

HTML 4.0

HTML 4.0 estende HTML con nuovi 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.).
- **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).

La progettazione di documenti con HTML 4.0

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, più 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 pagine richiedono gli autori di considerare appropriatamente le specifiche esigenze di tutti.

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 support per l'internazionalizzazione passa attraverso 3 meccanismi:

- **Il Set di Caratteri:** vedendo l'insufficienza di ASCII, HTML adotta dal 4.0 il set di caratteri denominato UCS (*Universal Character Codes*), che è uguale ma non equivalente a Unicode 2.0. Secondo questo standard, ogni carattere di ogni alfabeto esistente o desiderato (incluso il Klingon!) può essere rappresentato da una sequenza di byte. Poiche' per rappresentare questi caratteri é necessario usare uno o più byte a seconda del linguaggio, è opportuno prevedere, o nell'header Content-type, o come tag META nell'intestazione del documento, il set di caratteri adottato (e quindi la lunghezza in byte di ogni carattere).

```
Content-Type: text/html; charset=Latin-1
```

oppure

...

```
<META http-equiv="Content-Type" content="text/html; charset=Latin-1">
```

...

- **L'attributo lang:** permette di specificare, tag per tag, il linguaggio di base del contenuto dell'elemento. Si usa un codice a due lettere (it: italiano, en: inglese, fr:francese, ecc.)

```
<P lang="it">I Beatles ottennero il primo successo con la  
canzone <Q lang="en">Love me do</Q>, nel 1962</P>
```

L'attributo lang è pensato per aiutare i motori di ricerca, sintetizzatori di voce, verificatori di grammatica, ecc., che debbono conoscere la lingua degli elementi per trattarli deguatamente.

- **L'attributo dir:** permette di specificare, tag per tag, la direzionalità del contenuto dell'elemento. Serve per documenti che contengono un misto di elementi con direzionalità da sinistra a destra (valore LTR) e da destra a sinistra (RTL), come ebraico, arabo, ecc. Ad esempio, una citazione in ebraico sarà:

```
<Q lang="he" dir="RTL"> ... citazione </Q>
```

Meccanismi piuttosto complicati sono descritti per permettere la scrittura, nello stesso blocco, di testi con allineamento LTR e pezzi RTL.

- **Il tag BDO (di-directional override):** Permette di superare le regole imposte dall'algoritmo tradizionale di convivenza tra testi LTR e testi RTL.

Nuovi e vecchi tag in HTML 4.0

Sono definiti alcuni nuovi tag in HTML:

- **Q**: elemento in-line (cioè dentro ad un blocco) che identifica una citazione. Lo user agent dovrebbe fornire le virgolette o quant'altro appropriato per la corretta identificazione dell'elemento.
- **INS e DEL**: elementi in-line che identificano operazioni di editazione di un testo. INS serve per identificare parti di testo che sono state inserite, DEL parti di testo che sono state cancellate. Utile per testi di legge e cose simili dove é importante identificare a colpo d'occhio le modifiche.
- **DIV e SPAN**: rappresentano gli elementi generici e minimali di composizione di elementi testuali. Sono definiti come l'elemento di blocco (DIV) e l'elemento in-line (SPAN) privi di caratteristiche semantiche o tipografiche predefinite. Tramite stylesheet è possibile fornire loro un effetto grafico preciso. E' possibile associare dunque ogni elemento tipografico di HTML ad un appropriato DIV o SPAN senza perdita di resa grafica.

Estensioni delle tabelle

Sono stati aggiunti vari elementi per rendere più sofisticate le tabelle. In particolare si sono aggiunte delle sezioni per esplicitare le righe della tabella che ne costituiscono l'intestazione, il corpo e la coda. Inoltre si è inserito un meccanismo per gestire le caratteristiche delle colonne in maniera integrata ed unita.

