Guida XHTML

(www.html.it) - http://xhtml.html.it/guide/leggi/52/guida-xhtml/

Introduzione

26 gennaio **2000**: il World Wide Web Consortium (**W3C**) rilascia la prima specifica del linguaggio di markup destinato a sostituire HTML. Quel linguaggio si chiama **XHTML**. La specifica di XHTML occupa una sola pagina. **Non ci sono nuovi tag**, non troverete nulla di rivoluzionario rispetto ad HTML 4.0. Eppure, quello che è attualmente il linguaggio raccomandato dal Consortium per la realizzazione di pagine web, è davvero un passo decisivo e fondamentale.

L'elemento chiave è la **X**. Sta per **EXTENSIBLE**, è la stessa X su cui si fonda quella che è sicuramente la pietra angolare della comunicazione digitale del futuro: **XML** (Extensible Markup Language). La cosa "rivoluzionaria" è appunto questa: HTML è ora una parte della grande famiglia XML, ne condivide regole di base e potenzialità. Risultato: **il tempo dell'anarchia, del codice "sporco ma basta che funzioni", degli elementi proprietari imposti è finito**. E tutti quelli seriamente interessati allo sviluppo di un web migliore dovrebbero esserne felici.

Con XHTML chi progetta una pagina "standard compliant" ha la certezza che sarà visualizzata senza problemi con questi strumenti e che lo sarà nel futuro. È l'ora della "forward compatibility".

L'obiettivo di XHTML è di separare il contenuto dalla presentazione e dalla logica di programmazione.

Cos'è XHTML

HTML + XML = XHTML

HTML è un linguaggio di marcatura per presentare i contenuti di una pagina web. La sua semplicità è la base dell'esplosione di Internet. L'ultima raccomandazione rilasciata dal W3C è la 4.01 (dicembre 1999).

XML è una sorta di "super-linguaggio" che consente la creazione di nuovi linguaggi di marcatura. Potente, flessibile e rigoroso è alla base di tutte le nuove specifiche tecnologiche rilasciate dal W3C e adottate ormai come standard dall'industria informatica. I principali obiettivi di XML, dichiarati nella prima specifica ufficiale (ottobre 1998), sono pochi ed espliciti: utilizzo del linguaggio su Internet, facilità di creazione dei documenti, supporto di più applicazioni, chiarezza e comprensibilità.

XHTML è la riformulazione di HTML come applicazione XML. Ciò significa essenzialmente una cosa: un documento XHTML deve essere **valido** e **ben formato**. Niente nuovi tag, attributi o metodi. Questi rimangono essenzialmente quelli di HTML 4.0. Il vocabolario rimane uguale, ma cambiano le regole sintattiche. XHTML mantiene la compatibilità con i software che supportano HTML 4.0

Versioni di XHTML

Le versioni di XHTML attualmente disponibili e pubblicate come raccomandazioni dal W3C sono tre.

XHTML 1.0

Pubblicata il 26 gennaio 2000 e seguita da una versione rivista dell'ottobre 2001. Consiste, come detto, nella riscrittura in XML di HTML 4.0 e si basa sulle tre DTD già definite per questo linguaggio:

DTD Strict

DTD Transitional

DTD Frameset

XHTML Basic

È una versione "ridotta" del linguaggio. Specificamente pensata per dispositivi mobili (PDA, cellulari), contiene solo gli elementi che si adattano a questi dispositivi (esclude, ad esempio, i frames che non hanno ovviamente senso in tale contesto). È destinata a sostituire WML come linguaggio di base per le applicazioni WAP.

XHTML 1.1

Basata sulla DTD Strict della versione 1.0, rappresenta la prima formulazione pratica del concetto di modularità. In questa visione, gli elementi fondamentali (in pratica l'insieme di tag che definiscono la struttura di un documento) sono raggruppati in una serie di moduli indipendenti, che possono essere implementati o esclusi secondo le necessità. Secondo il W3C è la base della futura estensione di XHTML con altri set di linguaggi o moduli, anche personalizzati.

