuzione di header dati, sarebbe il caso di valutare *Com Tab*.

La parte delle tabelle assegnata al nostro gruppo costituirà solo una parte del validatore finale, per questo è anche necessario pensare in termini globali. La linea guida è quella di fornire un report degli errori e controlli da eseguire manualmente, e lasciare all'esterno eventuali correzioni, con magari riferimenti a strumenti che aiutino in questo senso, ma senza implementarli direttamente. Il perché di questa scelta è facilmente intuibile: se immaginiamo che il validatore nella sua forma completa data una pagina ne controllerà la conformità nei riguardi di tutti e 22 i requisiti, sarebbe probabilmente scomodo a un certo punto, o in ripetuti punti della pagina di report, lo strumento di associazione dati/header di cui abbiamo parlato, e se non scomodo risulterebbe comunque poco chiaro.

Per questo la nostra scelta ricade su *EvalAccess* e *Hera* seppur tenendo a mente *Tablin* e l'utilità della sua funzione.

Giunti a questo punto del progetto si passa alla fase di analisi vera e propria, che deve essere eseguita stabilendo uno schema comune e definendo una serie di task e obiettivi da valutare. Abbiamo preferito approcciarci prima alla nostra funzione specifica, la validazione delle tabelle, su i tre validatori in

contemporanea, così da ottenere un miglior confronto, dedicandoci in ultimo step al comportamento generale di *Achecker*.

1.2 Definendo gli obiettivi

Partendo dall'allegato A che riassumeva tutti i requisiti della Stanca a paragone con le WCAG 1.0, abbiamo ora evidenziato i punti fondamentali che un Validatore per Tabelle conformi alla Stanca deve seguire, realizzando un nuovo schema che trovate come allegato C, la cui sintesi è questa:

- Saper distinguere una tabella dati da una tabella di layout
- Verificare l'esistenza del *summary* relativo alle tabelle dati
- Verificare l'esistenza dell'elemento *caption*
- Verificare che gli elementi *th* e gli attributi *headers*, *scope*, *axis* non siano utilizzati in tabelle di stile e che siano correttamente utilizzati nelle tabelle dati
- Verificare e individuare eventuali raggruppamenti dati da *thead*, *tbody*, *tfooter* e *colgroup*
- Verificare l'esistenza di etichette e della loro forma abbreviata
- Verificare eventuali attributi *dir* volti a definire una direzione di lettura.
- Verificare o avvertire di controllare che la tabella abbia senso linea rizzata.

Il comportamento di un validatore prevede indubbiamente che esso sia in grado di individuare e segnalare probabili errori e dare avvertimenti sui controlli da eseguire manualmente, questo implica che sappia leggere sia codice sbagliato e non conforme, ma anche codice corretto. Per questo dobbiamo aggiungere al nostro file usato in precedenza durante la prima fase delle altre varianti.

Prima di procedere mettendo alla prova i validatori dobbiamo riscrivere almeno 4 pagine soggetto di test: layout corretto, layout non conforme, dati accessibile, dati non accessibile.

Abbiamo preferito creare diversi file, ognuno rivolto a un determinato attributo e distinto i casi accessibile/non accessibile.

Oltre ad usare file test troviamo anche utile definire un paio di pagine da siti pubblici, alcuni che si dichiarano "Accessibili" e che dovrebbero esserlo in quanto dell'amministrazione pubblica e altri di alcuni colossi. Abbiamo scelto per questo di prendere ad esempio come siti per testare i validatori:

- due pagine dal sito dell'Università di Bologna e dipartimenti, <http://www.unibo.it/Portale/Studenti/Esamiecalendariolezioni/orari.htm> (in cui la tabella ha funzione unicamente di layout, e <http://internet.unibo.it/sdi-orario-i-anno-corso-di-laurea-triennale> in cui la tabella contiene dati;
- Trenitalia <http://www.trenitalia.com/>, pagina con risultato dei treni disponibili in un certo giorno a un certo orario su una determinata tratta (essendo la pagina dinamica, abbiamo salvato il risultato di una ricerca in locale)
- Turismo Comune di Cervia, http://www.turismo.comunecervia.it/ospitalita/dove_mangiare/-

[ristoranti/](#) (si è vantato di recente di essersi rinnovato per essere accessibile)

I file da noi creati, sono disponibili in versione codice sorgente all'allegato D, mentre quelli di siti noti non sono stati inclusi per la lunghezza del codice sorgente, ma sono disponibili in formato html, nella cartella APPENDICE D allegata alla relazione.

1.3 Valutazione dei validatori

All'allegato E trovate la tabella di risultato dell'analisi di ogni file per ogni validatore, trovate il file nel suo formato originale all'interno della cartella del progetto e non in calce a questo documento per favorirne la leggibilità.

Nel corso delle analisi abbiamo avuto modo di appurare il funzionamento non sempre lineare dei validatori. Spesso i risultati attesi erano diversi da quelli ottenuti, in particolare nel caso di file appositamente creati per non essere accessibili. Cercheremo di entrare ora un po' più nel dettaglio, collegando i risultati alle nostre conclusioni che sono state le congetture base su cui strutturare il nostro validatore.

Indipendentemente dalla struttura dei validatori e dai servizi offerti, i risultati dovevano essere pressoché identici nei rispetti delle WCAG 1.0 e non sempre è stato così. Probabilmente da un lato il layout del report risultati trae in inganno, facendo apparire diversi gli errori valutati, ma dall'altra alcuni di essi interpretano differentemente le occorrenze di alcuni tag.

Prendiamo in considerazione **Achecker**. Questo è l'unico dei tre validatori in grado di valutare secondo linee guida differenti, fra cui anche la nostra italianissima legge Stanca. Come input prevede file da url o file caricati da locale, applicando per default la valutazione in base alle WCAG 2. E' possibile scegliere di validare una pagina anche secondo più criteri allo stesso tempo. Nelle nostre analisi abbiamo in separata sede valutato secondo le WCAG 1.0 per poter paragonare Achecker agli altri validatori, e poi secondo la Stanca, per vedere come la nostra legge veniva trattata.

La pagina di output dei risultati si presenta come un riassunto diviso per gruppi di codice, nelle nostre analisi abbiamo incontrato metadata, script, form, tables, images, text, links, css, headers. La specifica WCAG completa supportata da AChecker prevede anche lists, object, applets, frames. Questa divisione ci è apparsa molto utile, perché chi sta provando a validare le sue pagine anche se non è un esperto potrebbe facilmente capire a quali parti di codice ci si riferisce, e se sono parti di sua competenza cui ha apportato modifiche o se magari dipendono da chi ha strutturato il sito. Ricordiamoci che questo progetto è stato commissionato dalla regione per fornire uno strumento disponibile alle altre PA, e che spesso i dipendenti pubblici usano CMS per gestire i contenuti e non sono reali esperti di codice. Per cui se dalla validazione esce un errore su una tabella, probabilmente sono in grado di capirlo poiché causa loro che hanno male inserito i dati, ma se viceversa riguardasse un applet è molto più probabile che la colpa sia di chi ha realizzato la struttura.

Oltre a questa divisione le altre ci paiono però meno chiare. La legenda ci istruisce riguardo al significato di *Conosciuti, Probabili e Potenziali*, ma forse il livello di priorità associato a “Errore” o “Avviso di controllo”, sarebbero stati più chiari e più utili allo scopo di validazione. In fondo anche questo sistema non è così sbagliato perché separa fra ciò che è certamente sbagliato da ciò che va controllato, probabilmente associato a un indicazione circa i livelli di priorità sarebbe stato perfetto. Cliccando su ogni gruppo, o su ogni sezione si apre poi l’elenco degli errori, che riporta al fianco anche il numero della riga del nostro codice in cui è stato trovato. E’ poi possibile cliccare anche su ogni errore ed ottenere dei suggerimenti. Ora, l’esperto sicuramente con l’indicazione della riga del codice trova e modifica, ma chi non ha le capacità di comprendere il codice e ha solo inserito una news nel portale, potrebbe rinunciare a capire. Per questo sarebbe stato utile anche uno strumento che mostrasse tramite un’anteprima in che punto della pagina si trova l’errore, ove possibile naturalmente. Anche l’indicazione dell’errore non offre un suggerimento, o un link al sito del W3C dove ottenere consigli ed esempi concreti su come poter attuare la correzione.

Una volta osservata la struttura, passiamo a ciò che in sintesi, è emerso dai nostri test:

- non raggruppa lo stesso errore, ma fornisce una riga per ogni errore. Per cui se ho 20 immagini tutte senza ALT, troverò 20 righe che me lo segnalano anziché una che mi riassume tutte le righe di codice in cui ho commesso questo errore
- nel momento in cui suggerisce di controllare che una tabella abbia senso linearizzata, non fornisce un link o uno strumento utile a tale controllo
- non viene indicato a quale livello di priorità corrisponde l’errore
- se una tabella non contiene l’attributo *summary* o il tag *caption* in automatico la rileva come tabella di layout e non fornisce nessuno dei warning sul controllo di intestazioni e associazione dati a intestazioni. Cosa sbagliata a nostro parere, perché non il non inserimento di uno di questi due elementi potrebbe essere semplicemente un errore. Infatti i nostri file appositamente non accessibili, sono stati identificati come tabelle di layout e per questo bypassati senza errori, con l’unico warning di controllare la linearizzazione.
- Quando i risultati sono divisi fra più pagine, lo strumento di navigazione per cambiare pagina appare solo in alto, e potrebbe sfuggire a una prima occhiata che i risultati sono divisi in più pagine.
- Il linguaggio di report degli errori è specifico

La struttura del codice è realizzata in java e i dizionari delle diverse linee guida e del testo che apparirà al verificarsi di errori sono scritti in xml. Da notare che sebbene sia possibile cliccare su “Italiano” per accedere alla lingua italiana, la versione della traduzione è molto parziale. Abbiamo anche trovato scomoda la posizione del link a “Report” che consente di tornare alla schermata di riepilogo una volta entrati nel dettaglio di una sezione. Probabilmente sul lato sinistro salterebbe all’occhio più facilmente. Altra pecca, spesso il sito è lento, e ancora più spesso risulta non raggiungibile. La maggioranza dei test sono stati effettuati in giorni separati dagli altri validatori, proprio per questo motivo.

