



## XHTML 2: le novità

**DARIO DE JUDICIBUS**

La nuova versione dell'XHTML presenta una serie di novità che, se da una parte semplificano il linguaggio, dall'altra lo rendono ancora più flessibile e potente, prime fra tutte l'eliminazione di fatto degli elementi `<a>` e `<img>`.

Per molti «sito web» è ancora sinonimo di HTML. In effetti sono ancora molti i siti che utilizzano questo linguaggio, soprattutto quelli amatoriali, anche se sempre più sviluppatori si sono ormai convertiti all'utilizzo dei fogli di stile e dei siti *frameless*, facendo così proprio il principio che stabilisce che le specifiche che descrivono la struttura di un documento e quelle che ne descrivono la sua rappresentazione rimangano ben separate. Malgrado fosse già possibile garantire tale separazione con l'HTML 4.01 e il CSS 1, questo importante principio ha trovato la sua piena realizzazione nel linguaggio XHTML, la controparte XML del buon vecchio HTML. Sebbene i due linguaggi possano sembrare molto simili, essi si rifanno infatti a due modi di pensare molto differenti. L'HTML era «il linguaggio», ovvero con esso si faceva tutto: struttura, aspetto, interazione, ogni elemento era presente in una sintassi semplice e facile da imparare. Con l'XML tutto questo è cambiato. L'XHTML si è concentrato fondamentalmente sulla struttura logica del documento, delegando completamente ai fogli di stile le questioni riguardanti la sua rappresentazione, anche se qualcuno parla di un «ritorno alle origini», dato che il primo HTML sembra fosse stato disegnato con gli stessi criteri dell'XHTML all'inizio. Sebbene questo approccio abbia portato a una maggiore complessità e creato qualche problema a chi era abituato a realizzare pagine web in modo artigianale, soprattutto per quello che riguarda la definizione dei fogli di stile, la nuova architettura ha dato ai professionisti della rete una serie di strumenti più potenti e rigorosi che hanno permesso di industrializzare il processo di sviluppo.

Con l'XHTML 2 questo orientamento si è ulteriormente accentuato. Arrivate alla settima stesura con il *draft* pubblicato dal W3C il 27 maggio 2005, le specifiche dell'XHTML 2 hanno raggiunto un tale grado di stabilità da ritenere opportuno anticipare qui le novità di questa versione che si presenta estremamente interessante.

Prima di entrare nel merito dei singoli elementi, tuttavia, vediamo quali sono stati i criteri utilizzati dagli sviluppatori delle specifiche nel definire questa nuova versione di un linguaggio ormai utilizzato in tutti i siti professionali.

Innanzitutto si è cercato di rendere ancora più coerente il linguaggio all'interno dell'architettura XML. Questo vuol dire evitare qualsiasi duplicazione nei confronti di altri linguaggi XML eliminando tutti quegli elementi che fossero già presenti altrove e la cui esistenza era dovuta esclusivamente a questioni di compatibilità con il vecchio HTML. Inoltre si è deciso di scorporare dall'XHTML la componente relativa alle *frame*, dato che questa tecnica aveva dimostrato una serie di limiti che aveva portato diversi sviluppatori a concentrarsi piuttosto sulle divisioni (**div**), le cui proprietà venivano poi definite nei fogli di stile. In realtà le *frame* non sono scomparse, ma hanno dato origine a un nuovo modulo denominato XFrames e del quale parleremo in un prossimo articolo. Si è ritenuto infatti opportuno salvaguardare il concetto di *frame*, che solo in parte può essere simulato attraverso le divisioni, cercando al contempo di eliminarne i difetti, anche perché le tecniche che utilizzano le **div** sono particolarmente vantaggiose là dove ci si può avvantaggiare di un linguaggio sul lato *server*, ma presentano alcuni limiti realizzativi se si opera solo sul lato *client*, quali ad esempio la duplicazione di tutta una serie di blocchi di codice in pagine differenti.

Come già detto, si è accentuata la separazione fra struttura e presentazione aggiungendo nuovi elementi specificatamente intesi a strutturare meglio il documento, riducendo conseguentemente tutto ciò che poteva essere definito tramite i fogli di stile. Questo ha reso inoltre il linguaggio molto più facile da scrivere e da leggere. Naturalmente c'è un prezzo da pagare che risiede nella maggiore complessità dei fogli di stile, complessità comunque gestibile grazie all'utilizzo di opportune applicazioni pensate per disegnare e generare questo tipo di sorgenti.

