



CSS3 – Lo stato dell'arte

DARIO DE JUDICIBUS

È in discussione il livello 3 dello standard CSS. La nuova versione si presenta ricca di novità, soprattutto per quello che riguarda il livello di controllo sull'impaginazione degli elementi e la gestione degli alfabeti orientali.

Sono passati ben otto anni da quando sono state pubblicate per la prima volta le specifiche dei fogli di stile per il linguaggio HTML, ovvero i Cascading Style Sheets livello 1, successivamente riviste e ripubblicate nel novembre del 1999 (<http://www.w3.org/TR/CSS1>), sei dalla pubblicazione delle prime specifiche del livello 2, consolidato definitivamente come livello 2.1 agli inizi del 2004.

I fogli di stile hanno rappresentato una svolta significativa nel modo di disegnare e realizzare siti web. La separazione fra struttura del documento e presentazione dello stesso è fondamentale in un sistema dove la condivisione delle informazioni deve prescindere dai canali attraverso i quali queste devono essere diffuse. Inizialmente il linguaggio HTML comprendeva sia elementi concettuali, come `<P>` o `<H1>`, sia elementi di presentazione, come `<FONT>`. Inoltre diversi elementi, come `<TABLE>`, ad esempio, includevano sia aspetti di natura strutturale e concettuale che di natura editoriale. Ancora oggi sia l'HTML 4.01 che il più recente XHTML 1.0 mantengono molte delle caratteristiche delle prime versioni dell'HTML, più che altro per motivi di compatibilità, ma l'introduzione dei CSS e l'utilizzo di tecniche di generazione delle pagine dal lato server hanno cambiato radicalmente il modo di sviluppare i siti web.

C'è da dire che non si è trattato di un cambiamento indolore, soprattutto per la lentezza con la quale alcune applicazioni hanno adottato il nuovo standard. La non completa conformità alle specifiche di alcuni browser, interpretazioni non corrette delle stesse — leggi box model — e veri e propri errori nel codice hanno reso difficile questa transizione. Ancora oggi chi sviluppa codice HTML basato sui fogli di stile deve fare i conti con tutta una serie di incompatibilità che spesso sono risolvibili solo con trucchi che sfruttano a loro volta determinati banchi nel codice dei browser.

Inoltre la versione attuale dei fogli di stile, la CSS 2.1, non permette ancora alcuni meccanismi di impaginazione il che costringe a sviluppare codice la cui manutenzione è comoda solo se si generano le pagine sul lato server, a meno di non ricorrere a ulteriori linguaggi di programmazione, come il JavaScript. Ad esempio, le classiche strutture con intestazione, barra di navigazione orizzontale, menù laterale e piè di pagina che una volta si ottenevano con i frame, possono essere prodotte in modo molto più efficiente con l'elemento `<DIV>` e i fogli di stile. Questa tecnica, tuttavia, costringe a riproporre in più pagine gli stessi blocchi di codice. Se si usa un meccanismo di generazione sul lato server, come ASP o PHP, o anche semplicemente ci si può avvalere di un meccanismo di inclusione, questo non rappresenta un problema, dato che si possono mantenere comunque i vari blocchi in file separati, ma se si deve lavorare solo sul lato cliente, allora si ha una duplicazione di codice che, per siti di grosse dimensioni, può essere inaccettabile. Inoltre, se si desidera mantenere una forte compatibilità anche con versioni più vecchie dei browser, si devono introdurre nel codice alcuni dei trucchetti summenzionati (hack) o meglio ancora sviluppare fogli di stile differenti a seconda del browser utilizzato dal visitatore delle pagine.