Hera

Permette di validare solo pagine singole a partire da un determinato url solo relativamente alle WCAG 1.0. Non sono previste forme per caricare file da locale o scrivere pezzi di codice. E' sviluppato in php, con supporto di sql e xml. Sul sito è anche presente una guida supportata da immagini che spiegano il funzionamento del validatore, del quale sembrano essere disponibili anche plugin per browser, ma purtroppo queste ultime informazioni sono scritte unicamente in spagnolo. Il codice sorgente è liberamente disponibile e scaricabile.

Le lingue disponibili sono diverse, spagnolo, inglese, tedesco, francese, catalano, rumeno, serbo e italiano. Diversamente da AChecker, in questo caso la traduzione è totale ed include ogni segnalazione e report di errore.

I risultati della validazione sono espressi in una tabella riassuntiva, che elenca in un sommario il numero di controlli effettuati e il tempo impiegato, indicandoci poi il numero di errori totale e quello dei controlli da effettuare manualmente. Sotto troviamo una tabella riassuntiva divisa per livelli di priorità e *Verificare, Giusto, Sbagliato, Non applicabile*. In questo modo possiamo monitorare separatamente ciò che dobbiamo controllare manualmente, ciò che è sicuramente sbagliato, ma anche sollevarci guardando ciò che abbiamo fatto giusto. Molti potrebbero ritenere inutile controllare i punti giusti, ma noi lo troviamo uno stimolo per chi ha realizzato la pagina, nonché uno strumento di controllo utile a chiarire eventuali dubbi. Se ad esempio ho scritto il codice pensando di fare giusto, ma non ne sono certa, questo potrebbe essere la conferma grazie alla quale continuerò ad utilizzare lo stesso metodo. L'unica colonna forse irrilevante è quella dei punti non applicabili, d'altronde anche questa potrebbe essere per controllo.

Già questa suddivisione dei risultati ci sembrava soddisfacente, permettendoci di accedere al dettaglio di ciò che ci interessa, ma sotto al riassunto per priorità troviamo un altro interessante sistema di navigazione che suddivide per requisito. Ad esempio, noi interessati alle tabelle potremmo semplicemente cliccare su *Linea guida 5* ed accedere unicamente a warning, errori e punti giusti relativi a queste.

Entrando nel dettaglio delle segnalazioni, troviamo citato un pezzetto della linea guida di riferimento, seguito da una nota meno formale e più intuitiva anche per i meno esperti. Alla destra di ogni dettaglio troviamo 3 icone che ci permettono di: visualizzare istruzioni su come correggere, visualizzare l'errore nell'anteprima della pagina, visualizzare l'errore nel codice sorgente della pagina valutata. Troviamo queste opzioni molto utili, anche se forse sarebbe più rapido avere anche la riga del codice in cui si è trovato l'errore nel primo dettaglio. Gli esperti in questo modo potrebbero agire sul codice direttamente senza un passaggio in più. Nelle istruzioni inoltre, troviamo che sarebbe utile inserire il link alla pagina della specifica W3C per maggiori approfondimenti su come poter correggere.

Altre utilità sono rappresentate dalla tabella riassuntiva per priorità e tipo di warning che rimane in forma miniaturizzata nella parte superiore di ogni pagina. Questo permette un passaggio in meno per muoversi fra un livello e l'altro.

Come nelle pagine di AChecker in alto a destra troviamo un menù per tornare alla home, al riepilogo generale, ma anche per ottenere un aiuto, che aggiunge maggiori informazioni sul warning, e due funzioni nuove. Una si chiama form, e permette di aggiungere dei commenti ad ogni singolo warning, l'altra si chiama report e consente di aggiungere dei commenti alla valutazione eseguita ed

esportare poi l'intero risultato in formato pdf, html o rdf. Tornando all'esempio fatto nei commenti su AChecker, se un dipendente di una pubblica amministrazione tenta di validare le pagine ed ottiene errori che non sa come gestire, perché magari dipendono da chi ha realizzato il CMS, potrebbe essere utile esportare i risultati per poi inviarli a chi di dovere. Viceversa, se esiste un dipartimento informatico all'interno dell'ente, potrebbe occuparsi della validazione e poi inviare i suggerimenti ai vari impiegati addetti ad inserire contenuti.

Hera ci è sembrato davvero un validatore ben strutturato, facendo apparire la validazione meno difficile e tecnica di quello che sembra, grazie al suo modo di esporre gli errori. Veramente originali, o quantomeno utili sono i tool aggiuntivi come commenti e report. Se vogliamo trovare qualche pecca però, pensiamo che inserire i link al W3C come aiuto sarebbe utile, mentre invece questo appare solo nel caso di DTD non valido, rimandando direttamente alla pagina di validazione del W3C. Nel nostro ambito di interesse delle tabelle poi, Tablin non è mai stato nominato, e pensiamo che sarebbe utile fare un riferimento a questo strumento quando si parla di linearizzazione.

Diversamente da AChecker comunque, sottoponendo ad Hera la pagina di test *04_tabelleNonAccessibili_TD*, non riceviamo solo il warning relativo al controllo sulla linearizzazione di tabelle di layout, ma viene segnalata la mancanza dell'attributo summary, di caption, di th e dell'alternativo linearizzato. Gli errori sono quindi duplicati, ma in questo modo al soggetto che esegue i test è possibile prendere visione dei diversi tipi di errore che potrebbe aver commesso.

Evalaccess

Realizzato in java e xml, valida le pagine in base alle WCAG 1.0. La piattaforma è unicamente in lingua inglese e il codice sorgente non è scaricabile liberamente. Abbiamo contattato il team sviluppatore, che ci ha parlato del progetto di evoluzione di EvalAccess, nella nuova versione ogni utente potrà caricare il proprio dizionario di linee guida realizzato in xml. Al momento non è tuttavia disponibile e del vecchio sorgente non hanno documentazione, per cui dopo la nostra richiesta ci è stato inviato solamente un file di testo per comprendere la struttura e il file XML del dizionario WCAG 1.0.

Come gli altri offre la possibilità di validare a partire da un determinato URL, ma a differenza degli altri è possibile eseguire anche l'analisi su gruppi di pagine, o inserire un pezzo di codice sorgente in un'apposita form.

Possiamo scegliere a quale livello delle WCAG 1.0 vogliamo validare la pagina, come in AChecker e se farci mostrare solo gli errori o anche i warning. L'output dei risultati è per elenco, con una mini tabella riassuntiva che divide per priorità ed *errore o warning*. L'elenco parte con gli errori divisi in blocchi per priorità, sotto seguono i warning divisi allo stesso metodo.

Ogni warning viene ripetuto in singola istanza, raggruppando in una colonna della tabella in numero di righe in cui è stato rilevato. Il messaggio è molto sintetico e tecnico, e per eventuali suggerimenti si è rinviiati alla pagina del W3C. Troviamo questo report molto sintetico e tecnico e più indirizzato a utenti esperti, il nostro impiegato comunale esempio che si occupa solo di usare il CMS per inserire news potrebbe avere diversi problemi nell'utilizzare questo validatore. Come nel caso di AChecker se trova una tabella priva di summary e caption giunge alla conclusione che è di layout, senza fornire informazioni circa il controllo che sia effettivamente una tabella volta all'impaginazione.

Sebbene siano presenti gli utili link alle linee guida WCAG per maggiori informazioni, non è indicato nessun riferimento a Tablin per controllare la linearizzazione delle tabelle.

In conclusione abbiamo trovato che ognuno dei tre validatori ha punti di forza e punti deboli, e che per definirne uno migliore dell'altro è necessario sapere chi è il soggetto che lo utilizza. Infatti AChecker ed EvalAccess sono sicuramente più indicati per persone più esperte e pratiche con la programmazione web, mentre Hera è più user-friendly. In quest'ultimo sono stati riscontrati diverse funzioni utili che gli altri non presentano, come la possibilità di commentare i risultati e di esportare un report. La diversità sotto alcuni aspetti di questi strumenti, ci ha permesso di formulare meglio le nostre idee implementative riguardo al nostro validatore della Stanca, troverete le nostre congetture al capitolo seguente.

2. Progettazione del validatore

Grazie alle analisi eseguite sui validatori esistenti, la nostra idea di cosa volevamo ottenere era abbastanza chiara. Tuttavia, dovendoci occupare soltanto della parte relativa alle tabelle del validatore il prodotto finale è risultato ben diverso, cercando di realizzare qualcosa che fosse facilmente riusabile nella struttura definitiva.