Una maggiore attenzione è anche stata data agli aspetti relativi all'internazionalizzazione, aspetti che abbiamo visto essere preminenti nella versione 3 del CSS. Parimenti si è cercato di rendere sempre più la struttura di un documento indipendente dalla periferica utilizzata per renderlo fruibile, aspetto anch'esso delegato a questo punto ai fogli di stile. Si è anche cercato di ridurre al massimo la necessità di utilizzare all'interno delle pagine web programmi di *script*, dato che questo contrasta in parte con il tentativo di rendere il file che descrive la struttura del documento quanto più possibile indipendente dalla periferica. Ovviamente il Javascript continuerà ad avere ancora una sua importanza, soprattutto se non si può operare sul lato server con l'ASP, il Perl, il PHP o altri linguaggi. Tuttavia è evidente che un uso eccessivo di questo linguaggio può rendere un file XHTML troppo legato a determinate assunzioni sulla sua rappresentazione e vanificare così il tentativo di separare struttura e riproduzione.

E veniamo al linguaggio vero e proprio. Assumeremo qui una conoscenza di base dell'HTML e quindi anche dell'XHTML versione 1 per concentrarci piuttosto sulle differenze

fra quest'ultima e la nuova versione proposta dal W3C. Di conseguenza non vedremo tutti quegli elementi che sono rimasti immutati con il passaggio dalla versione 1 alla versione 2.

L'XHTML, seguendo uno schema già visto per il linguaggio CSS, si è strutturato in moduli. Ogni modulo contiene elementi e/o attributi logicamente connessi. La lista completa dei vari moduli è riportata nella *tabella 1* in fondo all'articolo. Oltre ai moduli interni, l'XHTML 2 si avvale inoltre di tre moduli, detti «esterni» (vedi *tabella 2*), definiti in altri linguaggi e precisamente *Ruby Annotation*, *XForms* e *XML Events*. Da notare che gli sviluppatori dell'XHTML hanno deciso di non gestire il collegamento fra documenti — i cosiddetti *link* — con il linguaggio *XLink*, ma di continuare a utilizzare l'approccio ereditato dall'HTML, sia perché più semplice, sia perché si è ritenuto che l'XLink non soddisfacesse ancora quelle che sono le esigenze dell'XHTML.

Il modulo che ha subito maggiori modifiche è stato ovviamente quello che contiene gli elementi che descrivono la struttura del documento, ovvero lo *Structural Module*. Innanzi tutto è stato introdotto il concetto di sezione (**section**). Una sezione è generalmente formata da un'intestazione (**h**) seguita da uno o più paragrafi (**p**). Ovviamente le sezioni possono essere annidate una dentro l'altra a formare una struttura ad albero, come nell'esempio seguente:

```
<section>
  <h>Introduzione</h>
  <p>....</p>
  <h>Il regno animale</h>
  <p>...</p>
  <section>
    <h>Gli anfibi</h>
    <p>...</p>
    <section>
      <h>Gli anuri</h>
      <p>...</p>
    </section>
    <section>
      <h>Gli urodeli</h>
      <p>...</p>
    </section>
  </section>
  <section>
    <h>I rettili</h>
    <p>...</p>
  </section>
</section>
```

Fondamentalmente questa struttura è equivalente a quella tradizionale che usa i livelli predefiniti delle intestazioni (**h****n**)

```
<h1>Introduzione</h1>
<p>....</p>
<h1>Il regno animale</h1>
<p>...</p>
<h2>Gli anfibi</h2>
<p>...</p>
<h3>Gli anuri</h3>
<p>...</p>
<h3>Gli urodeli</h3>
<p>...</p>
<h2>I rettili</h2>
<p>...</p>
```

Le vecchie intestazioni sono ancora utilizzabili nell'XHTML 2, ma dato che questo tipo di elementi permette di costruire costrutti non validi in XHTML, è preferibile passare quanto prima al nuovo sistema. Con gli elementi di tipo **hn** è infatti possibile costruire sequenze che saltano alcuni livelli, come **h1...h3...h2** che, seppure accettate dai browser, sono semanticamente scorrette. Con la nuova tecnica, invece, non ci si deve più preoccupare del livello delle intestazioni che verrà automaticamente definito in base alla struttura che è stata creata. Di fatto una sezione non è altro che un contenitore, più o meno come lo sono le divisioni.

