

Skip navigation: in nome dell'accessibilità

Di Christian Fusi

Premessa

Nella maggioranza dei casi le pagine web vengono viste tramite i più diffusi browser grafici e su uno schermo di computer, ma non è sempre così. Esistono categorie di utenti che navigano grazie a dispositivi "assistivi", usano cioè tecnologie e strumenti studiati per ovviare a disabilità fisiche che renderebbero loro impossibile utilizzare un computer. Tra questi vanno annoverati coloro che soffrono di cecità o disturbi visivi e che usano uno screen reader, tipo [JAWS](#), per poter utilizzare il computer oppure un browser vocale, tipo [Home Page Reader](#) di IBM, per navigare nel web*. Questi software "leggono ad alta voce" tramite la scheda audio del PC ciò che lo schermo mostra. Per farlo devono prima compiere una operazione detta **linearizzazione**. Con quest'azione il contenuto di un sito web viene tradotto in una sequenza di parole, lette poi una di seguito all'altra. Le immagini vengono sostituite con il testo specificato nell'attributo ALT; il testo e le eventuali tabelle sono scomposte e riordinate da sinistra verso destra riga per riga.

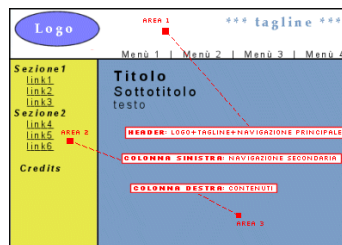
I problemi

Vista la premessa è facile capire alcuni dei grossi problemi che nascono in questa situazione per colpa di una scarsa attenzione all'accessibilità, tra cui:

- 1) Mancanza di testo alternativo alle immagini
- 2) Errata linearizzazione o difficile comprensione di una tabella dati**
- 3) Errata linearizzazione dei contenuti se il layout del sito è ottenuto tramite tabelle

E' proprio il terzo caso quello di cui ci occuperemo. Se la pagina viene letta da sinistra a destra, riga per riga, si rischia che prima di giungere ai contenuti del sito, l'utente debba "ascoltarsi" tutto il contenuto dell'header: il testo posto in alternativa al logo, la tagline, i link della navigazione primaria, e poi anche il contenuto dell'eventuale colonna di sinistra come i link della navigazione secondaria e tutto il resto. L'utente è così obbligato a pazientare e, anche nel caso potesse muoversi nel sito grazie alle opzioni fornite dal software (saltare tramite la tastiera tra un "blocco" e l'altro), difficilmente capirebbe come è organizzato non potendo avere la "visione d'insieme" di un utente normovedente. Ne resta mortificato e la sua esperienza d'uso del nostro sito sarà quanto meno negativa.

Un esempio



Il layout di un sito, come il comune "tre colonne" dell'immagine sopra, può essere attualmente realizzato in due modi: tramite CSS o con le tabelle.

Nel primo caso, potendo sfruttare i DIV e il posizionamento via CSS2, la visualizzazione non dipende da come il codice html è ordinato, ma dalle istruzioni fornite tramite CSS. Quindi possiamo tranquillamente scrivere nell'HTML per primo ciò che vogliamo sia letto prima: il <DIV ID="contenuto">...</DIV> (in fig. l'area 3) prima del <DIV ID="navigazione">...</DIV> (area 2). Nel secondo caso, ovvero se il layout è ottenuto utilizzando le tabelle, l'ordine di lettura dello screen reader seguirà quello espresso dal codice che a sua volta determina quello che si vede a schermo. Quindi verrà letto:

Logo - Tagline - Menù1 - Menù2 - Menù3 - Menù4 - Sezione1 - Link1 - Link2 - Link3 - Sezione2 - Link4 - Link5 - Link6 - Credits - **Titolo - Sottotitolo - Testo**

Potremmo fare in modo di avere i contenuti subito dopo l'header, magari invertendo tra loro le colonne sinistra e destra così da portare la navigazione locale a destra (quindi "dopo" i contenuti), ma questa è una scelta di design e deve essere compiuta, se ritenuta adeguata al progetto, nella fase iniziale di stesura della proposta grafica.