Il punto di partenza è stato l'allegato C realizzato durante i testing sull'esistente, nel quale abbiamo elencato tutti i vari controlli sulle tabelle che secondo noi sono necessari.

Riassumendoli in breve abbiamo:

- cercare le occorrenze del tag `TABLE` come presupposto di partenza
- dato il tag `TABLE` cercare l'attributo `summary`, se presente restituire un warning di controllo correttezza e descrittività del sommario della tabella, se non presente restituire warning di avviso che per le tabelle dati è necessario inserire l'attributo sommario, se presente restituire warning che avverta che nelle tabelle di layout l'attributo `summary` non va utilizzato
- dato il tag `TABLE` cercare tag `CAPTION`. Se presente, warning di avviso sul carattere descrittivo che deve avere il titolo della tabella e sull'evitare questo attributo in tabelle di layout. Se non presente warning di avviso che nelle tabelle dati è bene inserire il titolo della tabella tramite l'elemento `caption`
- verificare l'esistenza dei TH. Se presenti warning di avviso sul loro utilizzo solo ai fini di intestazione di righe o colonne in tabelle dati, se non presenti warning di avviso poiché è necessario utilizzarli nelle tabelle dati
- verificare esistenza di ID nei TH, utili a collegare le intestazioni di righe e colonne ai dati tramite headers
- verificare esistenza di ABBR nei TH ed eventualmente nei TD, utili a fornire versioni abbreviate di campi di intestazione articolati

- verificare aggregazione dei dati per righe o colonne tramite `thead`, `tfoot`, `tbody`, `colgroup` e `col` sempre nei casi di tabelle dati
- verificare che tabelle di layout abbiano senso se linea rizzate
- verificare esistenza di eventuali attributi `DIR` che definiscano la direzione di lettura della tabella
- verificare l'associazione fra dati e intestazioni tramite `SCOPE`, `HEADERS` e `AXIS`.

Una volta definito l'elenco dei controlli necessari, si è passati a definire la struttura del validatore. Considerando le linee guida fornite e che l'applicazione finale sarà fruibile via web abbiamo scelto php come linguaggio di programmazione in combinazione con una grammatica in xml per l'output della validazione.

Per il tool in versione definitiva ci pare utile fornire strumenti volti al caricamento di un file da url, da locale, di gruppi di pagine ed eventualmente da pezzi di codice scritti in una form. Pensando che queste funzioni saranno gestite a livello globale, ci siamo incentrate inizialmente su altri fattori, lasciando come ultimo step quello di provare a implementare la validazione da un url.

In questa versione quindi, il file da validare è definito alla variabile `$path`, definita in cima al file `index.php` che rappresenta il cuore dell'applicazione. Con la variabile `$path` viene richiamato il costruttore `new DomDocument` che ci permette di poter manipolare il codice con funzioni di tipo `Dom`.

In base alle nostre conoscenze personali, abbiamo optato per non smontare Hera, come avevamo inizialmente pensato, ma riscrivere le funzioni a noi necessarie da zero usando funzioni di tipo `Dom` anziché manipolazione di stringhe.

Dovendo restituire in output un file XML, i dati dell'analisi vengono memorizzati nella variabile `$result`. La fase di creazione del `new DomDocument` è eseguita fuori dal ciclo `for`, poiché viene il file dei risultati viene creato una volta sola.

Nella variabile `$file`, come da specifiche DOM W3C, abbiamo un array di `NodeList element`. L'accesso a questi si ha per mezzo di un costrutto di scorrimento e l'identificazione avviene tramite manipolazione con la funzione `$file->item($i)`. Questa ritorna un `NodeElement` sul quale è possibile eseguire tutte le funzioni di valutazione a noi indispensabili.

I vari controlli sono stati divisi per tag `html`, e per identificare i gruppi di controllo all'interno del nostro programma abbiamo utilizzato una separazione per mezzo di commenti del tipo: `"// PROPRIETA' <TAG>"`. Occupandoci delle tabelle, il primo controllo è volto a trovare le occorrenze del tag `TABLE`, per poi eseguire il resto dei controlli all'interno. Ciò è realizzato tramite un costrutto `for`, che applica poi i controlli per ogni `NodeElement` tramite funzioni `if/else`.

Validatori e Validazione di tabelle

Il codice XML generato relativo al Warning o all'errore, segue la struttura seguente:

```
//WARNING: la correttezza dell'organizzazione dei dati
$warning = $result->createElement("warning");
$root->appendChild($warning);

$priorita = $result->createElement("priorita");
$warning->appendChild($priorita);
$livello = $result->createTextNode("1");
$priorita->appendChild($livello);

$stanca = $result->createElement("stanca");
$warning->appendChild($stanca);
$requisito = $result->createTextNode("Requisito n. 10");
$stanca->appendChild($requisito);

$legge = $result->createElement("legge");
$warning->appendChild($legge);
$stesto = $result->createTextNode("Attenzione controllo
manuale: E' stata rilevata la presenza del tag colgroup, utile a raggruppare i
dati per colonne, assicurarsi quindi che i dati siano effettivamente
organizzati per colonne e i gruppi siano corretti e creati tramite i tag COL e
gli attributi SPAN. ");
$legge->appendChild($stesto);

$urlstanca = $result->createElement("urlstanca");
$warning->appendChild($urlstanca);
$link = $result->createTextNode("http://www.pubbliaccesso.it/normative/DM080705-A.htm");
$urlstanca->appendChild($link);

$wcag = $result->createElement("wcag");
$warning->appendChild($wcag);
$checkpoint = $result->createTextNode("5.2");
$wcag->appendChild($checkpoint);

$urlwcag = $result->createElement("urlwcag");
$warning->appendChild($urlwcag);
$link = $result->createTextNode("http://www.w3.org/TR/WAI-
WEBCONTENT/#gl-table-markup#gl-table-markup");
$urlwcag->appendChild($link);

$suggerimenti = $result->createElement("suggerimenti");
$warning->appendChild($suggerimenti);
$link = $result->createTextNode("http://www.w3.org/TR/html401/struct/tables.html#edef-
COLGROUP");
$suggerimenti->appendChild($link);
$stesto = $result->createTextNode("Se si hanno dubbi relativi
al raggruppamento dei dati per colonne, controllate la pagina della specificca
dell'attributo COLGROUP");
$suggerimenti->appendChild($stesto);
```

Abbiamo scelto di riportare sia il collegamento alla Stanca che alle WCAG 1.0, poiché nello stesso regolamento attuativo della legge italiana sono posti i riferimenti ai relativi punti delle linee guida dettate dal W3C. Spesso abbiamo trovato utile aggiungere anche un campo per suggerimenti, in cui il link rimanda alle linee guida del W3C sulle tecniche per passare un determinato checkpoint. La descrizione dell'errore o del warning vuole seguire l'esempio di Hera, fornendo sia il dettaglio tecnico, ma anche un linguaggio il più possibile esplicativo per i meno esperti.

In questa versione del validatore sono presenti molti commenti da noi utilizzati durante il debug e il test per capire e sperimentare come stava

funzionando. E' possibile attivare o disattivare la stampa dei vari echo settando la variabile \$test rispettivamente a 1 o 0. Per mostrarvi il risultato della validazione in xml è già stato settato a 0.

Stiamo attualmente ancora lavorando all'inserimento in un layout accessibile dell'output xml e all'inserimento della funzione per valutare un file da url. In questa versione per testare il programma con file diversi è necessario modificare il contenuto della variabile \$path.

Abbiamo infine testato il file oltre che in locale su un web server Linux per verificare il corretto funzionamento.

3. Conclusioni

Sebbene l'ottimizzazione del nostro validatore sia ancora in corso abbiamo raccolto diverse idee circa le funzioni che dovrebbe avere il validatore finale. Pur centrando il nostro interesse verso le tabelle abbiamo scoperto alcuni controlli utili eseguiti dagli strumenti che abbiamo analizzato, come il warning automatico in caso di immagini con estensioni .gif riguardo all'evitare lo sfarfallio e le immagini in movimento. Durante il periodo di svolgimento di questo progetto Cristina ha avuto modo di incontrare direttamente persone che si stavano scontrando con il problema dell'accessibilità all'interno delle pubbliche amministrazioni, parliamo in particolare dei responsabili del sito turistico del Comune di Cervia, che stanno svolgendo un'intensa attività volta a renderlo conforme alla legge Stanca. Si tratta di personale senza formazioni informatiche specifiche che per realizzare le pagine utilizza un cms dato in licenza dalla provincia e fatto personalizzare da una società esterna. Il fulcro principale del problema è per loro capire come ad esempio collegare i dati delle tabelle tramite intestazioni e come impaginare documenti senza il supporto di tabelle. Nel loro merito non hanno a che fare direttamente con il codice html, per cui parlare di linearizzazione, attributi headers, scope, colgroup etc, è quasi come parlare in arabo. Proprio dopo questo incontro la modalità di presentazione dei risultati di Hera ha assunto un carattere diverso ai nostri occhi.

In conclusione troviamo che essendo il progetto destinato alla regione Emilia Romagna, sia fondamentale fornire non solo un output degli errori di carattere prettamente tecnico con indicazione dei numeri di riga di codice contenenti gli errori, ma che sia necessario fornirne anche una versione comprensibile a coloro che non agiscono direttamente sul codice, fornendo come Hera la possibilità di vedere l'errore tramite l'anteprima della pagina e dando indicazioni non solo tramite elenco dei tag, ma fornendone anche una versione descrittiva. Parlando di persone non necessariamente con formazione informatica è ancor più utile inserire dei suggerimenti e collegamenti diretti a pagine in cui trovare informazioni su come poter correggere gli errori.