Nel livello 1 del CSS, la larghezza e l'altezza complessiva di un elemento a blocco rettangolare viene calcolata sommando la dimensione, il sovrappiù e il bordo. La versione 5 di Internet Explorer, tuttavia, include nella larghezza fornita anche il sovrappiù e il bordo, lasciando fuori solo il margine. Così la definizione

```
div#scatola {  
    width: 300px ;  
    padding: 10px ;  
    border: 2px dotted red;  
    margin: 20px;  
}
```

produce un elemento largo $300 + 2 \times (10 + 2) = 324$ pixel in tutti i browser conformi al CSS1, ma di soli 300 pixel in IE5.x. Esistono vari trucchi (hack) per risolvere il problema. Uno di questi consiste nello scrivere lo stile della divisione con identificativo scatola nel seguente modo (Simplified Box Model hack):

```
div#scatola {  
    width: 300px ; \width: 324px; w\idth: 300px;  
    padding: 10px ;  
    border: 2px dotted red;  
    margin: 20px;  
}
```

Il trucco funziona perché `width` è letta da quei browser che sono conformi al CSS1 ma ignorano i caratteri di escape, come Opera 5; `\width` è invece letta da IE5.x che tuttavia ignora la terza parte della riga; infine `w\idth` è letta da tutti browser più moderni, ovvero dalle versioni 6 o successiva di Netscape, Opera e Internet Explorer, oltre che da Mozilla e Firefox.

In conclusione, utilizzare i fogli di stile richiede una certa competenza e una buona conoscenza del loro comportamento nei vari browser. Internet è pieno di tabelle di compatibilità dei browser rispetto ai vari livelli di CSS, così come di suggerimenti e meccanismi per risolvere tutta una serie di specifici problemi. È evidente, tuttavia, come una situazione di questo tipo non possa essere accettata nel lungo termine. Lo

sviluppatore deve potersi concentrare sul disegno e la realizzazione del sito, non impantanarsi per risolvere tutta una serie di problemi che in fondo nascono da bachi o scelte di non conformità da parte di chi ha realizzato i browser. Ecco allora che con il livello 3 si è deciso di suddividere lo standard in una serie di moduli. Ogni browser potrà dichiararsi conforme a uno o più moduli, evitando, se lo riterrà opportuno, di implementare lo standard in modo esaustivo. Ovviamente ci sono alcune dipendenze fra i vari moduli, ma il principio non cambia. Questa scelta è dovuta a due ragioni: la prima è che, con il terzo livello, lo standard è diventato particolarmente complesso e quindi è verosimile pensare che ci possano essere alcune difficoltà a realizzarlo pienamente, soprattutto su periferiche semplici come palmari e smartphone; la seconda è che il nuovo livello pone particolare accento su alcuni aspetti di internazionalizzazione e di gestione di modi di presentazione alternativi allo schermo che possono non interessare tutte le implementazioni. Ma di questo parleremo in un altro articolo.

Prima di continuare, tuttavia, è opportuno fare una precisazione: il livello 3 è in piena fase di disegno, ovvero è ben lungi dall'essere consolidato, per cui molte delle novità di cui parleremo in questo e altri articoli potranno subire ulteriori modifiche se non addirittura essere completamente rivoluzionate. Iniziare a ragionarci sopra può tuttavia essere utile, sia per avere un'idea di dove sta andando quello che oggi è uno degli standard più importanti del W3C, assieme all'XML, sia perché si è ancora in tempo per fare proposte e chiedere modifiche, o quantomeno intervenire nella discussione con idee, suggerimenti e questioni.

Vediamo allora lo stato dell'arte, ovvero come si presenta il livello 3 dello standard CSS al momento. In un secondo articolo entreremo più approfonditamente nei cambiamenti più significativi di quelli che possono essere considerati i moduli più importanti, soprattutto in relazione all'impatto che avranno nel modo di disegnare le pagine. Il terzo articolo si focalizzerà su un gruppo molto consistente di novità che riguardano la gestione dei linguaggi non occidentali, soprattutto quelli orientali e quello arabo. I tre articoli saranno comunque del tutto indipendenti l'uno dall'altro. Anche così, solo una parte delle tante modifiche apportate al livello 2 potranno essere affrontate, tuttavia ha poco senso in questo momento entrare in un dettaglio maggiore dato che siamo appena agli inizi di una nuova revisione dello standard CSS.

