

XHTML

Massimo Martinelli

Massimo.Martinelli@isti.cnr.it



Consiglio Nazionale delle Ricerche - CNR
Istituto di Scienza e Tecnologie della Informazione - ISTI



Cosa è XHTML?

- eXtensible HyperText Markup Language
- XHTML 1.0 Una riformulazione di HTML 4.01 come applicazione XML
- Uno standard W3C
raccomandazione dal 26 Gennaio 2000
- “XHTML è una famiglia di attuali e futuri tipi di documenti che riproduce, ingloba ed estende l'HTML 4”

Perchè usare XHTML? Obiettivi

- Non è un *complotto* per rendere più complicato e di elite il Web, tutt'altro!
- Separare la struttura del documento dallo stile
 - Struttura: intestazioni <h>, paragrafi , tabelle, ...
 - Cascading Style Sheets (CSS): colori, tipi di carattere font, grandezze, posizioni,...
- Integrare nuovi vocabolari (SVG, MathML, SMIL, Metadati,...)
- Utilizzare nuovi dispositivi (cellulari, palmari, ...)

Perchè usare XHTML?

- Uno standard per tutti i dispositivi collegati ad internet, dai computer a cellulari alle TV
- Incoraggiare gli sviluppatori di browser ad aderire ad uno standard
- XHTML 1.0, 1.1 è compatibile con i browser che supportano HTML 4
- XHTML 2.0 (Working Draft 18 Dicembre 2002) non è pensato per essere compatibile con le versioni precedenti (HTML 4.01, XHTML 1.x)
- Codifica più pulita

Markup difettoso (*bad* - cattivo)

```
<html>
```

```
<head>
```

```
<title>HTML difettoso</title>
```

```
<body>
```

```
<h1> HTML difettoso      </h1>
```

```
</body>
```

```
</html>
```

- Non tutti i dispositivi hanno le risorse per interpretare un linguaggio di markup "difettoso"

Perche Modularizzare XHTML

- Quando si parla di modularizzazione di XHTML ci si riferisce al lavoro di specifica di insiemi ben definiti di elementi XHTML che possono essere combinati ed estesi per rendere più flessibile agli sviluppatori di codice l'invio di contenuti su un numero più ampio ed eterogeneo di piattaforme.
- L'uso di HTML è in continua crescita, e si prevede l'ulteriore crescita e l'adozione anche da parte di nuove piattaforme quali computer portatili, cellulari, televisioni ... tutti dispositivi con differenti esigenze e capacità.
- La modularizzazione di XHTML fornisce ai produttori un sistema per specificare quali elementi sono supportati da un dispositivo utilizzando metodi standard.
La modularizzazione permette inoltre di estendere le capacità di presentazione di XHTML usando l'estensibilità di XML senza violare lo standard XHTML.

Moduli XHTML

- Non unica dichiarazione di elementi e attributi
- Ciascun modulo definisce un insieme di elementi e attributi per una determinata classe di oggetti
- Modulo strutture: struttura di base di un documento
(head, html, title, body, <!-- comment-->)
- Modulo tabelle (table, tr, th, td, caption)
- Modulo form (form, input, textarea, select, option)
- Modul liste (ul, ol, li, dl, dt, dd)
- Moduli immagini, link, eventi, ...
- Posso utilizzare un modulo o non utilizzarlo a seconda delle necessità (mi servono i form, includo il modulo dei form— non mi servono i form, non includo il modulo dei form)

Differenze principali tra HTML e XHTML

- Gli elementi di XHTML devono **essere annidati correttamente**
- I documenti di XHTML devono essere **ben formati**
- I nomi dei tag devono essere scritti con **lettere minuscole**
- Tutti gli elementi di XHTML devono essere **chiusi**

Documento ben-formato, valido

Un documento si dice "ben formato" quando:

- contiene almeno un elemento;
- esiste un *tag* unico di apertura e di chiusura contenente l'intero documento;
- tutti i tag sono nidificati
- tutte le entità sono dichiarate.

Un documento si dice "valido" quando

- contiene un DTD e rispetta le regole definite nel DTD.