Durante le fasi iniziali di discussione del progetto si è parlato anche di eventuali aiuti da personale della regione a supporto degli enti minori per la validazione, per questo potrebbe anche essere utile inserire strumenti per commentare l'operato della validazione automatica e di report per poter poi esportare i risultati e inviarli agli addetti.

ALLEGATO A

Validatori e Validazione di tabelle

WCAG			SECTION508	STANCA	
CHECK POINT	PRIORITA'	ENUNCIATO		ENUNCIATO	REQUISITO
		3 LIVELLI PRIORITA': LIVELLO 1 PER NON ESCLUDERE, LIVELLO 2 PER RIMUOVERE BARRIERE ALL'ACCESSO, LIVELLO 3 MIGLIORARE L'ACCESSO		VERIFICA TECNICA + VERIFICA SOGGETTIVA	
3.03	PRIORITA' 2	3.3 USARE CODICE NON SOLO SINTATTICAMENTE MA ANCHE SEMANTICAMENTE IN MODO CORRETTO (H1 H2 E VIA DICENDO USATI PER TITOLI NON PER DARE STILE E USATI IN SCALA		VERIFICA TECNICA: VALIDAZIONE AUTOMATICA - VALIDAZIONE CON SUPPORTO ESPERTO PER VALUTARE IL CORRETTO USO SEMANTICO DEL CODICE A MARCATORI - CONTROLLO MULTI BROWSER - DIFFERENZE LUMINOSITA' E COLORI - REDAZIONE RAPPORTO	
		usare testo quando possibile in alternativa a immagini e suono (ALT E LONGDESC). NON FARE AFFIDAMENTO SOLO A COLORE			
3	PRIORITA' 2	LINEA GUIDA 3 - USARE LINGUAGGIO A MARCATORI SPECIFICO SE ESISTE (MATHML PER FORMULE, CSS PER STILE) DICHIARARE IL DTD E USARE CODICE IN MODO CORRETTO		HTML VERSIONE 4.01 / XHTML VERSIONE 1.0	
3 + 11.1 - 11.2	PRIORITA' 2	DICHIARARE DTD (3) USARE TECNOLOGIE W3C QUANDO DISPONIBILI E NON QUELLE DISAPPROVATE (TAG FONT)		DTD TIPO STRICT	1
3.3 - 3.6 - 3.7	PRIORITA' 2	3.3 USARE FOGLI DI STILE, NON USARE TABELLE PER LAYOUT POSSIBILMENTE, DICHIARARE CORRETTAMENTE CITAZIONI (BLOCKQUOTE 3.7), ELENCHI (UL-LI-DL 3.6)		FOGLI DI STILE	

Validatori e Validazione di tabelle

12.01	PRIORITA' 1	TITOLO AI FRAME	1194.22 (i)	NO A FRAME IN SITI NUOVI, SITI VECCHI POSSONO IN PRIMA SEDE TENERLI CON HTML 4,01 / XHTML 1 CON DTD FRAMESET MA SOLO SE STILE ESTERNO CON CSS E FRAME CON TITOLO E SCOPO SPECIFICATI + PIANIFICARE VERSO DTD STRICT	2	
12.02	PRIORITA' 2	DEFINIRE SCOPO E LONG DESC DEI FRAME SE NON E' EVIDENTE DA TITOLO				
1.01	PRIORITA' 1	Fornire un equivalente testuale per ogni elemento non di testo (per esempio, mediante "alt", "longdesc" o contenuto nell'elemento stesso).	1194.22 (a)	FORNIRE ALTERNATIVA TESTUALE A CONTENUTO NON TESTUALE, IN FORMA COMMISURATA A FUNZIONE DELL'OGGETTO NEL CONTESTO (ESEMPIO PER L'IMMAGINE DI UN LOGO L'ALT E' MEGLIO INDICARLO "" VUOTO, DATO CHE NON E' ESISTENZIALE SAPERE COS'E' IL LOGO). AGGIORNARLI QUANDO IL CONTENUTO DINAMICO CAMBIA.	3	
6.02	PRIORITA' 1	AGGIORNARE LA VERSIONE ALTERNATIVA QUANDO IL CONTENUTO DINAMICO CAMBIA				
2.01	PRIORITA' 1	ASSICURARSI CHE L'INFORMAZIONE NON SIA VEICOLATA UNICAMENTE AL COLORE MA PERCEPIBILE GRAZIE AI MARCATORI E AL CONTESTO	1194.22 (c)	NON VINCOLARE L'INFORMAZIONE AL COLORE	4	
7.01	PRIORITA' 1	EVITARE SFARFALLII FINO A QUANDO I BROWSER NON PERMETTERANNO ALL'UTENTE DI GESTIRLI	1194.22 (j)	EVITARE LAMPEGGIAMENTI E SE NON SI POSSONO ELIMINARE AVVISARE PRIMA DELLA LORO PRESENZA (EPILESSIA)	5	
7.2 - 7.3	PRIORITA' 2	EVITARE SCRITTE LAMPEGGIANTI E MOVIMENTO NELLE PAGINE FINCHE' I BROWSER NON PERMETTERANNO DI CONTROLLARLO				
2.02	PRIORITA' 2 PER IMMAGINI / PRIORITA' 3 PER TESTO	ASSICURARSI CHE IL CONTRASTO TRA COLORE SFONDO E TESTO SIA SUFFICIENTE PER COLORO CHE HANNO DEFICIT DI PERCEZIONE COLORI O SE IL CONTENUTO VIENE VISUALIZZATO SU SCHERMO IN BIANCO E NERO		GARANTIRE CONTRASTO TRA CONTENUTO INFORMATIVO E SFONDO (FOREGROUND - BACKGROUND) (ANCHE NEI SUONI) - NO A TESTO SOTTO FORMA DI IMMAGINE	6	

Validatori e Validazione di tabelle

9.01	PRIORITA' 1	IMMAGINI SENSIBILI LATO CLIENT, SALVO IL CASO IN CUI NON ABBIANO FORME GEOMETRICHE VALIDE	1194.22 (f)	MAPPE IMMAGINI SENSIBILI LATO CLIENT E NON SERVER, LATO SERVER SOLO SE AREE SENSIBILI NON SEGUONO FORME STANDARD DEFINITE NEL DTD	7	
1.02	PRIORITA' 1	Fornire ridondanti collegamenti di testo per ogni zona attiva di una immagine sensibile sul lato server.	1194.22 (e)	FORNIRE COLLEGAMENTI DI TESTO ALTERNATIVI PER INFO E AREE RAGGIUNGIBILI TRAMITE MAPPE IMMAGINE	8	
5.01	PRIORITA' 1	PER LE TABELLE IDENTIFICARE INTESTAZIONE RIGHE E COLONNE (TD CELLA / TH INTESTAZIONE)	1194.22 (g)	NELLE TABELLE USARE MARCATORI PER IDENTIFICARE INTESTAZIONE RIGHE E COLONNE	9	TABELLE
5.05	PRIORITA' 3	PER LE TABELLE FORNIRE SOMMARI				
5.06	PRIORITA' 3	PER LE TABELLE FORNIRE ABBREVIAZIONI PER LE INTESTAZIONI (TH ABBR)				
5.02	PRIORITA' 1	PER TABELLE CON PIU' LIVELLI LOGICI INTESTAZIONE RIGHE COLONNE, USARE MARCATORI PER ASSOCIARE DATI E INTESTAZIONE (THEAD TFOOT TBODY COL COLGROUP ATTRIBUTI "axis", "scope" e "headers" per descrivere relazioni più complesse fra i dati.	1194.22 (h)	NELLE TABELLE CON DUE O PIU' LIVELLI DI INTESTAZIONE USARE GLI ATTRIBUTI DEL DTD NECESSARI PER ASSOCIARE CELLE DATI CELLE INTESTAZIONE	10	
3.03	PRIORITA' 2	USARE FOGLI DI STILE PER PRESENTAZIONE E IMPAGINAZIONE	1194.22 (d)	USARE FOGLI DI STILE PER LA PRESENTAZIONE IN MODO CHE CONTENUTI SIANO LEGGIBILI ANCHE SE I FOGLI DI STILE SONO DISABILITATI O NON SUPPORTATI	11	
6.01	PRIORITA' 1	ORGANIZZARE DOCUMENTI IN MODO CHE POSSANO ESSERE LETTI ANCHE SENZA FOGLIO DI STILE				
3.04	PRIORITA' 2	3.4 USARE MISURE RELATIVE IN EM O % E NON ASSOLUTE IN CSS		PRESENTAZIONE E CONTENUTI DEVONO ADATTARSI A INTERFACCIA USATA SENZA SOVRAPPOSIZIONI DEGLI OGGETTI PRESENTI O PERDITA INFORMAZIONE, ANCHE CON INGRANDIMENTO E RIDUZIONE DEI PARAMETRI DI DEFAULT (LAYOUT LIQUIDO)	12	

