# HTML e XHTML

Fabio Vitali

***Venerdì 17*** marzo 2000

# Introduzione

Oggi esaminiamo in breve:

- La storia di HTML
- Caratteristiche di HTML 4
- Il DTD di HTML 4
- XHTML 1.0



# Introduzione ad HTML (1)

Il linguaggio HTML è un tipo di documenti SGML (esiste un DTD di HTML).

HTML viene usato dai browser WWW per visualizzare documenti ipertestuali. Tramite HTML è possibile realizzare documenti con una semplice struttura, aspetti grafici anche sofisticati, che contengono testo, immagini, oggetti interattivi e connessioni ipertestuali ad altri documenti

Fino ad oggi i browser si sono preoccupati poco della correttezza sintattica o strutturale dei documenti HTML. Questo significa che tra un documento HTML ***visualizzabile da un browser WWW*** ed un documento HTML ***corretto*** esistono differenze anche sensibili.



# Introduzione ad HTML (2)

È normale associare un significato strutturale agli elementi definiti in un DTD. HTML associa anche significati grafici agli elementi che definisce. Cioè dà istruzioni più o meno precise su come rendere graficamente gli elementi che definisce.

Questo porta anche ad abusi della resa grafica che a noi interessano poco. Per noi la resa grafica finale, in assenza di linguaggi di stile appropriato, è secondaria.

HTML non forza strutture gerarchiche nei suoi documenti. Inoltre in HTML i vincoli di contenimento tra elementi sono pochi e piuttosto ovvi. I browser WWW sono ancora più lassisti a questo proposito.



# Storia di HTML (1)

HTML è esistito in varie versioni dal 1989 ad oggi:

- 0.9 (?): il linguaggio di HTML disponibile sul browser WWW aveva strutture di base per intestazioni, paragrafi e stili base, più ovviamente il tag A che ne costituiva la novità.
- 1.0 (1992): I primi browser shareware e freeware al mondo (il più importante di questo periodo fu Arena, ma esistevano anche MacWeb ed altri) implementavano alcune versioni di HTML leggermente diverse. Tra queste ebbe un certo successo la proposta HTML+. Viene introdotto il tag IMG e il supporto per il GIF.
- 2.0 (1994): La prima versione veramente nota di HTML. Questa è quella implementata su Mosaic, da cui deriverà Netscape. E' la prima versione ad essere formalizzata su un DTD SGML, invece che ispirarsi vagamente ad SGML. Introduce i form.



# Storia di HTML (2)

- 3.0 (1995): Questa versione non è mai stata ufficialmente approvata. Durante la sua discussione vennero proposte molte aggiunte. Alcune di queste vennero implementate prima di raggiungere un consenso (tabelle), altre (ad es. supporto per la matematica) mai prese in considerazione.
- 3.2 (1997): Quando divenne chiaro che i browser non avrebbero supportato tutto il 3.0, si lavorò per generarne un sottoinsieme su cui ci fosse consenso, e che tenesse conto delle aggiunte proprietarie dei vari produttori. Questa versione include tabelle, applet, script e altre migliorie, ma non i frame, sebbene Netscape e Microsoft le avessero già implementate fin dal 1995.
- 4.0 (1997): supporto per l'internazionalizzazione, per gli style sheet, per i frame, tabelle molto più ricche, il tag OBJECT, ecc.
- 4.01 (dic. 1999): contiene alcune minime variazioni e correzioni.



# Storia di HTML (3)

- XHTML1.0: Nel 1998 parte l'iniziativa di riformulare HTML come applicazione di XML, piuttosto che di SGML.

  Il 26 gennaio 2000 esce la prima recommendation del W3C, XHTML 1.0, che è una semplice riformulazione di HTML 4 in termini di XML, senza nessuna introduzione di nuove forme.

  XHTML però identifica anche un percorso di evoluzione verso la creazione di una famiglia di tipi di documenti che estendano localmente o semplifichino XHTML per una vasta gamma di usi e device.

  Esistono già due working group attivi:

  - Modularizzazione di XHTML: per l'identificazione delle regole di scomposizione e modularizzazione dei vari aspetti di XHTML.
  - XHTML Basic: per l'identificazione del sottoinsieme minimale di XHTML per l'uso in apparecchi mobili, telefoni, sistemi embedded, ecc.