DTD - Document Type Definition

- contiene le regole di definizione dei *tag*
- indica gli elementi e il loro ordine all'interno del documento XML
- il suo uso non è obbligatorio; ne è consigliato l'utilizzo
- può essere interno o esterno al documento

Corretto annidamento degli elementi

Sbagliato:

<i>questo testo è grassetto e corsivo</i>

Corretto:

<i>questo testo è grassetto e corsivo</i>

Corretto annidamento degli elementi

Sbagliato !

```
<ul>  
  <li>Caffe</li>  
  <li>Tè  
    <ul>  
      <li>Tè nero</li>  
      <li>Tè verde</li>  
    </ul>  
  <li>Latte</li>  
</ul>
```

Corretto:

```
<ul>  
  <li>Caffe</li>  
  <li>Tè  
    <ul>  
      <li>Tè nero</li>  
      <li>Tè verde</li>  
    </ul>  
  </li>  
  <li>Latte</li>  
</ul>
```

I documenti devono essere ben-formati

- **Tutti gli elementi di XHTML devono essere annidati all'interno dell'elemento radice <html>**
- **Tutti gli altri elementi possono avere elementi sotto-elementi (figli).**
- **I sotto-elementi devono essere in coppia (aperti e chiusi) ed annidati correttamente all'interno del loro elemento genitore**

La struttura di base del documento è:

<html>

<head> ... </head>

<body> ... </body>

</html>

I nomi dei tag devono essere scritti con lettere minuscole

- Questo perché i documenti XHTML sono applicazioni di XML e queste hanno i tag scritti con lettere minuscole. XML è case-sensitive (le lettere minuscole vengono interpretate diversamente dalle maiuscole). I tag come `
` e `
` sono interpretati come tag differenti.

- Questo è sbagliato:

```
<BODY>
```

```
<P>Questo è un paragrafo</P>
```

```
</BODY>
```

- Questo è corretto:

```
<body>
```

```
<p>Questo è un paragrafo</p>
```

```
</body>
```

Tutti gli elementi XHTML devono essere chiusi

- **Gli elementi non vuoti devono avere un tag di chiusura**

- Questo è sbagliato:

```
<p>Questo è un paragrafo
```

```
<p>Questo è un altro paragrafo
```

- Questo è corretto:

```
<p>Questo è un paragrafo</p>
```

```
<p>Questo è un altro paragrafo</p>
```

Anche gli elementi vuoti devono essere chiusi

- *Anche gli elementi vuoti devono avere un tag di chiusura oppure il tag di apertura deve finire con />.*

- *Questo è sbagliato:*

Questa è una interruzione

Qui c'è una linea orizzontale:<hr>

Qui c'è un'immagine

- *Questo è corretto:*

Questa è una interruzione

</br>

Qui c'è una linea orizzontale:<hr />

Qui c'è un'immagine

IMPORTANTE: per rendere il nostro XHTML compatibile con gli odierni browser, sarebbe necessario aggiungere uno spazio supplementare prima del simbolo "/" come questo:
 e questo: <hr />.

Altre regole di sintassi XHTML

- I nomi di attributo devono essere scritti con **lettere minuscole**
- I valori di attributo devono **essere quotati** ("nome_attributo")
- La minimizzazione di attributi è **proibita**
(non si può usare la sintassi abbreviata HTML)
`<input type="radio" CHECKED>` ERRATO
`<input type="radio" checked="checked" />` CORRETTO
- L'attributo d'identificazione ***id*** **sostituisce** l'attributo name
- Il DTD XHTML definisce **gli elementi** obbligatori

I nomi degli attributi devono essere scritti con lettere minuscole

Questo è sbagliato:

```
<table WIDTH="100%">
```

Questo è corretto:

```
<table width="100%">
```

I valori degli attributi devono essere quotati (tra apici)

Questo è sbagliato:

```
<table width=100%>
```

Questo è corretto:

```
<table width="100%">
```

E' proibito minimizzare gli attributi

Questo è sbagliato:

```
<input checked>  
<input readonly>  
<input disabled>  
<option selected>  
<frame noresize>
```

Questo è corretto:

```
<input checked="checked" />  
<input readonly="readonly" />  
<input disabled="disabled" />  
<option selected="selected">  
<frame noresize="noresize">
```

L'attributo id sostituisce l'attributo name

- HTML 4.01 definisce un attributo name per gli elementi a, applet, frame, iframe, img, e map. In XHTML l'attributo name è deprecato (disapprovato). Bisogna invece usare id

Questo è sbagliato:

```

```

Questo è corretto:

```

```

Nota: Per interoperare con i vecchi browsers fino a che continueranno ad essere usati, sarebbe necessario usare sia name che id, con identici valori degli attributi , come in questo caso:

```

```

Nota IMPORTANTE per la compatibilità:

Per rendere il nostro XHTML compatibile con gli odierni browser, sarebbe necessario aggiungere uno spazio prima del simbolo "/".