Validatori e Validazione di tabelle

5.03	PRIORITA' 2	NON USARE TABELLE PER LAYOUT SE IL CONTENUTO NON E' COMPRESIBILE LETTO IN FORMA LINEARIZZATA. IN CASO NON SIA COMPRESIBILE FORNIRE UNA VERSIONE ALTERNATIVA		SE TABELLE USATE PER LAYOUT ACCERTARSI CHE CONTENUTO SIA COMPRESIBILE SE TABELLA VIENE LINEARIZZATA E USARE ATTRIBUTI TABELLA SECONDO VALORE SEMANTICO	13	TABELLE
5.04	PRIORITA' 2	SE SI USA TABELLA PER LAYOUT NON USARE MARCATORI DI STRUTTURA PER DEFINIRE LO STILE (NON USARE TH PER METTERE CONTENUTO CENTRATO, MA DEFINIRE UNA CLASSE PER TD)				
10.02	PRIORITA' 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. ETICHETTA DEVE PRECEDERE CONTROLLO E SU STESSA RIGA		NELLE FORM ASSOCIARE ESPLICITAMENTE ETICHETTE A CONTROLLI POSIZIONANDOLE IN MODO DA FACILITARE COMPILAZIONE A CHI USA TECNOLOGIE ASSISTIVE	14	
12.04	PRIORITA' 2	ASSOCIARE EPLICITAMENTE ETICHETTE AI LORO CONTROLLI (LABEL + ATTRIBUTO FOR)				
6.03	PRIORITA' 1	ASSICURARE PAGINE USABILI QUANDO SCRIPT APPLET E OGGETTI DI PROGRAMMAZIONE SONO DISABILITATI O NON SUPPORTATI. SE NON POSSIBILE FORNIRE EQUIVALENTE TESTUALE IN UNA PAGINA ALTERNATIVA OPPURE USARE SCRIPT LATO SERVER INVECE DI LATO CLIENT	1194.22 (l),1194.22 (m)	GARANTIRE CHE PAGINE FUNZIONINO UGUALMENTE SENZA SCRIPT, APPLET E ALTRI OGGETTI, O FORNIRE UN ALTERNATIVA TESTUALE	15	
6.04	PRIORITA' 2	PER QUANTO RIGUARDA SCRIPT E APPLET VERIFICARE CHE SIANO INDIPENDENTI DA SPECIFICI DISPOSITIVI DI INPUT	1194.22 (l),1194.22 (m)	GESTORI DI EVENTI (ACTION SCRIPT....) SIANO INDIPENDENTI DA UN DETERMINATO DISPOSITIVO DI INPUT	16	

Validatori e Validazione di tabelle

9.02	PRIORITA' 2	ASSICURARSI CHE OGNI ELEMENTO CON UNA SPECIFICA INTERFACCIA SIA INDIPENDENTE DAI DISPOSITIVI DI INPUT				
9.03	PRIORITA' 2	NEGLI SCRIPT USARE GESTORI DI EVENTO LOGICI PIUTTOSTO CHE DA DISPOSITIVO				
8.01	PRIORITA' 1 se funzioni importanti, altrimenti PRIORITA' 2	Fare in modo che elementi di programmi come script e applet siano direttamente accessibili o compatibili con le tecnologie assistive	1194.22 (l),1194.22 (m)	GARANTIRE CHE FUNZIONI E INFO VEICOLATE DA OGGETTI DI PROGRAMMAZIONE, APPLET, SCRIPT E TECNOLOGIE NON DEFINITE DA GRAMMATICHE FORMALI PUBBLICATE SIANO ACCESSIBILI O COMPATIBILI CON TECNOLOGIE ASSISTIVE	17	
1.03	PRIORITA' 1	FINO A QUANDO INTERPRETI NON LEGGERANNO CONTENUTO FILMATO FORNIRE DESCRIZIONE AUDIO E SINCRONIZZARLA				
1.04	PRIORITA' 1	Per ogni presentazione multimediale temporizzata (per es. un film o una animazione), sincronizzare alternative equivalenti (per es. didascalie o descrizioni parlate del filmato) con la presentazione	1194.22 (b)	SE FILMATI O PRESENTAZIONI MULTIMEDIALI INDISPENSABILI PER FORNIRE INFORMAZIONE COMPLETA INSERIRE SOTTOTITOLAZIONE SINCRONIZZATA O ASSOCIARE RIASSUNTO/ETICHETTA A SECONDA DEL GRADO DI IMPORTANZA E DIFFICOLTA' (ESEMPIO PRESENTAZIONI IN TEMPO REALE)	18	
13.01	PRIORITA' 2	IDENTIFICARE CON CHIAREZZA L'OBIETTIVO DI OGNI COLLEGAMENTO				
13.06	PRIORITA' 3	FINO A QUANDO NON LO FANNO INTERPRETI RAGGRUPPARE COLLEGAMENTI CORRELATI, IDENTIFICARE GRUPPI E FORNIRE METODI PER SALTARLI.	1194.22 (o)	RENDERE CHIARA LA DESTINAZIONE DEI LINK CON TESTI SIGNIFICATIVI ANCHE SE LETTI FUORI CONTESTO E ALTERNATIVI ESPLICATIVI DELLA FUNZIONE. PREVEDERE MECCANISMI PER EVITARE LETTURA DI COLLEGAMENTI RIPETITIVA AD OGNI PAGINA (TIPICAMENTE MENU)	19	

Validatori e Validazione di tabelle

7.04	PRIORITA' 2	FINO A QUANDO GLI INTERPRETI NON RENDERANNO POSSIBILE BLOCCARE L'AUTOAGGIORNAMENTO, NON PREDISPORRE PAGINE CON AUTOAGGIORNAMENTO (HTTP-EQUIV REFRESH)	1194.22 (p)	AVVISARE L'UTENTE SE ESISTE UN TEMPO MASSIMO PER IL COMPLETAMENTO DI DETERMINATE AZIONI VOLTO ALLA FRUIZIONE DEL SERVIZIO E FORNIRE ALTERNATIVE	20	
7.05	PRIORITA' 2	NON USARE MARCATURA PER REINDIRIZZAMENTO AUTOMATICO DELLE PAGINE FICHE' GLI INTERPRETI NON RENDERANNO POSSIBILE BLOCCARLO				
				RENDERE LINK ATTIVABILI TRAMITE COMANDI DA TASTIERA O SISTEMI DI PUNTAMENTO DIVERSI DAL MOUSE. GARANTIRE DISTANZA VERTICALE E SPAZIATURA ORIZZONTALE LINK DI ALMENO 0,5em , DISTANZE PULSANTI ALMENO 0,5em E DIMENSIONI PULSANTI TALI DA RENDERE LEGGIBILI LE ETICHETTE	21	
11.04	PRIORITA' 1	SE NONOSTANTE OGNI SFORZO NON E' POSSIBILE CREARE UNA PAGINA ACCESSIBILE FORNIRE IL LINK ALTERNATIVO DI UNA PAGINA CHE USI LE TECNOLOGIE W3C E SIA EQUIVALENTE PER CONTENUTI NONCHE' AGGIORNATA CON LA STESSA FREQUENZA	1194.22 (k)	IN SEDE DI PRIMA APPLICAZIONE SE NON E' POSSIBILE RISPETTARE I REQUISITI FORNIRE IL COLLEGAMENTO A UNA PAGINA CHE LI RISPETTI E CHE VENGA AGGIORNATA CON LA STESSA FREQUENZA	22	

ALLEGATO B

Validatori e Validazione di tabelle

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
<title>Tabelle non accessibili</title>
</head>

<body>
<p>Qui sotto metto tabelle non accessibili, lo stile lo
definisco qui e non in css e metto doppie intestazioni e senza
caption</p>

<table align="center" bgcolor="#585554" border="1px solid
#000000">
<tr>
<td></td>
<th>docente</th>
<th>orario</th>
</tr>
<tr>
<th>Algoritmi</th>
<td>Margara</td>
<td>9-11</td>
</tr>
<tr>
<th>Matematica</th>
<td>Fioresi</td>
<td>16-18</td>
</tr>
</table>