Iniziamo dai moduli. Al momento sono ben 26, il che già la dice lunga sull'impresa che il W3C si accinge a compiere con il CSS3. La maggior parte non sono neanche in fase di bozza, ovvero non si è ancora iniziato a discuterne approfonditamente. È tuttavia verosimile che alcuni saranno consolidati in un modulo centrale, o Core, la cui implementazione sarà obbligatoria per tutti i browser che intendono essere conformi al livello 3. Qui di seguito useremo il termine browser per indicare quello che in W3C si chiama più propriamente user agent (UA), ovvero un'applicazione lato client in grado di rappresentare un codice HTML o XML su una specifica periferica. Ovviamente un UA può anche non avere affatto l'aspetto di un browser. Ad esempio può servire per la riproduzione

audio di un documento HTML o stampare direttamente lo stesso su diapositiva o microfilm.

I quattro moduli principali riguardano ovviamente la sintassi, i selettori, le unità di misura, i valori degli attributi e i meccanismi di ereditarietà fra stili. In particolare i selettori sono stati sensibilmente potenziati. Oltre ad avere aggiunto tutta una serie di nuove pseudo-classi e di averle ancor meglio differenziate dai cosiddetti pseudo-elementi, già introdotti nel livello 2.1, viene esplorata in pieno la possibilità di effettuare una vera e propria query, analoga a quelle usate nel linguaggio XML Path, per identificare un determinato elemento. Questo meccanismo era già stato introdotto nel livello 2 e ha suscitato parecchio interesse negli sviluppatori, tanto che ci sono già parecchi suggerimenti su come arricchirlo ulteriormente. Vedremo nel prossimo articolo quanto è potente questo meccanismo limitandoci a riportare qui alcuni esempi.

Vogliamo selezionare tutti i paragrafi allineati al centro che siano all'interno di una divisione, direttamente o su un livello più basso di annidamento. Il selettore sarà:

```
div * p[align="center"]
```

Vogliamo selezionare tutti gli elementi che abbiano un attributo **src** e che siano inclusi direttamente in un collegamento. Il selettore sarà:

```
a > [src]
```

Vogliamo ora selezionare tutte le intestazioni **h3** immediatamente successive ad una **h2** il cui testo sia in francese. Il selettore sarà:

```
h2[lang=fr] + h3
```

Vogliamo selezionare tutti i collegamenti a siti aventi dominio italiano. Il selettore sarà:

```
a[href*=".it"]
```

Vogliamo selezionare tutte le righe pari di una tabella. Il selettore sarà:

```
tr:nth-child(2n)
```

Vogliamo far sì che in documento tutte le immagini si alternino a destra e a sinistra del testo. Basterà allora scrivere

```
img:nth-of-type(2n+1) { float: right; }
```

```
img:nth-of-type(2n) { float: left; }
```

Vogliamo aggiungere davanti ad ogni paragrafo di tipo nota il testo "**Nota: **". Dovremo allora scrivere:

```
p.nota:before { content: "Nota:&nbsp;" }
```

Altri quattro moduli fondamentali sono quelli che riguardano il box model, che tanti problemi ha creato in passato a causa di differenti interpretazioni dello standard, il posizionamento degli elementi, lo sfondo e le tabelle. A queste ultime si affianca un nuovo modulo che finalmente affronta il delicato problema degli elementi a più colonne. Molto interessante è in particolare la possibilità di far scorrere il testo all'interno delle colonne come avviene, ad esempio, in un quotidiano.

Supponiamo di voler creare una divisione formata da quattro colonne e supponiamo che ogni colonna debba essere larga almeno **80px**, separata dalle altre da almeno **5px** e che la divisione abbia una larghezza complessiva di **400px**. Scriveremo allora

```
div {  
  width: 400px;  
  column-count: 4;  
  column-width: 80px;  
  column-gap: 5px;  
  column-rule: none;  
  column-width-policy: flexible;  
}
```

dove l'ultima proprietà permette al browser di allargare le colonne in modo da coprire l'intera larghezza di **400px**. Se invece avessimo specificato **strict**, allora avremmo avuto **25px** di spazio in eccedenza che avremmo dovuto posizionare in qualche modo. Questo può essere fatto utilizzando la proprietà **column-space-distribution**. Inoltre, grazie alla proprietà **column-span** è possibile permettere a determinati elementi nel testo, come un'immagine o l'intestazione di un sottocapitolo, di attraversare più colonne. Il resto del testo invece scorrerà nelle varie colonne come succede in un giornale, ovvero, arrivato in fondo alla colonna riprenderà in cima alla colonna successiva. Eventuali elementi trasversali a più colonne possono tuttavia alterare questo flusso.