Specifiche: http://www.w3c.it/traduzioni/xhtml1-it.html

Vantaggi di XHTML

Milioni di pagine sono scritte in HTML. Una gran parte di esse non è valida rispetto alle raccomandazioni del W3C, ma funziona. **L'adozione** di un nuovo linguaggio (XHTML) **non è obbligatoria** né forzata: tutti i browser in commercio sono in grado di interpretare HTML 4. XHTML non introduce nuove features rispetto al suo predecessore. "Perché allora imparare un nuovo linguaggio?". "Perché dovrei riscrivere tutto il mio sito?". Sono domande scontate, ma legittime. Ecco alcune possibili risposte ed un semplice elenco dei vantaggi di XHTML.

Codice pulito e ben strutturato

Il passaggio ad XHTML può essere visto come un ritorno alle origini. HTML è nato come linguaggio per definire la struttura di un documento. Non ha nulla a che vedere con la presentazione. Eppure, è stato stiracchiato da tutte le parti, costretto a svolgere compiti per cui non è stato creato. Il caso delle tabelle è quello macroscopico. Sono nate per la presentazione di dati tabulari. Sono state impiegate come l'unico mezzo per costruire il layout della pagina.

E ancora: quando si sentì la necessità di gestire la tipografia dei documenti, venne introdotto il tag <font> con i relativi attributi. La conseguenza è quella di pagine cariche di elementi inutili, più pesanti, difficili da modificare.

La responsabilità principale per questo uso improprio di HTML non ricade sugli sviluppatori. Nel 1996 (le date sono importanti!) il W3C ha rilasciato la versione definitiva di CSS1. Era questa la tecnologia destinata a definire la presentazione dei documenti. L'idea era chiara: **HTML per la struttura, CSS per lo stile** e il layout. Eppure, per avere browser che supportano decentemente questi standard si è dovuto attendere il 2000-2001. Quattro anni, tre generazioni di browser, milioni di siti costruiti tentando di risolvere incompatibilità e bug.

Con XHTML, almeno nella sua versione Strict, si torna ad un linguaggio che definisce solo la struttura. Semplicemente, se inserite elementi non supportati (font, larghezza per le celle di tabelle o margini per il body, per citare solo alcuni esempi) il documento non è valido. Quindi: la formattazione si fa con i CSS. Mai più tag <font>, mai più gif di un pixel. Tra poco, forse, niente più tabelle per il layout. Risultato: codice più pulito, più logico, più gestibile.

Portabilità

Portabilità è la capacità/possibilità di un documento di essere visualizzato e implementato efficacemente su diversi sistemi: PC, PDA, cellulari WAP/GPRS, WebTV. Ora, se pensate alle limitazioni in termini di memoria, ampiezza dello schermo e usabilità di un terminale mobile, capirete perché è importante un linguaggio essenziale, centrato sulla struttura del documento, privo di orpelli inutili. Ciò che ha senso è avere un titolo della pagina, un'intestazione, un paragrafo, una lista per scandire gli argomenti.

Sulla portabilità poggia l'enfasi con cui aziende del calibro di Nokia, Motorola, Ericsson o Siemens guardano ad XHTML. Dopo l'accettazione da parte del Wap Forum si può affermare con certezza che la piattaforma WAP 2 e tutta l'evoluzione dei servizi mobili sarà fondata sull'integrazione tra XHTML e CSS, con il supporto delle necessarie tecnologie sul lato server.

Estensibilità

Dal momento che XHTML è XML diventa estensibile. Significa che sarà facilissimo incorporare in un documento parti scritte in uno dei tanti linguaggi della famiglia XML.

Accessibilità

I documenti scritti in XHTML e validati sono naturalmente più accessibili. Da una parte la validazione richiede l'uso obbligatorio di funzionalità come il testo alternativo per le immagini. Dall'altra, una pagina che evita elementi non standard, ben definita nella struttura è di gran lunga meglio gestibile da browser alternativi come quelli vocali o testuali.

Regole di base

Definiscono i requisiti minimi essenziali di un documento. Se non si rispettano questi semplici punti il documento non può essere definito XHTML.