L'attributo lang

- L'attributo lang si applica a quasi tutti gli elementi XHTML. Esso specifica il linguaggio del contenuto di un elemento.
- Se usiamo l'attributo lang in un elemento, dobbiamo aggiungere l'attributo xml:lang, come in questo caso:

```
<div lang="en" xml:lang="en">Hello !</div>
```

Elementi XHTML obbligatori

- Tutti i documenti di XHTML devono avere una dichiarazione di DOCTYPE. Gli elementi html, head, body devono essere presenti title deve essere presente all'interno dell'elemento head.
- Questo è un modello minimo di documento XHTML:

```
<!DOCTYPE qui va messo il Doctype>
<html>
<head>
<title>Il titolo va messo qui</title>
</head>
<body>
  Il testo del corpo va messo qui
</body>
</html>
```

Nota: La dichiarazione DOCTYPE non fa parte dello stesso documento XHTML. Non è un elemento XHTML, e non dovrebbe avere un tag di chiusura.

Document Type Definition - DTD

“Dizionario” degli elementi e degli attributi

Esistono attualmente 4 DTD di XHTML1 DTDs:

- XHTML 1.0 Transitional
- XHTML 1.0 Strict
- XHTML 1.0 Frameset

- XHTML Basic

Sintassi XHTML

Document Type Definitions (DTDs)

- XHTML 1.0 Transitional
 - Permette l'uso di elementi deprecati da HTML 4 (<center>)
 - Più facile da imparare
 - contiene l'intero linguaggio ammesso per HTML, inclusi quegli elementi "deprecati" che vengono mantenuti per compatibilità col passato.
- La DTD:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN" "DTD/xhtml1-  
transitional.dtd">
```

Sintassi XHTML

Document Type Definitions (DTDs)

- XHTML 1.0 Strict
 - Nessuna formattazione all'interno del codice, lo stile si gestisce interamente con CSS
 - Non permette elementi deprecati
- La DTD:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Strict//EN" "DTD/xhtml1-strict.dtd">
```

Sintassi XHTML

Document Type Definitions (DTDs)

- XHTML 1.0 Frameset
 - Utilizzata da pagine con frame
- La DTD:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Frameset//EN" "DTD/xhtml11-frameset.dtd">
```

Sintassi XHTML

Document Type Definitions (DTDs)

- XHTML Basic
 - Utilizzata per piccoli dispositivi con capacità di visualizzazione limitata
 - Non è permesso il tag script
- La DTD:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic  
1.0//EN" "http://www.w3.org/TR/xhtml-basic/xhtml-  
basic10.dtd">
```

Gestione degli spazi

- Se il contenuto degli attributi contiene più spazi consecutivi (spazi, a capo, tabulatori), questi vengono sostituiti da un unico spazio dallo user agent

Proprietà whitespace (CSS2)

La proprietà whitespace determina la significatività degli dei caratteri spazio, tabulazione, a capo. I possibili valori sono

- normal
 - pre
 - nowrap
-
- Il valore di default, normal, significa che più spazi sono condensati in uno solo.
 - Il valore pre funziona come l'elemento pre (preformattato) in HTML.
 - Tutti gli spazi sono considerati significativi e riprodotti fedelmente. (sarebbe bene usare in combinazione con un fonto monospace).
 - Il valore nowrap considera significativi gli a capo ma condensa gli altri caratteri in un unico spazio.

```
p { display: block; white-space: pre; font-family: monospace; }
```

Elementi Script e Style

- In XHTML gli elementi script e style sono dichiarati di tipo #PCDATA
 - Parsed Character Data elaborabili dal parser (analizzatore del codice)
- `<` & sono interpretati come inizio di markup, e le entità come `<` and `&` sono riconosciute come riferimenti ad entità
- (`<` &).
- Includendo il contenuto di questi elementi all'interno di una sezione CDATA evitiamo l'espansione di queste entità
 - CDATA = Character Data (il parser non analizza il contenuto)
- `<script type="text/javascript">`
 - `<![CDATA[... unescaped script content ...]]>`
- `</script>`

Riferimenti ad entità

- È possibile utilizzare un riferimento ad un valore esadecimale.
- Ad esempio per i caratteri diacritici o per specificare i colori.
- Quando esistono, si possono utilizzare le parole chiave
- Un altro metodo consente di utilizzare il codice esadecimale
- Usando `&xnn;` (x e nn in minuscolo)
- Esempio `&ffffff;` (bianco)

Elementi deprecati

<basefont>

<center>

<i>

- Come fare senza questi elementi ?
Usare CSS!

ESERCIZIO

Costruire un documento XHTML
contenente una intestazione, una tabella
e una lista non ordinata