

Perchè creare pagine web secondo gli standard?

Di Christian Fusi

Estratto

Usare codice valido, secondo le raccomandazioni W3C, permette di ridurre i tempi ed i costi di realizzazione e di manutenzione. Inoltre, le pagine così realizzate hanno una vita più lunga: scopriamo perché è importante appoggiarsi agli standard del web.

Sommario

- [Usare codice valido](#)
- [Dividere contenuto e forma](#)
- [Scrivere codice Xhtml](#)
- [Concludendo](#)
- [Risorse](#)

Usare codice valido

Usare codice valido secondo le raccomandazioni W3C significa adottare uno standard*.

Questo ci permette ora e nel prossimo futuro di avere, se non la certezza, almeno il maggior grado di probabilità che la nostra pagina sarà visualizzata senza problemi. Ciò significa nella pratica ridurre i tempi ed i costi di realizzazione e di manutenzione ed inoltre dotare le nostre pagine di "una vita più lunga". Allo stesso tempo questa scelta (unita ad una buona progettazione) comporta anche giovamenti all'accessibilità del nostro sito rendendo le pagine visibili da tutti e con diversi tipi di browser o device. Infine l'uso di strumenti di validazione offre un aiuto ai webmaster nel correggere eventuali errori ed è una garanzia per il committente di ricevere un lavoro ben fatto sotto l'aspetto della "pulizia del codice".

Naturalmente ci sono ancora problemi di supporto alle specifiche da parte delle differenti versioni dei browser, ma le case produttrici (di browser come quelle di editor html) stanno facendo maggiore attenzione e la via sembra tracciata. Inoltre è da sottolineare come ogni progetto web debba essere il risultato di una ponderata valutazione tra le possibilità tecniche e le necessità dell'utente/committente.

Non trovo invece valida l'obiezione di una maggior difficoltà di utilizzo ed apprendimento di queste tecniche: codice corretto e buona conoscenza dei CSS sono allo stato attuale delle cose, bagaglio imprescindibile di un professionista del settore.

Dividere contenuto e forma

All'inizio l'HTML è stato progettato per visualizzare tramite un browser semplici documenti. I pochi tag mutuati dall'SGML (il linguaggio da cui deriva, molto complicato e questo sì standard ISO)

dovevano rendere il creare documenti un'operazione semplice per tutti. Il web nella mente del suo creatore doveva essere in percentuali pari visualizzare documenti e crearne di nuovi.

L'evolversi dei browser e l'aumentare dell'attenzione (soprattutto economica) verso il web ha portato i creatori di documenti a cimentarsi nel renderli maggiormente "attraenti". Ben presto i creativi, provenienti dalla grafica su carta e quindi fedeli al principio del "totale controllo sul risultato", hanno cercato i modi di sfruttare per le loro necessità ciò che il nuovo mezzo dava a disposizione. In questa direzione si è iniziato ad esempio con l'impaginare tramite tabelle annidate. Allo stesso tempo la "guerra dei browser" ha fatto sì che Netscape e Explorer inventassero tecnologie proprietarie adottate solo dal loro software (così ad esempio nacquero i Frame). In tale situazione, per anni spesso e volentieri, i webdesigner sono stati costretti a creare duplici versioni dei loro siti.

Oggi le cose sono cambiate, e in meglio. Il ragionamento migliore che si dovrebbe fare in un progetto web ritengo sia questo: **voglio che il mio sito sia accessibile da chiunque con qualunque browser/dispositivo. Non mi importa se visualizzato non sarà perfettamente identico, l'utente può ben rinunciare alla formattazione ma non gli si deve MAI impedire di accedere ai contenuti.**

Ciò significa cambiare mentalità: **riconoscere che il contenuto e il modo in cui questo è rappresentato siano due aspetti differenti del progetto.** Quindi occorre distinguerli e per farlo abbiamo a disposizione due linguaggi differenti: HTML e CSS che, interagendo tra loro, producono il risultato finale. Ciò rende più agevole la manutenzione del sito (spesso maggiormente onerosa della stessa realizzazione), più facile la lettura del codice, più semplice un restyling, più veloci le nostre pagine da scaricare e più attento il sito verso l'utente finale (fornendo al designer maggiori strumenti per ottenere accessibilità, usabilità, personalizzazione).

Scrivere in XHTML

L'XHTML è la riformulazione in XML del linguaggio HTML, dunque è una ridefinizione di un linguaggio già esistente (e che vi ricordo non vedrà più nuove versioni dopo la 4.01). Un suo miglioramento (mantenendolo compatibile) che permette di fare da ponte verso un futuro di solo utilizzo dell'XML. Senza grossi cambiamenti e costi, iniziamo a prendere confidenza con concetti come il rigore sintattico, la portabilità, la possibilità di elaborazione, la semantica e la modularità, che saranno il pregio del futuro linguaggio della rete. Inoltre, essendo nato per definire specificatamente la struttura di un documento, obbliga a separare contenuti e formattazione con i vantaggi descritti sopra. Così i contenuti, che dovrebbero rappresentare la vera ricchezza per un sito (ma anche il maggior costo-investimento) possono ad esempio, caricando css differenti, essere visti su un PC ma anche su cellulari evoluti, su PDA e così via.

Concludendo:

Sebbene siano ancora poche le pagine validate (e invece secondo un famoso articolo di Jeffrey Zeldman il [99,9%, del web risulta obsoleto](#)), credo però che dopo dieci anni il web (o meglio una parte di esso) debba entrare in una fase "adulta", prepararsi al suo futuro che, senza attenzioni anche

da parte di chi crea le pagine, non potrà certo essere ancor più felice.

Note:

* Il W3C non emette standard ISO (prerogativa della International Standard Organization) ma "raccomandazioni". Queste, essendo stabilite dalle aziende e dagli enti (raccolti appunto nel Consorzio) che materialmente contribuiscono e sovrintendono all'evolversi del web, sono "di fatto" uno standard riconosciuto.

Risorse:

- [XHTML 1.0](#) sul W3C
- [Il validatore del W3C](#)
- [Guida a XHTML](#) su www.html.it, di C.Lamanna
- [Web Standard Project](#) il manifesto
- [A List Apart](#) [*webdesign*]
- [Meyer Web](#) [*CSS*]
- [Linee Guida per l'Accesibilità - WAI](#)