# HTML 4

Una prima recommendation è del 18 dicembre 1997, una revisione del 24 aprile 1998. Il 24 dicembre 1999 esce HTML 4.01, che contiene alcuni cambiamenti editoriali e precisazioni.

La diffusione dei fogli di stile ha liberato i web designer dall'obbligo di forzare i tag HTML ad assumere uno scopo tipografico e decorativo.

Però alcuni tag esistono solo allo scopo di fornire istruzioni tipografiche di base. Per compatibilità col passato, tuttavia, si è deciso di mantenere il supporto per alcuni elementi che non hanno più senso con l'uso dei fogli di stile.



# I DTD di HTML 4

Per esigenze di compatibilità, HTML 4 è composto di tre DTD alternativi:

- Transitional DTD (detto anche loose): contiene l'intero linguaggio ammesso per HTML, inclusi quegli elementi "deprecati" che vengono mantenuti per compatibilità col passato.
- Strict DTD: contiene i soli elementi di HTML che non vengono influenzati dall'uso degli style sheet (ma sono escluse le tabelle, che non sono gestite da CSS).
- Frameset DTD: un semplicissimo DTD per quei documenti in cui al posto di BODY si usano i tag dei frame.



# Criteri di sviluppo (1)

HTML 4.0 estende HTML 3.2 con meccanismi per i fogli di stile, gli script, i frame, oggetti embedded, criteri di internazionalizzazione, tabelle più ricche e miglioramenti ai form.

Questi sono i criteri di sviluppo più importanti:

- **Internazionalizzazione** (***Internationalization*** o ***I18N***): l'adozione dei meccanismi necessari per il supporto di linguaggi e notazioni di tutto il mondo, e per la creazione di documenti contenenti linguaggi misti.
- **Accessibilità**: l'adozione dei meccanismi necessari per il supporto delle esigenze degli utenti con limitazioni fisiche (visive, uditive, etc.).



# Criteri di sviluppo (2)

- **Tabelle sofisticate**: l'adozione di meccanismi necessari per creare tabelle ancora più sofisticate delle precedenti.
- Documenti composti: l'adozione dei meccanismi necessari per inserire (embed) in maniera generalizzata oggetti di ogni possibile media all'interno di una pagina HTML.
- **Style sheet**: l'adozione di meccanismi per specificare in maniera precisa e sofisticata la resa tipografica di una pagina senza appesantire la gestione del contenuto.
- **Scripting**: l'adozione dei meccanismi necessari per realizzare sul client degli oggetti attivi, in grado di eseguire computazioni locali (ad esempio, per pre-verificare la correttezza delle informazioni inserite in un form).



# Usare HTML 4 (1)

## Separazione di struttura e presentazione

- Via via che HTML tende ad assomigliare al suo antenato SGML, molti dei suoi aspetti presentazionali vengono sostituiti o affiancati da altri meccanismi, in particolare gli style sheet. Questo porta in particolare a "deprecare" gli aspetti più presentazionali di HTML (ad esempio, FONT), e a proporre meccanismi alternativi, piu' indipendenti e sofisticati (ad esempio, gli style sheet).

## Considerare l'accessibilità universale al Web

- Via via che si considera il supporto di un numero maggiore di utenti, maggiori saranno le differenze tra di essi di cui tenere conto: esigenze linguistiche specifiche, minorazioni fisiche, architetture diverse, modalità diverse di fruizione delle pagina richiedono gli autori di considerare appropriatamente le specifiche esigenze di tutti.



# Usare HTML 4 (2)

## Aiutare i browser con il rendering incrementale

- Immagini, oggetti embedded, tabelle complesse rendono complicato (e quindi lento) il meccanismo di impaginazione dei documenti HTML. L'adozione di misure per permettere la visualizzazione incrementale delle caratteristiche del documento favorisce una velocità percepita di visualizzazione utile per il buon successo delle proprie pagine.