<p>Stessa tabella con caption</p>

<table align="center" bgcolor="#33FFFF" border="1px solid
#000000">
<caption>Insegnamenti</caption>
<tr>
<td></td>
<th>docente</th>
<th>orario</th>
</tr>
<tr>
<th>Algoritmi</th>
<td>Margara</td>
<td>9-11</td>
</tr>
<tr>
<th>Matematica</th>
<td>Fioresi</td>
<td>16-18</td>
</tr>
</table>
</body>
</html>
```

ALLEGATO C

Validatori e Validazione di tabelle

WCAG		STANCA			esempio Codice	controlli
CHECK POINT WCAG	PRIORITA'	ENUNCIATO	ENUNCIATO	REQUISITO		
5.1	PRIORITA' 1	PER LE TABELLE IDENTIFICARE INTESTAZIONE RIGHE E COLONNE (TD CELLA / TH INTESTAZIONE)	NELLE TABELLE USARE MARCATORI PER IDENTIFICARE INTESTAZIONE RIGHE E COLONNE	9	<pre> <TR> <TH id="header1">Name</TH> <TH id="header2">Cups</TH> <TH id="header3" abbr="Type">Type of Coffee</TH> <TH id="header4">Sugar?</TH> </TR> <TR> <TD headers="header1">T. Sexton</TD> <TD headers="header2">10</TD> <TD headers="header3">Espresso</TD> <TD headers="header4">No</TD> </TR> </pre>	<p>verificare esistenza di tag <td> e di tag <th></p> <p>warning su controllo manuale che i <th> siano usati propriamente per le intestazioni (se esistenti)</p> <p>warning se non esistono <th> che segnala la mancanza di campi di intestazione della tabella (se non si trovano <th> nella tabella)</p>
					<pre> <CAPTION>Cups of coffee consumed by each senator</CAPTION> </pre>	<p>verificare esistenza del tag CAPTION per l'intestazione della tabella. Se non esiste Warning</p>

Validatori e Validazione di tabelle

<p>5.5</p>	<p>PRIORITA' 3</p>	<p>PER LE TABELLE FORNIRE SOMMARI</p>			<pre><TABLE number of senator, summary="This table charts the cups of coffee consumed by each the type of coffee (decaf or regular), and whether taken with sugar."></pre>	<p>verificare esistenza di attributo SUMMARY a tag <table> e restituire un warning che avvisi in caso di mancanza "Se è una tabella dati va inserito il sommario dei contenuti della tabella", in caso di presenza "Verifica che il sommario dei contenuti della tabella sia conforme ai dati in essa contenuta e che questa tabella non sia usata solo per fini di layout, in tal caso l'attributo Summary dovrebbe essere rimosso. Verifica che il sommario sia esplicativo --> inserire quindi un controllo su</p>
-------------------	------------------------	---	--	--	--	--

Validatori e Validazione di tabelle

						numero minimo di caratteri
5.6	PRIORITA' 3	PER LE TABELLE FORNIRE ABBREVIAZIONI PER LE INTESTAZIONI (TH ABBR)			<TH id="header3" abbr="Type">Type of Coffee</TH>	<p>verificare esistenza attributo ABBR= in TH, e dare warning su verifica del suo contenuto. Se non esiste suggerire di usarlo.</p> <p>verificare esistenza attributo ABBR= in TD, e dare errore in caso si trovi</p>

Validatori e Validazione di tabelle

<p>5.2</p>	<p>PRIORITA' 1</p>	<p>PER TABELLE CON PIU' LIVELLI LOGICI INTESTAZIONE RIGHE COLONNE, USARE MARCATORI PER ASSOCIARE DATI E INTESTAZIONE (THEAD TFOOT TBODY COL COLGROUP ATTRIBUTI "axis", "scope" e "headers" per descrivere relazioni più complesse fra i dati.</p>	<p>NELLE TABELLE CON DUE O PIU' LIVELLI DI INTESTAZIONE USARE GLI ATTRIBUTI DEL DTD NECESSARI PER ASSOCIARE CELLE DATI INTESTAZIONE</p>	<p>10</p>	<p>THEAD TBODY TFOOTER</p>	<pre> <THEAD> <TR> <TH>Code- Page
ID</TH> <TH>Name</TH> <TH>ACP</TH> <TH>OEMCP</TH> <TH>Windows
NT 3.1</TH> <TH>Windows
NT 3.51</TH> <TH>Windows
95</TH> </TR> </THEAD> <TBODY> <TR> <TD>1200</TD> <TD>Unicode (BMP of ISO/IEC-10646)</TD> <TD></TD> <TD></TD> <TD>X</TD> <TD>X</TD> <TD>*</TD> </TR> </pre>	<p>verificare esistenza THEAD, se questo è presente verificare esistenza TBODY altrimenti dare warning.</p> <hr/> <p>se si trova Thead deve esistere tbody e warning su verifica che effettivamente sia un'intestazione di tabella utile soprattutto se condivisa da più pagine successive</p>
-------------------	------------------------	---	---	-----------	------------------------------------	--	--

Validatori e Validazione di tabelle

						verificare esistenza tfoot, se esiste deve essere tbody altrimenti warning. Warning che indica che i contenuti nel tfoot sono utili quando condivisi da più pagine
--	--	--	--	--	--	---

Validatori e Validazione di tabelle

				<pre> <COLGROUP> <COLGROUP> <COLGROUP span="2"> <COLGROUP span="3"> <THEAD> <TR> <TH>Code- Page
ID</TH> <TH>Name</TH> <TH>ACP</TH> <TH>OEMCP</TH> <TH>Windows
NT 3.1</TH> <TH>Windows
NT 3.51</TH> <TH>Windows
95</TH> </TR> </THEAD> <TBODY> <TR> <TD>1200</TD> <TD>Unicode (BMP of ISO/IEC-10646)</TD> <TD></TD> <TD></TD> <TD>X</TD> <TD>X</TD> <TD>*</TD> </TR> </pre>	<p>verificare esistenza COLGROUP, se c'è warnig su controllo manuale che i dati siano effettivamente organizzati per colonne e i gruppi siano corretti</p>	
				<pre> <TR> <TH scope="col">Name</TH> <TH scope="col">Cups</TH> <TH scope="col" abbr="Type">Type of Coffee</TH> <TH scope="col">Sugar?</TH> </TR> </pre>	<p>verificare l'esistenza dell'attr. SCOPE all'interno dei TH.</p>	<p>Nota: se in una tabella dati non abbiamo SCOPE, HEADERS o AXIS bisogna dare</p>
					<p>Se esiste SCOPE non devono essere presenti gli</p>	

Validatori e Validazione di tabelle

				<pre> <TR> <TD>T. Sexton</TD> <TD>10</TD> <TD>Espresso</TD> <TD>No</TD> </TR> <TR> <TD>J. Dinnen</TD> <TD>5</TD> <TD>Decaf</TD> <TD>Yes</TD> </TR> </pre>	<p>attributi HEADERS e AXIS --> in tal caso restituire errore</p> <p>verificare se esiste SCOPE in td --> errore dovrebbero essere usati in th</p> <p>Nota: con SCOPE è possibile solo creare livelli logici molto di base, quindi è opportuno che dia warning in caso di tabelle annidate che usino scope e non headers</p>	<p>warning. Idem se ci sono più di questi, ovvero headers mischiati a scope. Mentre Axis può unirsi a SCOPE e Headers.</p>
				<p>HEADERS</p> <pre> <TR> <TH id="header1">Name</TH> <TH id="header2">Cups</TH> <TH id="header3" abbr="Type">Type of Coffee</TH> <TH id="header4">Sugar?</TH> </TR> <TR> <TD headers="header1">T. Sexton</TD> <TD headers="header2">10</TD> <TD headers="header3">Espresso</TD> </pre>	<p>Verificare che nei TH sia presente l'attributo ID.</p> <p>verificare che nei TD sia richiamato tramite l'attributo HEADERS uno degli ID definiti all'interno dei TH. Se vengono richiamati headers non esistenti dare errore.</p>	

Validatori e Validazione di tabelle

					<pre><TD headers="header4">No</TD> </TR></pre>	Se nei TH troviamo l'attributo HEADERS verificare che che richiami un headers già definito dall'ID di altri TH
--	--	--	--	--	--	--

Validatori e Validazione di tabelle

				<p>AXIS</p>	<pre> <TR> <TH></TH> <TH id="a2" axis="expenses">Meals</TH> <TH id="a3" axis="expenses">Hotels</TH> <TH id="a4" axis="expenses">Transport</TH> <TD>subtotals</TD> </TR> <TR> <TH id="a6" axis="location">San Jose</TH> <TH></TH> <TH></TH> <TH></TH> <TD></TD> </TR> <TR> <TD id="a7" axis="date">25-Aug- 97</TD> <TD headers="a6 a7 a2">37.74</TD> <TD headers="a6 a7 a3">112.00</TD> <TD headers="a6 a7 a4">45.00</TD> <TD></TD> </TR> <TR> <TD id="a8" axis="date">26-Aug- 97</TD> <TD headers="a6 a8 a2">27.28</TD> <TD headers="a6 a8 a3">112.00</TD> <TD headers="a6 a8 a4">45.00</TD> <TD></TD> </TR> <TR> <TD>subtotals</TD> <TD>65.02</TD> <TD>224.00</TD> </pre>	<p>nei TH possiamo trovare AXIS. Questo attributo può essere ripetuto per più TH con lo stesso valore. Sarà poi l'header assegnato ai TD ad associare oltre all'ID dell'intestazione anche il corretto axis. Verificare quindi esistenza di AXIS.</p>
						<p>AXIS può anche essere attributo dei TD in questo caso dare warning per controllo che la</p>

Validatori e Validazione di tabelle

					<pre> <TD>90.00</TD> <TD>379.02</TD> </TR> <TR> <TH id="a10" axis="location">Seattle</TH> <TH></TH> <TH></TH> <TH></TH> <TD></TD> </TR> <TR> <TD id="a11" axis="date">27- Aug-97</TD> <TD headers="a10 a11 a2">96.25</TD> <TD headers="a10 a11 a3">109.00</TD> <TD headers="a10 a11 a4">36.00</TD> <TD></TD> </TR> </pre>	<p>semantica sia giusta e che ha senso ai fini della categorizzazione delle celle</p>
5.3	PRIORITA' 2	<p>NON USARE TABELLE PER LAYOUT SE IL CONTENUTO NON E' COMPRENSIBILE LETTO IN FORMA LINEARIZZATA. IN CASO NON SIA COMPRENSIBILE FORNIRE UNA VERSIONE ALTERNATIVA</p>	<p>SE TABELLE USATE PER LAYOUT ACCERTARSI CHE CONTENUTO SIA COMPRENSIBILE SE TABELLA VIENE LINEARIZZATA E USARE ATTRIBUTI TABELLA SECONDO</p>	13		<p>verificare che insieme ad AXIS sia associato uno scope (ma in tal caso solo AXIS in TH e non in TD), oppure un ID richiamato da headers.</p>
						<p>warning ai TH: verificare che i TH non siano usati solo per centrare e bold ma che abbiano funzione di intestazione</p>

Validatori e Validazione di tabelle

<p>5.4</p>	<p>PRIORITA' 2</p>	<p>SE SI USA TABELLA PER LAYOUT NON USARE MARCATORI DI STRUTTURA PER DEFINIRE LO STILE (NON USARE TH PER METTERE CONTENUTO CENTRATO, MA DEFINIRE UNA CLASSE PER TD)</p>	<p>VALORE SEMANTICO</p>				<p>warning su controllo manuale tramite tablin che la versione linearizzata della tabella abbia senso. Da valutare se includere la funzione e mostrarlo in output</p>
							<p>verificare presenza di eventuali attributi DIR che indichino la direzione di lettura</p>

ALLEGATO D

Validatori e Validazione di tabelle

01_colspan.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>ColGroup</title>
</head>

<body>
<TABLE summary="Code page support in different versions
                of MS Windows.">
  <CAPTION>CODE-PAGE SUPPORT IN MICROSOFT WINDOWS</CAPTION>
  <COLGROUP>
  <COLGROUP>
  <COLGROUP span="2">
  <COLGROUP span="3">
  <THEAD>
    <TR>
      <TH>Code-Page<BR>ID</TH>
      <TH>Name</TH>
      <TH>ACP</TH>
      <TH>OEMCP</TH>
      <TH>Windows<BR>NT 3.1</TH>
      <TH>Windows<BR>NT 3.51</TH>
      <TH>Windows<BR>95</TH>
    </TR>
  </THEAD>
  <TBODY>
    <TR>
      <TD>1200</TD>
      <TD>Unicode (BMP of ISO/IEC-10646)</TD>
      <TD></TD>
      <TD></TD>
      <TD>X</TD>
      <TD>X</TD>
      <TD>*</TD>
    </TR>
    <TR>
      <TD>1250</TD>
      <TD>Windows 3.1 Eastern European</TD>
      <TD>X</TD>
      <TD></TD>
      <TD>X</TD>
      <TD>X</TD>
      <TD>X</TD>
    </TR>
    <TR>
      <TD>1251</TD>
      <TD>Windows 3.1 Cyrillic</TD>
      <TD>X</TD>
      <TD></TD>
      <TD>X</TD>
      <TD>X</TD>
      <TD>X</TD>
    </TR>
    <TR>
      <TD>1252</TD>
      <TD>Windows 3.1 US (ANSI)</TD>
      <TD>X</TD>
      <TD></TD>
      <TD>X</TD>
      <TD>X</TD>
      <TD>X</TD>
    </TR>
    <TR>
      <TD>1253</TD>
      <TD>Windows 3.1 Greek</TD>
      <TD>X</TD>
      <TD></TD>
      <TD></TD>
      <TD></TD>
      <TD></TD>
    </TR>
  </TBODY>
</TABLE>
```