1. Validazione

Un documento deve essere convalidato rispetto ad una delle tre DTD XHTML del W3C. La validazione controlla essenzialmente due cose: che il documento sia valido e ben formato.

Documenti ben formati

Il documento deve rispettare le regole della sintassi XML: presenza di un elemento radice, corretto annidamento degli elementi, chiusura obbligatoria dei tag vuoti, etc.

Documenti validi

Un documento è valido se usa correttamente un linguaggio, vale a dire se usa nel modo giusto solo elementi specifici e consentiti. Ma dove vengono stabilite queste regole? Per XHTML (e per XML in genere, anche se in questo ambito si sta sviluppando l'uso degli schemi) le regole sono definite nelle DTD (Document Type Definition). Una DTD identifica gli elementi (tag) consentiti, cosa essi significano, come devono essere trattati (ad esempio, stabilisce quali sono gli attributi possibili per ciascun elemento). In un documento XHTML la DTD deve essere obbligatoriamente specificata all'inizio.

2. Elemento radice

Ogni documento XML deve contenere un elemento radice. Si tratta dell'elemento che contiene al suo interno tutti gli altri:

```
<rubrica>
  <contatto>
    <nome>Marco</nome>
    <cognome>Rossi</cognome>
  </contatto>
</rubrica>
```

Nell'esempio l'elemento radice è <rubrica>. In un documento XHTML l'elemento radice deve essere **<html>**.

3. Namespace XHTML

L'elemento radice <html> deve contenere la dichiarazione di un namespace XML (spazio dei nomi) tramite l'attributo xmlns. Il namespace usato deve essere "http://www.w3.org/1999/xhtml".

4. Dichiarazione DOCTYPE

In un documento XHTML l'elemento radice deve essere preceduto da un elemento <!DOCTYPE>. All'interno di questo elemento è necessario specificare la DTD di riferimento e il suo URI.

La mia prima pagina XHTML

Esempio di codice XHTML Strict

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>
      La mia prima pagina XHTML
    </title>
  </head>
  <body>
    <h1>Benvenuto!</h1>
    <p>Questo è il mondo di XHTML!</p>
  </body>
</html>
```

Colore	Significato
rosso	il prologo
verde	l'elemento radice (<html>)
viola	l'intestazione (detta anche "testata") (<head>)
giallo	il corpo del documento

A questo punto è opportuno imparare a fare la convalida. All'url http://validator.w3.org è possibile effettuare la validazione di qualunque documento presente in rete: basta inserire l'URI della pagina e cliccare su "Validate this page".

Il prologo

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Il prologo di un documento XHTML risulta composto da due parti:

- la dichiarazione XML
- la definizione del DOCTYPE.

Dichiarazione XML

La nostra pagina inizia con una riga di codice: <?xml version="1.0"?>. La sua funzione è semplice: rendere esplicito il fatto che il documento è XML. **Non è obbligatoria**, ma è il suo uso è consigliato dal W3C per tutti i documenti XML. Quando viene usata non deve essere preceduta da altre istruzioni.

All'interno della dichiarazione è possibile usare tre attributi. L'unico obbligatorio è version (il solo valore possibile è "1.0", in quanto non esistono altre versioni del linguaggio). Un altro attributo, opzionale, ma spesso usato è encoding. Serve a specificare la codifica del linguaggio:

```
<?xml version="1.0" encoding="UTF-8"?>
```

L'ultimo attributo possibile è standalone. Se il valore è "yes" vuol dire che il documento non fa riferimento ad entità esterne.

Se il vostro obiettivo è la massima compatibilità potete omettere la dichiarazione XML. Molti browser hanno mostrato problemi così come alcuni editor (Dreamweaver). Se avete la necessità di specificare una codifica per i caratteri potete sempre usare un meta tag:

```
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8"/>
```

Definizione del DOCTYPE

La dichiarazione del DOCTYPE (obbligatoria!) è composta da due sezioni:

1. Un FPI (Identificatore Formale Pubblico) riferito ad una delle tre DTD XHTML
2. L'URI della DTD

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

Essa, dunque, ha lo scopo di stabilire a quale delle tre DTD XHTML intendiamo conformarci e di dire al browser dove essa è collocata. Nel nostro esempio la DTD di riferimento è quella Strict, collocata sul sito del W3C. Il DOCTYPE, inoltre, non ha alcun effetto sulla presentazione della pagina. Serve solo al validatore per stabilire le regole della convalida.

L'elemento radice

L'elemento radice di un documento XHTML deve essere <html>, che deve a sua volta contenere tutti gli altri elementi.

<HTML> è obbligatorio

Il namespace

L'elemento <html> può assumere questi attributi:

dir	Determina la direzione del testo
lang	Specifica il linguaggio di base dell'elemento quando è interpretato come HTML
xml:lang	Specifica il linguaggio di base dell'elemento quando è interpretato come XML
xmlns	Specifica il namespace predefinito per XHTML

L'unico attributo obbligatorio è xmlns. Il W3C, come visto, specifica anche il valore obbligatorio di tale attributo: "http://www.w3.org/1999/xhtml"

La testata del documento

```
<head>
  <title>
    La mia prima pagina XHTML
  </title>
</head>
```

Ecco l'elenco degli elementi che possono apparire nella testata:

<base>	Usato per definire l'URL di base della pagina. Utilissimo per la risoluzione dei link relativi.
<isindex>	Sconsigliato. È un modo per creare elementi simili alle caselle di testo.
<link>	Contiene informazioni su documenti esterni collegati. Usato soprattutto per i CSS.
<meta>	Specifica informazioni di vario tipo sul documento.
<noscript>	Usato per visualizzazioni alternative nei browser che non supportano gli script.
<object>	Racchiude un oggetto.
<script>	Contiene script di programmazione .
<style>	Definisce le regole di formattazione per il documento corrente
<title>	Specifica il titolo del documento che compare nella barra del titolo del browser

Di questi elementi l'unico richiesto nelle DTD XHTML 1.0 è title.

Il corpo del documento

```
<body>
  <h1>Benvenuto!</h1>
  <p>Questo è il mondo di XHTML!</p>
</body>
```

Il corpo del documento è la sezione in cui si sviluppa il contenuto. È racchiusa, come in HTML, tra i tag <body>...</body>. Gli elementi che possono comparire all'interno del corpo sono in genere suddivisi in due categorie: elementi blocco ed elementi inline.

Elementi blocco

Gli elementi blocco sono quelli che definiscono la struttura del documento. Possono contenere altri elementi blocco, elementi inline o testo. Quando sono inseriti danno origine ad una nuova riga nel flusso del documento. Se si inseriscono in una pagina queste due righe di codice:

```
<h1>Titolo</h1>
<p>Paragrafo</p>
```

il testo "Paragrafo" si troverà su una nuova riga, in quanto abbiamo inserito un nuovo elemento blocco (<p>).

Gerarchie e annidamento degli elementi blocco

Nella strutturazione del documento è fondamentale rispettare alcune semplici regole relative alla gerarchia e all'annidamento degli elementi blocco. Il primo elemento della gerarchia dovrebbe essere , che definisce in pratica una sezione del documento. Al suo interno trovano posto gli altri elementi. La cosa importante è che bisogna evitare annidamenti errati, che i browser fanno passare senza problemi, ma che il validatore segnala impietosamente in quanto violano le regole delle DTD.

Elementi inline

Gli elementi inline si distingono da quelli di tipo blocco per due motivi: quando sono inseriti non danno origine a una nuova riga e possono contenere solo dati (essenzialmente testo) o altri elementi inline. esempio:

```
<p>Questo tasto è<b>grassetto</b></p>
```

La parte delimitata dai tag ... non sarà posta su una nuova riga.

Attributi di body

Gli attributi per il testo, i link, il colore di sfondo e i margini dell'elemento <body> sono espressamente vietati solo nella DTD Strict,ma erano già considerati sconsigliati in HTML 4.0. Non vanno pertanto usati e devono essere sostituiti dai CSS.

Differenze con HTML

Gli elementi devono essere correttamente annidati

HTML	XHTML
`<b><i>un test</b></i>`	`<b><i>un test</i></b>`

Il primo esempio non è corretto in XHTML. Il tag <i> si sovrappone a <b>. La seconda colonna mostra invece un corretto annidamento degli elementi. La prima pratica è consentita in HTML. Certo, il browser dovrà interpretare qualcosa che è ambiguo, ma alla fine ci restituirà un testo in grassetto-corsivo (ciò che volevamo). In XHTML non possono sorgere ambiguità, tutto segue una regola.

I nomi di elementi e attributi devono essere in minuscolo

HTML	XHTML
`<TABLE><TR><TD>un test</TD></TR></TABLE>`	`<table><tr><td>un test</td></tr></table>`

Gli elementi non vuoti devono essere chiusi

HTML	XHTML
`<p>Test1` `<p>Test2`	`<p>Test1</p>` `<p>Test2</p>`

I valori degli attributi devono essere posti tra virgolette

HTML	XHTML
`<img src=test.gif>`	`<img src="test.gif">`

Ogni attributo deve avere un valore

Alcuni attributi di HTML non hanno un valore (si dice che sono standalone). È il caso dell'attributo selected usato per identificare l'opzione di un menu a tendina selezionata all'apertura del documento. In XHTML anche questi attributi devono essere valorizzati.

HTML	XHTML
`<option selected>test</option>`	`<option selected="selected">test</option>`

Gli elementi vuoti devono terminare con />

In XML tutti i tag devono essere chiusi. Ma cosa fare con gli elementi vuoti? Si tratta di quelli che non possono contenere nulla al loro interno ma semplicemente ricevere attributi: <meta>,
, <hr>, <img> sono i più comuni. In XHTML tali elementi vengono chiusi terminando la dichiarazione con />.

HTML	XHTML
` ` `<img src="test.gif">`	` ` `<img src="test.gif"/>`

Uso di `id` e `name`

Per identificare un elemento si deve usare l'attributo id e non name. Apparantemente la cosa non fa una grinza. In questo modo si ha una perfetta conformità con XML dove id è l'attributo standard per l'identificazione dei frammenti. In realtà qui il cambiamento con HTML è notevole, perché per elementi

come <img> o <a> l'attributo di identificazione è proprio name. Il passaggio a id non pone problemi nei browser più recenti, ma con altri (Netscape 4, per esempio) non funziona. In questo caso e se la compatibilità all'indietro è assolutamente necessaria, la stessa specifica XHTML 1.0 suggerisce di usare entrambi gli attributi, anche se ciò è contro le regole:

HTML	XHTML
<a name="test>	<a id="test"> Per compatibilità: <a id="test" name="test">

Nelle specifche successive (cosa che è accaduta con XHTML 1.1) l'attributo name è stato abolito completamente.

Gli script in XHTML

Uso di <script>

In HTML il tag <script> serve a incorporare nel documento codice di programmazione. Il linguaggio più comunemente usato è Javascript. Il tag è supportato anche in XHTML e va ugualmente inserito nella sezione <head>....</head>. L'elemento <script> supporta i seguenti attributi:

charset	Specifica la codifica dei caratteri
defer	Dice al browser che lo script non genera documenti
langauage	Specifica il linguaggio di scripting. È obbligatorio quando l'attributo src non è specificato.
src	Contiene l'URL di uno script contenuto in un file esterno
type	Obbligatorio. Indica il tipo MIME dello script: es "text/javascript"
xml:space	Usato per mantenere la spaziatura del codice

Dunque, uno script può essere direttamente incorporato nella pagina oppure inserito in un file esterno e richiamato:

Script incorporato

```
<script type="text/javascript"
language="javascript">
<!--
document.open()
document.writeln("Questo è XHTML!")
document.close()
//-->
</script>
```

Script collegato

```
<script type="text/javascript"
src="mioscript.js">

</script>
```