

E se qualcuno dicesse che i tuoi documenti XHTML non sono altro che HTML non valido?



Se sei uno **sviluppatore web**, avrai sicuramente sentito parlare di XHTML, la "nuova" versione di HTML.

E avrai sicuramente assistito a questa corsa agli standard: i sintomi sono piccoli badges in fondo ai siti di tutte le fogge e le dimensioni, che recano scritte del tipo "XHTML VALID", "I LOVE VALIDATOR", "XHTML IS MY LIFE" insomma tutto un genere di frasi sdolcinate che si dovrebbero dedicare alla propria ragazza e non ad un linguaggio di markup!

E se qualcuno un giorno ti dicesse che tutti i **tui documenti XHTML non sono altro che HTML non valido?**

Tutta una questione di content type

Infatti sotto l'aspetto tecnico è proprio così. Per spiegare il motivo di questo "raggiro", bisogna considerare in che modo i browser distinguono i vari tipi di file che prelevano dai siti web.

Ogni volta che un file viene scaricato, ad esso viene allegato una descrizione, detta "**content type**", la quale specifica di che tipo di file si tratta: per una foto in formato "jpg" sarà "image/jpeg", per un file in formato "css" sarà "text/css" e così via. **Il content type per i documenti XHTML è "application/xhtml+xml"**, che viene assegnato ai file con estensione .xhtml, mentre per i file .html il content type è "text/html"

Dunque tutti i documenti scritti seguendo le specifiche XHTML, ma salvati con estensione .html vengono automaticamente interpretati dai server come file HTML e dunque visualizzati come tali.

La ragione principale per usare un content type diverso è che **XHTML è un derivato di XML**, e quindi è soggetto ad una **gestione degli errori molto più rigida**, garantendo una maggiore qualità dei documenti: è ragionevole allora indicare la differenza ai browser. Inoltre, XHTML presenta alcune differenze di sintassi con HTML, una su tutte il fatto che i tag vuoti come

devono essere obbligatoriamente chiusi aggiungendo uno slash alla fine

Chi ha ucciso Mr. X?

Purtroppo molti browser (ad esempio Internet Explorer fino alla versione 8) non sono ancora in grado di **rendere i file XHTML con il giusto content type**, e dunque lo trasformano in "vecchio" **HTML non valido** perché contiene degli elementi che non fanno parte di questa specifica come abbiamo detto prima, ma dato che il parser HTML non è progettato per arrestarsi in presenza di un errore, rende la pagina come dovuto.

Il momento che attendevi: l'esempio!

Lasciamo stare per un attimo tutti questi paroloni che tanto piacciono ai tecnici: un esempio vale più di mille parole:

Nota: se non abbiamo installato [Firefox 3](#) è il momento giusto per farlo.

- Creiamo un file con il nostro editor preferito e chiamiamolo standard.html.
- Duplichiamo questo file, rinominandolo standard.xhtml.
- Inseriamo in entrambi i file il codice seguente, in cui includiamo volontariamente un errore di distrazione che potrebbe capitare a chiunque, ovvero dimenticare di chiudere un tag, in questo caso abbiamo dimenticato di chiudere il tag html alla fine del documento.

Sei pronto? Visualizza entrambi i file in Firefox, ed osserva come si comporta il browser. Nel caso del file con estensione html la pagina verrà visualizzata in tutta la sua standard bellezza:

I LOVE XHTML

- Un elemento della mia lista standard
- Idem come sopra

Il mio paragrafo standard

mentre nel caso dell'xhtml ti sarai trovato di fronte ad un messaggio d'errore non molto

rassicurante:

XML Parsing Error: unclosed token

Location: file:///home/just/Documenti/YIW/doctype/standard.xhtml

Line Number 15, Column 1:

```
</html  
^
```

HTML è morto...evviva HTML!

Gli autori che intendono rendere pubblico il loro lavoro dovrebbero continuare ad usare HTML 4.01 - [Ian Hickson](#) (membro del W3C ed editor per HTML5)

Dunque non ha nessun senso utilizzare **XHTML e servirlo come text/html**, perché il nostro codice, verrà trattato come **HTML non valido**. Molti sviluppatori hanno promosso tecniche (come la [Content Negotiation](#)) per servire il giusto content type ai browser che lo supportano, ma quest'ultime si basano su caratteristiche particolari dei browser che in futuro potrebbero cambiare, e quindi non è affatto il giusto approccio al problema.

In più, se anche fosse possibile utilizzare XHTML correttamente, sarebbe molto complicato, per un aspirante sviluppatore, accettare contenuto da parte degli utenti (ad esempio dei commenti in un blog), poichè ogni piccolo errore di sintassi rischierebbe di compromettere l'accessibilità dell'intero sito.

Quindi, dopo l'ondata di entusiasmo provocata da XHTML, [molte voci](#) autorevoli, stanno facendo un passo indietro, nell'attesa di un maggior supporto da parte dei browser, o del termine dello sviluppo della nuova versione di HTML.

In conclusione

- Se non si hanno delle buone motivazioni per servire le pagine in XHTML, ad esempio per documenti sviluppati in MathML (un linguaggio per formule matematiche) o SVG (un linguaggio per la gestione di immagini), **sarebbe meglio utilizzare il Doctype 4.01 Strict**

per i propri documenti; sarebbe valido e non meno standard della sua versione XHTML:

- Se si volesse comunque provare a servire il giusto content type ai browser che lo supportano, si potrebbe provare ad usare la Content Negotiation, ma questo implicherebbe l'uso di tecnologie lato server, o la modifica di alcune impostazioni del server, operazione non molto semplice, senza tuttavia apportare alcun beneficio di XML ai nostri siti.
- C'è una terza via, che sarebbe quella di utilizzare il **doctype di HTML5**, per cominciare a prendere confidenza con questo nuovo strumento. Faccio notare che questo dovrebbe essere fatto solo per siti personali o sperimentali, in quanto il supporto non è per nulla garantito dai vari browser. Approfondiremo il discorso in una serie di articoli dedicati proprio ad HTML5 ed alla ventata di novità che porterà con sé.