Inoltre, in aggiunta al box model tradizionale, è stato introdotto un secondo modulo dedicato al box model degli elementi in linea. Può sembrare strano pensare a un modello per questo genere di elementi, dato che finora il box model è sempre stato dedicato più che altro ai blocchi, eppure questo modulo va a riempire un vuoto che le vecchie specifiche avevano lasciato. Questo nuovo modulo, infatti, si occupa ad esempio del modo nel quale una linea di testo va a capo ed è molto importante quando in un documento vengono utilizzati alfabeti differenti. Sempre in relazione ad alcuni alfabeti orientali è stato introdotto un modulo che si occupa di elementi detti *ruby*, ovvero piccole linee di testo che affiancano o si pongono sopra o sotto il testo principale. È interessante notare come questo meccanismo possa essere utilizzato anche negli alfabeti occidentali per posizionare ad esempio gli accordi sopra il testo di una canzone.

Il modulo forse più interessato dai cambiamenti resta tuttavia quello relativo al testo. Il grado di controllo che il CSS3 dovrebbe fornire allo sviluppatore sembra essere ora molto sofisticato, praticamente quasi sullo stesso livello di quello fornito da linguaggi editoriali avanzati come l'SGML e TeX. Probabilmente questo comporterà una sempre maggiore difficoltà a scrivere del buon codice CSS senza l'ausilio di opportuni strumenti, ma questo è il prezzo che si paga sempre quando si rende troppo sofisticato un linguaggio, e comunque i programmatori moderni sono abituati da tempo ad utilizzare strumenti visuali. Sempre relativamente al testo ci sono poi due moduli dedicati a eventuali contenuti generati o modificati direttamente via CSS. Questo aspetto ovviamente introduce nel foglio di stile elementi in uno specifico linguaggio naturale, il che renderà il file CSS "localizzato". Si tratta di una scelta le cui conseguenze sono tutte da valutare.

Fra le novità relative al testo, ci sono tutta una serie di proprietà che riguardano la sottolineatura, la sovrallineatura e la linea passante. È infatti possibile specificare lo stile della linea. Ad esempio:

```
text-underline-style: dot-dashed;  
text-underline-color: rgba(240,240,0,0.75);  
text-underline-mode: skip-white-space;  
text-underline-width: thin;
```

Anche per l'ombreggiatura è possibile creare effetti molto complessi. Ad esempio

```
text-shadow: 2px 2px brown, yellow -4px 4px 6px, 3px -3px;
```

permette di creare tre ombre: la prima, di due pixel a destra e in basso, è di colore marrone; la seconda, gialla, è di quattro pixel a sinistra e altrettanti in basso, ma ha un effetto di sfocatura di **6px**; l'ultima è di tre pixel a destra e tre in alto e utilizza per l'ombra lo stesso colore utilizzato per rendere il testo. Le ombre successive ovviamente si sovrappongono a quelle precedenti, per cui se i colori specificati contengono una parziale trasparenza, come nel caso di **rgba(80%,80%,20%,50%)**, si possono ottenere effetti grafici molto interessanti.

Ci sono poi alcuni moduli specificatamente focalizzati sull'aspetto estetico dei vari elementi, come quello relativo ai colori, ai tipi di carattere e ai caratteri agganciabili direttamente dalla rete, caratteristica questa che dovrebbe permettere l'utilizzo di qualsiasi tipo di carattere in un sito mantenendone la rappresentazione anche su piattaforme differenti e senza bisogno che il tipo in questione sia disponibile sul sistema locale. Un altro concetto introdotto dal CSS3 è quello dell'aspetto (appearance), che estende quanto già si era iniziato a fare nel CSS2.1 quando si era data la possibilità di associare a determinati elementi le caratteristiche estetiche di alcuni controlli del sistema operativo locale sul quale il documento veniva rappresentato. Questo permette di costruire vere e proprie interfacce utente che abbiano lo stesso stile di quelle su cui gira il browser. Ad esempio, si potrà dire, «dai a questo testo l'apparenza e il comportamento di un pulsante», oppure, «rendi questo campo di immissione identico a quelli tipici del sistema su cui si stanno visualizzando le pagine». C'è quindi un modulo specifico per la realizzazione di interfacce utenti.