Possiamo invece sempre usare un semplice stratagemma: la **Skip Navigation**.

La tecnica

Porremo un link in cima al sito che ci porterà all'interno della pagina direttamente nel punto dove iniziano i contenuti, saltando così l'area di navigazione. Per far ciò basterà creare una semplice ancora. Poi con i CSS daremo istruzione per nascondere questo link ai browser visuali mentre gli screen reader e i browser vocali, che ignorano i CSS***, leggeranno il link per primo permettendo così di sfruttarlo.

Ecco il codice HTML per il link da mettere subito dopo <body>:

```
<DIV CLASS="nascosto"><A HREF="#contenuto" TITLE="Salta al contenuto"> Skip  
Navigation </A></DIV>
```

....

e l'ancora da mettere nel punto in cui iniziano i contenuti:

```
<A NAME="contenuto" ID="contenuto"></A>
```

....

N.B.: ID va aggiunto a NAME solo in XHTML 1.0; in XHTML 1.1 va invece tolto NAME.

Infine creiamo nel CSS la classe ".nascosto" per rendere invisibile il link:

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

E' molto semplice, no?

Un appunto finale può essere fatto citando le osservazioni che [Zeldman fa nel suo ultimo libro](#). La skip navigation è utile anche ad utenti che usano un browser non CSS (magari su un PDA o un telefonino) ed a utenti con disabilità motorie (ma non visive) che usano la tastiera per saltare tra la pagina. Se però quest'ultimi usano un browser CSS compatibile, il link gli viene nascosto. Una soluzione potrebbe essere quella di associare una accesskey al link "Skip Navigation", ma vi scontrerete con i [problemi](#) dovuti al loro uso. Se lo fate vi consiglio almeno di usare un numero e non una lettera (es. accesskey="1").

Vista la semplicità del codice e il notevole vantaggio che questa tecnica apporta, consiglio fortemente il suo utilizzo in ogni sito attento all'accessibilità. Le persone con disturbi visivi sono fedeli utenti/clienti dei siti che pensano anche a loro.

Note:

* Citiamo tra gli screenreader più diffusi anche [Windows-Eyes](#). Esistono pure dispositivi hardware per facilitare i non vedenti, come i display braille, che però sono piuttosto costosi.

** Si può venire incontro agli utenti anche in questo caso. Ne parleremo in un prossimo articolo.

*** Jaws, per quanto scrive [Maurizio Boscarol](#), ha un comportamento "bizzarro" con i fogli di stile, sembra però che se il CSS viene importato (e non linkato) si hanno più possibilità che venga correttamente ignorato (ma varia da versione a versione).

Da aggiungere che si possono scrivere fogli di stile "uditivi" per i browser vocali, con funzionalità appositamente previste nei CSS2. Sono richiamabili linkandoli nell'HTML e indicando come media type "AURAL". Purtroppo al momento nessun browser li supporta.

Qualche link:

- [Affrontare l'accessibilità: giorno 11](#)
http://www.francocarcillo.it/dive/day_11.html
- [Diodati.org](#) forum e molta documentazione utile
<http://www.diodati.org/>
- [Browser vocali e screenreader](#) su Stradeblu
<http://www.stradeblu.it/software.htm>
- [Lynx Viewer](#) per simulare il famoso browser testuale
<http://www.delorie.com/web/lynxview.html>
- [Corso accessibilità web](#) di Fucinaweb
http://www.fucinaweb.com/usabilita/access_programma.asp
- [ITList.org](#) mailing list di webaccessibile.org
<http://itlists.org/mailman/listinfo/webaccessibile>

i-use.it Web design accessibile e usabile

www.i-use.it

Verso un web accessibile e usabile.