## Internazionalizzazione (I18N)

- Il supporto per l'internazionalizzazione passa attraverso 4 meccanismi:
  - Il set di caratteri (UCS -8, 16, 32).
  - L'attributo lang
  - L'attributo dir
  - Il tag BDO (di-directional override)



# Principali aspetti di HTML

- Tipi di dati principali: colori, lunghezze, URI, ecc.
- Struttura dei documenti HTML (HTML, BODY, HEAD, ecc.)
- Elementi per testo e liste
- Link, oggetti inclusi ed immagini
- Tabelle
- Fogli di stile e script
- Form
- Frame
- Aspetti presentazionali (font, allineamenti, HR)



# Una visita al DTD di HTML 4: loose DTD

Clicca qui



# XHTML 1.0

XHTML 1.0 è una riformulazione di HTML 4 come un'applicazione di XML 1.0. La semantica degli elementi e degli attributi non è assolutamente cambiata da HTML 4.

XHTML è il primo di una serie di DTD che riproducono, limitano ed estendono HTML. Sono fatti per lavorare con user agent basati su XML, ma con un'esplicita strategia di transizione.

I documenti XHTML sono documenti XML, scritti per essere usati anche in ambienti HTML 4, danno accesso immediato a tulte le caratteristiche DOM.



# Differenze con HTML 4 (1)

- I documenti debbono essere ben formati, in particolare l'annidamento deve essere corretto.

```
<p>Paragrafo con <em>enfasi</p></em>
```

- I nomi di elementi e attributi sono minuscoli (XML è case-sensitive)

```
<P> Oggi siamo <EM>qui</EM></P>
```

- Il tag finale è obbligatorio per elementi non vuoti

```
<UL>
    <LI> Primo elemento
    <LI> secondo elemento
</UL>
```

- Gli elementi vuoti debbo seguire la sintassi XML

```
<HR/> <BR/>
```



# Differenze con HTML 4 (2)

- I valori degli attributi debbono sempre avere le virgolette

```
<input type=checkbox value=pippo>
```

- Non esiste il concetto di minimizzazione degli attributi

```
<input type="radio" checked>
<input type="radio" checked="checked">
```

- Elementi con attributi "id" e "name"

  In HTML esistono due tipi di attributi identificatori: "name" per i tag come a, applet, form, frame, iframe, img, e map, ed "id" per esprimere caratteristiche di stile per tutti gli elementi.

  In XML, solo UN attributo alla volta può essere di tipo ID. E' stato scelto quindi che l'attributo "id" sia di tipo ID, mentre "name non lo è. Quindi documenti XHTML debbono usare "id" come attributo per l'identificazione di frammenti, e "name" è diventato deprecato.



# Differenze con HTML 4 (3)

- Elementi di stile e script

  Poiché il contenuto di SCRIPT e STYLE è definito come #PCDATA, i caratteri “<“ e “&” sono significativi per XML e quindi vengono interpretati come markup. Per evitarlo, è necessario mettere gli script e gli stili esternamente, oppure usare sezioni CDATA.

```
<script>
        <![CDATA[ qualunque carattere ]]>
</script>
```

- Esclusioni SGML

  Il content model di alcuni elementi in HTML esclude esplicitamente l’annidamento ricorsivo (es., A non può contenere altri A). Questo purtroppo non è esprimibile in XML, ma è comunque non accettabile.



# Compatibilità con HTML

- **Processing instructions**

  Alcuni browser rendono le processing instructions come tag.

- **Elementi vuoti**

  Affinché un elemento vuoto sia comprensibile sia da motori XML che HTML, è possibile mettere uno spazio vuoto prima del carattere di fine tag:

  ```
  <HR /> <BR />
  ```

- **Stili e script embedded**

  Stili e script vanno posti fuori dal documento se contengono “<“, “&” o “]]>”. Da notare che molti processori XML rimuovono i commenti, quindi non si può più inserire gli script e gli stili in commenti SGML.

- **Identificatori di frammenti**

  E’ opportuno usare sia l’attributo id che l’attributo name, ogni volta che ve ne sia bisogno.



# Conclusioni

Oggi abbiamo parlato di

- La storia di HTML
- Caratteristiche di HTML 4
- Il DTD di HTML 4
- XHTML 1.0



# Riferimenti

## *Wilde's WWW, capitolo 7*

**Altri testi:**

- D. Raggett, A. Le Hors, I. Jacobs, ***HTML 4.01 Specification***, W3C Recommendation 24 December 1999, **`http://www.w3.org/TR/html401`**
- S. Pemberton et alii, ***XHTML™ 1.0: The Extensible HyperText Markup Language, A Reformulation of HTML 4 in XML 1.0***, W3C Recommendation 26 January 2000, **`http://www.w3.org/TR/xhtml1`**