Per quanto riguarda i paragrafi (**p**), nell'XHTML 1 potevano contenere solo testo. Adesso, finalmente, è possibile includere in un paragrafo anche liste e tabelle. Si è dato cioè a questo elemento un ruolo ben più ampio, conforme a quello che poi è il suo utilizzo in campo editoriale. Ne consegue una sintassi più coerente nel definire la struttura di un documento, sintassi che a questo punto entra in conflitto con un elemento molto usato — e abusato — che è l'interruzione di linea (**br**). Nella nuova versione dell'XHTML questo elemento non può trovar posto e difatti è stato sostituito da un elemento più conforme ai principi dell'XML chiamato appunto «linea» (**l**). A differenza dell'interruzione di linea, il nuovo elemento identifica una singola linea di testo e ha quindi un marcatore iniziale ed uno finale. Grazie a questo nuovo elemento, è ora anche possibile numerare le singole linee, o alternare il colore di fondo delle righe in modo da differenziare quelle pari da quelle dispari. Ovviamente il tutto grazie ai CSS.

## XHTML:

```
<p class="titoli">
Qui di seguito è riportata la lista completa di tutti i volumi che
fanno parte del ciclo de «I Pirati della Malesia» di Emilio Salgari:
<l>I Misteri della Jungla Nera (1895)</l>
<l>Le Tigri di Mompracem (1900)</l>
<l>I Pirati della Malesia (1896)</l>
```

```
<1>Le due Tigri (1904)</1>
<1>Il Re del Mare (1906)</1>
<1>Alla conquista di un impero (1907)</1>
<1>Sandokan alla riscossa (1907)</1>
<1>La riconquista del Mompracem (1908)</1>
<1>Il Bramino dell'Assam (1911)</1>
<1>La caduta di un impero (1911)</1>
<1>La rivincita di Yanez (1913)</1>
</p>
```

## CSS:

```
p.titoli { counter-reset: linenumber ; }
l:before {
    position: relative ; left: -1em ;
    counter-increment: linenumber ;
    content: counter(linenumber) ;
}
l:nth-of-type(2n+1) { background-color: white ; }
l:nth-of-type(2n) { background-color: silver ; }
```

Un'altra modifica a questo modulo riguarda l'elemento **hr**, ovvero quell'elemento che viene spesso usato per separare graficamente due blocchi di testo. Da un punto di vista tipografico, non è detto che esso debba essere rappresentato come una linea. Ad esempio, in molti libri la separazione è ottenuta con tre asterischi o stelline poste orizzontalmente. Inoltre, anche in XHTML, dove di fatto viene reso con una linea, questa potrebbe non essere necessariamente orizzontale. Dipende di fatto dal linguaggio. Ad esempio, in un documento cinese **hr** dà luogo a una linea verticale.

Nella versione 2 questo elemento è stato quindi eliminato per far posto a un più generico elemento di separazione: **separator**. Utilizzando i fogli di stile è ora possibile dare a questo elemento qualunque aspetto e qualunque orientamento.

Per quanto riguarda il *List Module*, è stato introdotto un nuovo elemento, ovvero la lista di navigazione (**nl**). Questo elemento serve a costruire liste di collegamenti ad altri documenti che poi possono essere rappresentate in vari modi, come ad esempio un menù. La lista deve contenere sempre un'etichetta, resa con l'elemento **label**:

```
<nl>
<label>Motori di ricerca</label>
<li href="http://www.google.it/">Google</li>
<li href="http://it.yahoo.com/">Yahoo!</li>
<li href="http://it.altavista.com/">AltaVista</li>
<li>
    <nl>
```

```
<label>Italiani</label>
<li href="http://www.virgilio.it/">Virgilio</li>
<li href=http://arianna.libero.it/>Arianna</li>
</nl>
</li>
</nl>
```

Da notare l'utilizzo dell'attributo **href** direttamente nell'elemento **li**. Questa è un'altra delle novità dell'XHTML 2. Nel vecchio HTML 3, infatti, era possibile creare collegamenti solamente fra elementi di tipo **a**, come nel seguente esempio:

### PIPPO.XHTML

```
<a href="pluto.html#uscita">Scappa da qui</a>
```

### PLUTO.XHTML

```
<a name="uscita">Ben fatto!</a>
```

Poi, nell'HTML 4 e nell'XHTML 1 ogni elemento è diventato potenzialmente il bersaglio di un collegamento, ma questo doveva comunque avvenire a partire da un elemento di tipo **a**. Adesso, in XHTML 2, ogni elemento può avere gli attributi che caratterizzano un *link*, principalmente **href** e **target**. Quindi è possibile scrivere, ad esempio:

```
<table>
<tr>
<td>Il mio blog:</td>
<td
  href=https://www.lindipendente.eu
  target="_blank">L' Indipendente</td>
</tr>
</table>
```

Il vantaggio è che l'intero elemento si trasforma di fatto in un *hyperlink*, qualunque sia la sua forma e dimensione. A questo punto l'elemento **a** diventa del tutto superfluo; tuttavia, dato che esso è ampiamente utilizzato, si è ritenuto opportuno mantenerlo comunque.