Validatori e Validazione di tabelle

```
<TD>X</TD>
<TD>X</TD>
<TD>X</TD>
</TR>
<TR>
  <TD>1254</TD>
  <TD>Windows 3.1 Turkish</TD>
  <TD>X</TD>
  <TD></TD>
  <TD>X</TD>
  <TD>X</TD>
  <TD>X</TD>
</TR>
<TR>
  <TD>1255</TD>
  <TD>Hebrew</TD>
  <TD>X</TD>
  <TD></TD>
  <TD></TD>
  <TD></TD>
  <TD>X</TD>
</TR>
<TR>
  <TD>1256</TD>
  <TD>Arabic</TD>
  <TD>X</TD>
  <TD></TD>
  <TD></TD>
  <TD></TD>
  <TD>X</TD>
</TR>
<TR>
  <TD>1257</TD>
  <TD>Baltic</TD>
  <TD>X</TD>
  <TD></TD>
  <TD></TD>
  <TD></TD>
  <TD>X</TD>
</TR>
</TBODY>
</TABLE>
</body>
</html>
```

Validatori e Validazione di tabelle

02_tabella_accessibile_axisHeaders.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Tabella Accessibile Axis e Headers</title>
</head>

<body>
<TABLE summary="Dati hotel trasporti citta">
  <CAPTION>Travel Expense Report</CAPTION>
  <TR>
    <Td></Td>
    <TH id="header2" axis="expenses">Meals</TH>
    <TH id="header3" axis="expenses">Hotels</TH>
    <TH id="header4" axis="expenses">Transport</TH>
    <TD>subtotals</TD>
  </TR>
  <TR>
    <TH id="header6" axis="location" colspan="5">San Jose</TH>

  </TR>
  <TR>
    <TD id="header7" axis="date">25-Aug-97</TD>
    <TD headers="header6 header7 header2">37.74</TD>
    <TD headers="header6 header7 header3">112.00</TD>
    <TD headers="header6 header7 header4">45.00</TD>
    <TD></TD>
  </TR>
  <TR>
    <TD id="header8" axis="date">26-Aug-97</TD>
    <TD headers="header6 header8 header2">27.28</TD>
    <TD headers="header6 header8 header3">112.00</TD>
    <TD headers="header6 header8 header4">45.00 </TD>
    <TD></TD>
  </TR>
  <TR>
    <TD>subtotals </TD>
    <TD>65.02</TD>
    <TD>224.00</TD>
    <TD>90.00</TD>
    <TD>379.02</TD>
  </TR>
  <TR>
    <TH id="header10" axis="location" colspan="5">Seattle</TH>

  </TR>
  <TR>
    <TD id="header11" axis="date">27-Aug-97</TD>
    <TD headers="header10 header11 header2">96.25</TD>
    <TD headers="header10 header11 header3">109.00</TD>
    <TD headers="header10 header11 header4">36.00</TD>
    <TD></TD>
  </TR>
  <TR>
    <TD id="header12" axis="date">28-Aug-97</TD>
    <TD headers="header10 header12 header2">35.00</TD>
    <TD headers="header10 header12 header3">109.00</TD>
    <TD headers="header10 header12 header4">36.00 </TD>
    <TD></TD>
  </TR>
  <TR>
    <TD>subtotals</TD>
    <TD>131.25</TD>
    <TD>218.00</TD>
    <TD>72.00</TD>
    <TD>421.25</TD>
  </TR>
  <TR>
    <TH>Totals</TH>
    <TD>196.27</TD>
    <TD>442.00</TD>
    <TD>162.00</TD>
    <TD>800.27</TD>
  </TR>
</TABLE>
</body>
</html>
```

Validatori e Validazione di tabelle

03_tabelle_accessibili_headers.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Tabella Dati Accessibile</title>
</head>

<body>
<p>I dati che seguono sono stati tratti da <a
href="http://www.astrosurf.com/cosmoweb/sistemasolare/pianeti/tabella.html">Astrosurf</a> e
riassenblati in formato accessibile</p>
<table summary="Tabella comparativa dei pianeti, contenente dati relativi a massa, diametro,
densit&agrave;;, rotazione distanza dal sole e altro">
<caption>Tabella comparativa dei pianeti con dati in relazione a quelli della
terra</caption>
<thead>
<tr>
<th id="pianeta" abbr="pian" scope="row">Pianeta</th>
<th id="mercurio" abbr="mer" headers="pianeta">Mercurio</th>
<th id="venere" abbr="ven" headers="pianeta">Venere</th>
<th id="marte" abbr="mar" headers="pianeta">Marte</th>
<th id="giove" abbr="gio" headers="pianeta">Giove</th>
<th id="saturno" abbr="sat" headers="pianeta">Saturno</th>
<th id="urano" abbr="ura" headers="pianeta">Urano</th>
<th id="nettuno" abbr="net" headers="pianeta">Nettuno</th>
</tr>
</thead>
<tbody>
<tr>
<th scope="row" id="massa" abbr="massa">Massa</th>
<td headers="massa mercurio">0,0553</td>
<td headers="massa venere">0,815</td>
<td headers="massa marte">0,107</td>
<td headers="massa giove">317,8</td>
<td headers="massa saturno">95,2</td>
<td headers="massa urano">14,5</td>
<td headers="massa nettuno">17,1</td>
</tr>
<tr>
<th id="diametro" abbr="dia" scope="row">Diametro</th>
<td headers="diametro mercurio">0,383</td>
<td headers="diametro venere">0,949</td>
<td headers="diametro marte">0,533</td>
<td headers="diametro giove">11,21</td>
<td headers="diametro saturno">9,45</td>
<td headers="diametro urano">4,01</td>
<td headers="diametro nettuno">3,88</td>
</tr>
<tr>
<th id="densita" abbr="dens" scope="row">Densit&agrave;;</th>
<td headers="densita mercurio">0,984</td>
<td headers="densita venere">0,951</td>
<td headers="densita marte">0,713</td>
<td headers="densita giove">0,240</td>
<td headers="densita saturno">0,125</td>
<td headers="densita urano">0,230</td>
<td headers="densita nettuno">0,297</td>
</tr>
<tr>
<th id="gravita" abbr="grav" >Gravit&agrave;;</th>
<td headers="gravita mercurio">0,378</td>
<td headers="gravita venere">0,907</td>
<td headers="gravita marte">0,377</td>
<td headers="gravita giove">2,36</td>
<td headers="gravita saturno">0,916</td>
<td headers="gravita urano">0,889</td>
<td headers="gravita nettuno">1,12</td>
</tr>
</tbody>
</table>

</body>
</html>
```