THEAD, TFOOT, TBODY: Elementi di raggruppamento delle righe delle tabelle. Permettono di dividere logicamente gruppi di righe e di specificare in maniera centralizzata aspetti comuni a questi gruppi. Una tabella allora è composta di un gruppo di righe di intestazione, un gruppo di righe di conclusione, e uno o più gruppi di righe di corpo. La struttura è come segue:

```
<TABLE>
  <THEAD><TR> ...header...</THEAD>
  <TFOOT><TR> ...footer...</TFOOT>
  <TBODY><TR> ...first row of block one data...</TBODY>
  <TBODY>
    <TR> ...first row of block two data...
    <TR> ...second row of block two data...
  </TBODY>
</TABLE>
```

COL e **COLGROUP**: Elementi di descrizione delle colonne della tabella. Permettono di specificare l'organizzazione strutturale e aspetto grafico delle colonne in maniera centralizzata. Prima della descrizione delle righe della tabella (che sono comunque necessarie), posso utilizzare gli elementi COL e COLGROUP per organizzare centralmente gli aspetti comuni a tutte le celle di una o più righe.

```
<TABLE>
  <COLGROUP span="10" width="50">
  <COLGROUP span="5" width="0*">
  <THEAD>
  <TR><TD> ...
</TABLE>
```

L'esempio qui contiene quindici colonne, di cui le prime dieci hanno larghezza prefissata in 50 pixel, e le rimanenti si dividono equamente lo spazio rimasto.

```
<COLGROUP id="format1">
  <COL width="20" >
  <COL width="30">
</COLGROUP>
```

Questo esempio contiene un gruppo di due colonne da formattare nella stessa maniera, ma in cui una è larga 20 pixel, e l'altra 30.

Un esempio di tabella in HTML 4.0

```
<TABLE border="2" frame="hsides" rules="groups">
  <CAPTION>CODE-PAGE SUPPORT IN MICROSOFT WINDOWS</CAPTION>
  <COLGROUP align="center">
  <COLGROUP align="left">
  <COLGROUP align="center" span="2">
  <COLGROUP align="center" span="3">
  <THEAD valign="top">
  <TR>
    <TH>Code-Page<BR>ID<TH>Name<TH>ACP<TH>OEMCP
    <TH>Windows<BR>NT 3.1<TH>Windows<BR>NT 3.51
    <TH>Windows<BR>95
  <TBODY>
  <TR><TD>1200<TD>Unicode...
  ...
  <TR><TD>1361<TD>Korean...
  <TBODY>
  <TR><TD>437<TD>MS-DOS United States<TD><TD>X<TD>X<TD>X<TD>X
  ...
</TABLE>
```


Il risultato

CODE-PAGE SUPPORT IN MICROSOFT WINDOWS

Code-Page ID	Name	ACP	OEMCP	Windows NT 3.1	Windows NT 3.51	Windows 95
1200	Unicode (BMP of ISO 10646)			X	X	*
1250	Windows 3.1 Eastern European	X		X	X	X
1251	Windows 3.1 Cyrillic	X		X	X	X
1252	Windows 3.1 US (ANSI)	X		X	X	X
1253	Windows 3.1 Greek	X		X	X	X
1254	Windows 3.1 Turkish	X		X	X	X
1255	Hebrew	X				X
1256	Arabic	X				X
1257	Baltic	X				X
1361	Korean (Johab)	X			**	X
437	MS-DOS United States		X	X	X	X
708	Arabic (ASMO 708)		X			X
709	Arabic (ASMO 449+, BCON V4)		X			X
710	Arabic (Transparent Arabic)		X			X
720	Arabic (Transparent ASMO)		X			X

I tag OBJECT e IFRAME

HTML era oppresso da un numero sempre più grande di tag speciali per includere oggetti non HTML all'interno del documento: IMG, APPLET, EMBED, BGSOUND, ecc.

Si è deciso allora di unificarli tutti in un solo tag, il tag OBJECT. Questa unificazione passa attraverso due stadi:

- HTML 4.0: si ignorano tutti i tag non standard precedenti (lasciando solo IMG e APPLET), e si “depreca” APPLET, introducendo al suo posto il tag OBJECT
- Si depreca IMG stesso (HTML 5?)

OBJECT: il tag OBJECT serve sia per includere immagini, sia applet che suoni, video, animazioni, e tutti gli oggetti che richiedono “plug-in”. I tag OBJECT possono annidarsi per fornire alternative di qualità in degradazione controllata:

```
<OBJECT title="The Earth from space" classid="http://www.obs.mars/TheEarth.class">
  <!-- MPEG --> <OBJECT data="TheEarth.mpeg" type="application/mpeg">
  <!-- GIF --> <OBJECT data="TheEarth.gif" type="image/gif">
  <!-- testo --> The <STRONG>Earth</STRONG> as seen from space.
</OBJECT> </OBJECT> </OBJECT>
```

IFRAME: Se vogliamo includere un documento HTML all'interno di un altro, possiamo includerlo sia con OBJECT che con IFRAME. La differenza è che IFRAME può essere target di un link, e che IFRAME non può annidarsi.