Un altro cambiamento significativo riguarda l'elemento immagine (**img**). In pratica questo elemento aveva diversi svantaggi: innanzi tutto permetteva di specificare una sola risorsa, inoltre il testo alternativo (**alt**) non poteva contenere marcatori e infine l'attributo **longdesc** era sostanzialmente ignorato dalla maggior parte degli sviluppatori. Così si è deciso di eliminare del tutto questo elemento, anche se lo si è formalmente lasciato per ragioni di compatibilità, e di dare la possibilità a qualunque elemento di avere un attributo **src**. Il meccanismo è il seguente: se un elemento ha un attributo **src**, allora il suo valore viene utilizzato per accedere a una risorsa esterna; se questa è disponibile viene caricata e

il contenuto dell'elemento ignorato; se invece non si riesce a caricare la risorsa, viene visualizzato il contenuto dell'elemento. Ovviamente il meccanismo è reiterativo.

Facciamo un paio di esempi. Consideriamo prima il seguente paragrafo:

```
<p src="montagna.png" srctype="image/png">
Pippo e Clarabella alle pendici del Monte Rosa.
</p>
```

In questo caso l'agente cerca prima un'immagine in formato PNG denominata **montagna.png**. Se la trova, la carica, altrimenti visualizza il testo nel paragrafo. Naturalmente il browser può decidere di utilizzare comunque il testo per visualizzare la classica nuvoletta che nell'HTML viene generata a partire dal valore dell'attributo **alt**, se presente.

Un vantaggio di questa tecnica è quello di poter definire una risorsa alternativa se quella richiesta non è disponibile. Ad esempio:

```
<p src="montagna.png" srctype="image/png">
  <span src="nondisponibile.gif" srctype="image/gif">
    Immagine non disponibile.
  </span>
</p>
```

In questo caso, se la foto delle vacanze in montagna non viene trovata, il sistema carica un'altra immagine di *default* chiamata **nondisponibile.gif**, e se neanche questa dovesse essere trovata, mostra la scritta «Immagine non disponibile». E indubbio che questa modifica abbia potenzialità non indifferenti e che sarà decisamente apprezzata da molti sviluppatori. L'attributo **src**, infatti, può puntare a qualunque tipo di risorsa, come specificato dall'attributo **srctype**, non solo a immagini. Ad esempio, se desidero far partire un brano musicale al caricamento di un documento, basta che scriva

```
<body>
<span src="tema.mp3" type="audio/x-mpeg">Who wants to live forever
(Queen)</span>
...
</body>
```

Da notare che non conviene assegnare gli attributi **src** e **srctype** direttamente a **body**, perché così facendo il corretto caricamento dell'MP3 preverrebbe la visualizzazione del contenuto dell'elemento **body**, ovvero dell'intero documento.

Naturalmente, per caricare un immagine, si può anche usare l'elemento **object**. Questi prevede la possibilità di mostrare un messaggio mentre sta caricando la risorsa, cosa

particolarmente utile nel caso di immagini particolarmente “pesanti”. Per far questo si usa l’elemento **standby** nel seguente modo:

```
<object
  src="montagna_grande.png"
  srctype="image/png"
  xml:base="http://www.lemievacanze.pid/images/">
  <standby>Un po' di pazienza, sto caricando 4MB!</standby>
  Il Monte Rosa in un'immagine ad alta risoluzione.
</object>
```

Altri cambiamenti minori riguardano l’attributo **dir**, utilizzato per indicare la direzione del testo, ora associabile a qualunque elemento, e gli attributi **edit** e **datetime** che possono essere usati per marcare i cambiamenti effettuati in un documento rispetto a una versione precedente. Sono state fatte inoltre alcune semplificazioni per quanto riguarda l’uso delle tabelle e sono stati introdotti gli attributi **scope**, **headers**, **abbr** e **axis**, ma di questo parleremo in un altro articolo completamente dedicato proprio alle tabelle, dato che ora è possibile definire strutture decisamente sofisticate. Sempre in un altro articolo parleremo dell’integrazione dell’XHTML 2 e del Web Semantico, con particolare riferimento agli elementi **meta** e **link** e al nuovo attributo **role**.