04_tabelle_NONaccessibili_td.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Tabella Dati NON Accessibile</title>
</head>

<body>
<p>I dati che seguono sono stati tratti da <a
href="http://www.astrosurf.com/cosmoweb/sistemasolare/pianeti/tabella.html">Astrosurf</a> e
riassenblati in formato accessibile</p>
Tabella comparativa dei pianeti con dati in relazione a quelli della terra
<table>

  <tr>
    <td>Pianeta</td>
    <td>Mercurio</td>
    <td>Venere</td>
    <td>Marte</td>
    <td>Giove</td>
    <td>Saturno</td>
    <td>Urano</td>
    <td>Nettuno</td>
  </tr>
  <tr>
    <td>Massa</td>
    <td>0,0553</td>
    <td>0,815</td>
    <td>0,107</td>
    <td>317,8</td>
    <td>95,2</td>
    <td>14,5</td>
    <td>17,1</td>
  </tr>
  <tr>
    <td>Diametro</td>
    <td>0,383</td>
    <td>0,949</td>
    <td>0,533</td>
    <td>11,21</td>
    <td>9,45</td>
    <td>4,01</td>
    <td>3,88</td>
  </tr>
  <tr>
    <td>Densit&agrave;</td>
    <td>0,984</td>
    <td>0,951</td>
    <td>0,713</td>
    <td>0,240</td>
    <td>0,125</td>
    <td>0,230</td>
    <td>0,297</td>
  </tr>
  <tr>
    <td>Gravit&agrave;</td>
    <td>0,378</td>
    <td>0,907</td>
    <td>0,377</td>
    <td>2,36</td>
    <td>0,916</td>
    <td>0,889</td>
    <td>1,12</td>
  </tr>
</table>

</body>
</html>

```

Validatori e Validazione di tabelle

06_tableLayoutLinear2.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Layout linearizzato</title>
<style>
body { font-family: verdana; font-size:1em;}
.menu {width:15%; background-color:#FFFCC; border: 1px solid #336600; text-
align:center; }
.menuitem { width:100%; background-color:#FFFCC; border-top: 1px solid #336600;
text-align:center; }
.central { width:83%; background-color:#CCFF33; padding:1em;}
.header { width:100%; background-color:#339966; color:#FFFFFF; font-size:3em; text-
align:center;}
.footer { width:100%; background-color:#339966; color:#FFFFFF; font-size:0.6em; text-
align:center;}
</style>
</head>
<body>
<table>
<thead>
<tr>
<td id="Favoliamo" class="header" colspan="2">Favoliamo</td>
</tr>
</thead>
<tbody>
<tr>
<td class="menu">
<table class="menuitem">
<tr>
<th id="menu">Menu'</th>
</tr>
<tr>
<td headers="menu" class="menuitem">Home</td>
</tr>
<tr>
<td headers="menu" class="menuitem">
Favole
</td>
</tr>
<tr>
<td headers="menu" class="menuitem">
Staff
</td>
</tr>
<tr>
<td headers="menu" class="menuitem">
Info
</td>
</tr>
</table>
</td>
<td class="central" id="content">
Il vento Borea, un vento di tramontana freddo e impetuoso, e il sole
giorno
litigavano per stabilire chi dei due fosse il pi&ugrave; forte. Un
si misero d&rsquo;accordo: avrebbero considerato vincitore chi fosse
riuscito a togliere di dosso i vestiti a un viandante. Cominci&ograve;
Borea che inizi&ograve; a soffiare forte. L&rsquo;uomo si stringeva
maggiore.
i vestiti e il vento si metteva a soffiare con violenza ancora
Allora il viandante, che era sempre pi&ugrave; tormentato dal freddo,
aggiunse a ci&ograve; che aveva addosso un secondo mantello. Borea,
stanchissimo, lasci&ograve; il suo uomo al sole. Questi dapprima
inizi&ograve;
a splendere con moderazione, senza esagerare; poi, quando l&rsquo;uomo
sprigion&ograve;
si tolse il mantello che aveva aggiunto agli altri vestiti,
un calore sempre pi&ugrave; forte. A questo punto, l&rsquo;uomo non
potendo pi&ugrave; resistere, si spogli&ograve; nudo e and&ograve; a
fare il bagno nel fiume che scorreva l&igrave; vicino.<br />
La favola ci fa capire che la persuasione &egrave; spesso pi&ugrave;
efficace della violenza.
</td>
</tr>
</tbody>
<tfoot>
<tr>
<td class="footer" id="footer" colspan="2">&copy; Favoliamo 2008</td>
</tr>
</tfoot>
</table>
</body>
</html>
```

Validatori e Validazione di tabelle

07_tableLayoutLinear.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Layout linearizzato</title>
<style>
body { font-family: verdana; font-size:1em;}
.menu {width:15%; background-color:#FFFCC; border: 1px solid #336600; text-
align:center; }
.menuitem { width:100%; background-color:#FFFCC; border-top: 1px solid #336600;
text-align:center; }
.central { width:83%; background-color:#CCFF33; padding:1em;}
.header { width:100%; background-color:#339966; color:#FFFFFF; font-size:3em; text-
align:center;}
.footer { width:100%; background-color:#339966; color:#FFFFFF; font-size:0.6em; text-
align:center;}
</style>
</head>

<body>
<table>

<tr>
<td id="Favoliamo" class="header" colspan="2">Favoliamo</td>
</tr>
<tr>
<td class="menu">
<table class="menuitem">
<tr>
<td id="menu"></td>
</tr>
<tr>
<td headers="menu" class="menuitem">Home</td>
</tr>
<tr>
<td headers="menu" class="menuitem">
Favole
</td>
</tr>
<tr>
<td headers="menu" class="menuitem">
Staff
</td>
</tr>
<tr>
<td headers="menu" class="menuitem">
Info
</td>
</tr>
</table>
</td>
<td class="central" id="content">
Il vento Borea, un vento di tramontana freddo e impetuoso, e il sole
litigavano per stabilire chi dei due fosse il pi&ugrave; forte. Un
giorno
si misero d&rsquo;accordo: avrebbero considerato vincitore chi fosse
riuscito a togliere di dosso i vestiti a un viandante. Cominci&ograve;
Borea che inizi&ograve; a soffiare forte. L&rsquo;uomo si stringeva
maggiore.
i vestiti e il vento si metteva a soffiare con violenza ancora
Allora il viandante, che era sempre pi&ugrave; tormentato dal freddo,
aggiunse a ci&ograve; che aveva addosso un secondo mantello. Borea,
stanchissimo, lasci&ograve; il suo uomo al sole. Questi dapprima
inizi&ograve;
a splendere con moderazione, senza esagerare; poi, quando l&rsquo;uomo
sprigion&ograve;
si tolse il mantello che aveva aggiunto agli altri vestiti,
un calore sempre pi&ugrave; forte. A questo punto, l&rsquo;uomo non
potendo pi&ugrave; resistere, si spogli&ograve; nudo e and&ograve; a
fare il bagno nel fiume che scorreva l&igrave; vicino.<br />
La favola ci fa capire che la persuasione &egrave; spesso pi&ugrave;
efficace della violenza.
</td>
</tr>
<tr>
<td class="footer" id="footer" colspan="2">&copy; Favoliamo 2008</td>
</tr>
</table>
</body>
</html>
```

Validatori e Validazione di tabelle

08_tableLayoutNoLinear.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Layout non linearizzato</title>
<style>
body { font-family: verdana; font-size:1em;}
.menu {width:15%; background-color:#FFFFCC; border: 1px solid #336600; padding:5px;
}
.central { width:83%; background-color:#CCFF33; padding:5px;}
</style>
</head>
<body>
<table>
<tr>
<td class="menu">
Home
</td>
<td class="central" rowspan="4">Il vento Borea, un vento di tramontana freddo
e impetuoso, e il sole litigavano per stabilire chi dei due fosse il pi&ugrave; forte.
Un giorno si misero d&rsquo;accordo: avrebbero considerato vincitore chi fosse
riuscito a togliere di dosso i vestiti a un viandante. Cominci&ograve; Borea che
inizi&ograve; a soffiare forte. L&rsquo;uomo si stringeva i vestiti e il vento si
metteva a soffiare con violenza ancora maggiore. Allora il viandante, che era sempre
pi&ugrave; tormentato dal freddo, aggiunse a ci&ograve; che aveva addosso un secondo
mantello. Borea, stanchissimo, lasci&ograve; il suo uomo al sole. Questi dapprima
inizi&ograve; a splendere con moderazione, senza esagerare; poi, quando l&rsquo;uomo
si tolse il mantello che aveva aggiunto agli altri vestiti, sprigion&ograve; un calore
sempre pi&ugrave; forte. A questo punto, l&rsquo;uomo non potendo pi&ugrave; resistere,
si spogli&ograve; nudo e and&ograve; a fare il bagno nel fiume che scorreva l&igrave;
vicino.<br />
La favola ci fa capire che la persuasione &egrave; spesso pi&ugrave; efficace
della violenza.
</td>
</tr>
<tr>
<td class="menu">
Info
</td>
</tr>
<tr>
<td class="menu">
Staff
</td>
</tr>
<tr>
<td class="menu">
dove siamo
</td>
</tr>
</table>
</body>
</html>
```