Particolare attenzione è stata data alla creazione di vere e proprie interfacce utente. È stato introdotto il concetto di apparenza, ovvero è ora possibile assegnare a qualunque elemento le caratteristiche tipiche di un particolare controllo del sistema operativo; sono stati aggiunti nuovi cursori; sono state infine aggiunte proprietà per gestire la navigazione fra i vari elementi interattivi e per controllarne il comportamento a fronte di specifici eventi. Ad esempio, se si vuole creare una finestra di dialogo non ridimensionabile, come si fa solitamente per molti pannelli nelle applicazioni, basterà scrivere

```
html {  
  width: 300px;  
  height: 200px;  
  overflow: clip;  
  resizer: none;  
}
```

Se invece vogliamo definire un pulsante di conferma simile a quello usato dal sistema operativo, con un contorno più spesso, forzando la scritta interna a "Va bene" nel caso il pannello sia in italiano e con lo stesso comportamento del classico pulsante "OK", allora potremmo scrivere qualcosa del genere:

```
input[type=button].ok:lang(it) {
```

```
display: inline-block;
appearance: push-button;
outline-width : thick;
user-select: none;
key-equivalent: system-ok;
content: "Va bene";
}
```

Un modulo molto importante riguarda l'impaginazione del documento. Impaginare un documento vuol dire dare al documento una struttura a pagine, come avviene in un libro o in un giornale. In genere noi siamo abituati a pensare i siti web in modo differente. Abbiamo sì il concetto di pagina web, ma questa è variabile, con componenti che possono cambiare sia in lunghezza che larghezza. In effetti una pagina web coincide in genere con un documento HTML. Se però stampiamo quella pagina, allora il contenuto va suddiviso su più pagine le cui caratteristiche sono definite dalle dimensioni fisiche dei fogli su cui si sta effettuando la stampa. In realtà il concetto di impaginazione è applicabile anche agli schermi del PC, basti pensare alle presentazioni multimediali, in cui ogni diapositiva è di fatto una pagina e tutte le diapositive hanno le stesse dimensioni e più o meno la stessa struttura. Proprio per facilitare lo sviluppo di presentazioni basate su HTML/XML e CSS, è stato introdotto inoltre un modulo specifico, chiamato appunto modulo di presentazione che permette da una parte di gestire le transizioni visive in una presentazione multimediale, dall'altra di visualizzare solo alcuni livelli di un documento strutturato secondo una determinata gerarchia. Ad esempio sarà possibile mostrare solo i capitoli di un documento, o di far apparire o sparire note a margine.

Ci sono infine tutta una serie di moduli molto particolari che si affiancano a linguaggi XML specializzati come il MathML, l'SVG e lo SMIL, un modulo che riguarda la riproduzione audio di parte o di tutto un documento e uno che prevede la possibilità di associare a un certo elemento uno specifico "comportamento". Quest'ultimo modulo, unito a quello per le interfacce utente, permetterà di sviluppare pagine web interattive molto più sofisticate di quelle attuali senza ricorrere ad applet Java o ad altri linguaggi di programmazione. Ovviamente diventa essenziale a questo punto poter sapere su quale tipo di periferica la pagina verrà visualizzata. Per questo motivo è stato definito un modulo apposito che prevede di poter richiedere tale informazione completata da una serie di dati caratteristici della periferica in questione come le dimensioni dello schermo, se applicabile, il numero di colori visualizzabili e via dicendo.

Concludendo, la versione 3 del CSS si presenta come estremamente potente e innovativa, permetterà un controllo fine del posizionamento del testo e dei vari elementi di una pagina, supporterà appieno alfabeti e modalità di scrittura differenti da quella latina estesa, permetterà di rappresentare impaginazioni molto complesse e di generare vere e proprie interfacce grafiche per l'interazione con l'utente finale... nella speranza che questa volta chi sviluppa i browser presti una maggiore attenzione all'implementazione delle specifiche.