C’è infine da segnalare come sempre più gli sviluppatori dell’XHTML ci tengano ad evidenziare l’importanza del significato concettuale di un elemento indipendentemente dalla sua rappresentazione. Ecco allora che se da un punto di vista tipografico si può decidere di usare il corsivo per più di un elemento, come ad esempio una citazione o un termine definito, da un punto di vista logico **cite**, **quote** e **dfn** sono a tutti gli effetti elementi differenti. L’obiettivo è ridurre sempre di più l’utilizzo generico di **em** e **strong** per dare enfasi a determinati termini, e spingere chi scrive un documento ad indicare *perché* si debba dare enfasi a un certo pezzo di testo, ovvero che cosa quel testo rappresenti in un certo contesto.

## TABELLE

| Modulo                                       | Elementi                                                                                          | Attributi                                                                             |
|----------------------------------------------|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| <i>Document Module</i>                       | <b>body, head, html, title</b>                                                                    |                                                                                       |
| <i>Structural Module</i>                     | <b>address, blockcode, blockquote, div, h, h1, h2, h3, h4, h5, h6, p, pre, section, separator</b> |                                                                                       |
| <i>Text Module</i>                           | <b>abbr, cite, code, dfn, em, kbd, l, quote, samp, span, strong, sub, sup, var</b>                |                                                                                       |
| <i>Hypertext Module</i>                      | <b>a</b>                                                                                          |                                                                                       |
| <i>List Module</i>                           | <b>dl, dt, dd, label, nl, ol, ul, li</b>                                                          |                                                                                       |
| <i>Core Attributes Module</i>                |                                                                                                   | <b>class, id, title</b>                                                               |
| <i>Hypertext Attributes Module</i>           |                                                                                                   | <b>href, hreftype, cite, target, rel, rev, access, nextfocus, prevfocus, xml:base</b> |
| <i>Internationalization Attribute Module</i> |                                                                                                   | <b>xml:lang</b>                                                                       |
| <i>Bi-directional Text Module</i>            |                                                                                                   | <b>dir</b>                                                                            |
| <i>Edit Attributes Module</i>                |                                                                                                   | <b>edit, datetime</b>                                                                 |
| <i>Embedding Attributes Module</i>           |                                                                                                   | <b>src, type</b>                                                                      |
| <i>Handler Module</i>                        | <b>handler</b>                                                                                    |                                                                                       |
| <i>Image Map Attributes Module</i>           |                                                                                                   | <b>usemap, ismap, shape, coords</b>                                                   |
| <i>Media Attribute Module</i>                |                                                                                                   | <b>media</b>                                                                          |
| <i>Metainformation Attributes Module</i>     |                                                                                                   | <b>about, content, datatype, property, rel, resource, restype, rev</b>                |
| <i>Metainformation Module</i>                | <b>meta, link</b>                                                                                 |                                                                                       |
| <i>Object Module</i>                         | <b>object, param, standby</b>                                                                     |                                                                                       |
| <i>Style Attribute Module</i>                |                                                                                                   | <b>style</b>                                                                          |
| <i>Stylesheet Module</i>                     | <b>style</b>                                                                                      |                                                                                       |
| <i>Tables Module</i>                         | <b>caption, col, colgroup, summary, table, tbody, td, tfoot, th, thead, tr</b>                    |                                                                                       |

**Tabella 1 - Lista completa dei moduli interni**

| Modulo                        | Elementi                                                                                                                                                                                                                                                                                                                                                                                                        | Attributi                                                                                |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| <i>Ruby Annotation Module</i> | <code>ruby, rbc, rtc, rb, rt, rp</code>                                                                                                                                                                                                                                                                                                                                                                         |                                                                                          |
| <i>XForms Module</i>          | <code>action, alert, bind, case, choices, copy, delete, dispatch, extension, filename, group, help, hint, input, insert, instance, item, itemset, label, load, mediatype, message, model, output, range, rebuild, recalculate, refresh, repeat, reset, revalidate, secret, select, select1, send, setfocus, setindex, setvalue, submission, submit, switch, textarea, toggle, trigger, upload, and value</code> | <code>repeat-model, repeat-bind, repeat-nodeset, repeat-startindex, repeat-number</code> |
| <i>XML Events Module</i>      | <code>listener</code>                                                                                                                                                                                                                                                                                                                                                                                           | <code>defaultAction, event, handler, objserver, phase, propagate, target</code>          |

**Tabella 2 - Lista completa dei moduli esterni**