OBJECT può essere usato anche per fare image map su immagini, animazioni, applet ed altro. L'attributo shapes permette di specificare che l'immagine è un image map.

```
<OBJECT data="navbar.gif" shapes>
  <A href="guide.html" shape="rect" coords="0,0,118,28">Access Guide</a> |
  <A href="shortcut.html" shape="rect" coords="118,0,184,28">Go</A> |
  <A href="search.html" shape="circle" coords="184,200,60">Search</A> |
  <A href="top10.html" shape="poly" coords="276,0,373,28,50,50,276,0">Top Ten</A>
</OBJECT>
```

Stylesheet

Gli stylesheet rappresentano un notevole progresso sull'uso di tag (più o meno proprietari) per specificare comportamenti grafici precisi.

Gli stylesheet permettono di evitare i trucchi sporchi fin qui usati per forzare certi comportamenti grafici ai documenti HTML, e di estendere comunque le capacità grafiche offerte. Inoltre permette di separare nettamente l'aspetto di strutturazione del documento dall'aspetto di organizzazione grafica.

Esistono vari meccanismi per specificare il comportamento grafico. Il più importante è Cascading Style Sheet (CSS), che è anch'esso uno standard W3C.

HTML 4.0 permette :

- posizionamento flessibile delle informazione di stile,
- indipendenza da qualunque linguaggio di stile,
- propagazione a cascata delle specifiche,
- dipendenza dai media
- Stili alternativi.

La sintassi dello stylesheet non è HTML, quindi bisogna permettere un certo numero di soluzioni alternative. Ad esempio, la seguente è un'istruzione in CSS:

```
P {
  color : green;
  border: solid red;
}
```

Ci sono vari modi per specificare lo stylesheet assegnato ad un documento.

Intestazioni HTTP: Tra le intestazioni del documento HTML viene anche specificato il documento di stylesheet.

```
HTTP /1.0 100 Ok
Server: XX
Date: XX
Content-Type: text/html
Content-Length: 1254
Content-Style-Type: text/css
Link: <http://www.acme.com/corporate.css>; REL=stylesheet

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <BODY>
    <P>This paragraph should have special green text.</P>
  </BODY>
</HTML>
```

Finte intestazioni HTTP con il tag META: Il modo è di fatto equivalente al precedente, ma non richiede modifiche al server:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <META http-equiv="Default-Style" content="special.css">
    <META http-equiv="Link"
      content="<http://www.acme.com/corporate.css>; REL=stylesheet">
  </HEAD>
  <BODY>
    <P>This paragraph should have special green text.</P>
  </BODY>
</HTML>
```

Documento separato: Lo stylesheet è memorizzato in un documento autonomo, ed il documento HTML vi fa riferimento con un elemento LINK:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <LINK href="special.css" rel="stylesheet" type="text/css">
  </HEAD>
  <BODY>
    <P>This paragraph should have special green text.</P>
  </BODY>
</HTML>
```

Intestazione: Lo stylesheet è memorizzato all'interno del blocco HEAD del documento HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <STYLE type="text/css">
      P {
        color : green;
        border: solid red;
      }
    </STYLE>
  </HEAD>
  <BODY>
    <P>This paragraph should have special green text.</P>
  </BODY>
</HTML>
```

Questo è il modo più semplice e sicuro.

Inserito nel testo: ogni elemento di HTML permette adesso l'attributo style. Come contenuto di questo elemento è possibile specificare una regola di stile:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <STYLE type="text/css">
      P {
        color : green;
        border: solid red;
      }
    </STYLE>
  </HEAD>
  <BODY>
    <P style="color: green; border: solid red;">
      This paragraph should have special green text.
    </P>
  </BODY>
</HTML>
```


Gli attributi id e class

Permetto di dare un identificativo univoco all'elemento, o di specificare che appartiene ad una classe prestabilita, in modo da assegnargli lo style in maniera opportuna

```
<HEAD>
  <STYLE type="text/css">
    P { color : green; border: solid red; }
    P.myclass {color: yellow; border: solid black ; }
    #myid { text-align: center}
  </STYLE>
</HEAD>
<BODY>
  <P> Questo P è verde con bordo rosso </P>
  <P class="myclass"> Questo P è giallo con bordo nero </P>
  <P id="myid"> Questo P è verde, con bordo rosso e allineato al centro </P>
  <P id="myid" class="myclass">
    Questo P è giallo, con bordo nero e allineato al centro
  </P>
</BODY>
```