

## Realizzare siti internet con css

La ratifica della specifica del livello 1 dei CSS da parte del W3C è datata 1996, ma, solo negli ultimi tre anni i fogli di stili offrono ai web designer la possibilità di **realizzare siti internet flessibili e accessibili**. I nuovi browsers, infatti, anche se con qualche piccola differenza, supportano gli attributi di posizionamento e gli elementi di formattazione.

La divisione dei contenuti dal layout è uno dei più importanti punti di forza dei css ed è **possibile creare più layout per dispositivi di output differenti con una sola fonte dati**:

**XHTML + CSS = output**

**XSLT + XML + CSS = output**

È possibile **realizzare siti con enorme flessibilità**: cambiando il valore della proprietà che setta il colore di sfondo del tag body può cambiare tutto il sito. Il sito [Zengarden](#) è tra gli esperimenti più riusciti ed è diventato punto di riferimento e ispirazione per migliaia di web designer.

Di seguito, propongo qualche semplice layout (che cercherò di ampliare lentamente) e parte del Vademecum CSS di David Shea (famoso web designer autore del blog [Mezzoblue](#)). La traduzione completa in italiano è curata da [Nousab](#).

### Qualche semplice layout con i css:



---

## Vademecum CSS

### 1. **Quando avete dubbi, validate il codice**

Mentre state facendo il debug del vostro sito, potete risparmiarvi il mal di testa semplicemente validando per prima cosa il vostro codice. L'accoppiata di XHTML e CSS non validi può portare a molti problemi di visualizzazione.

### 2. **Costruite e controllate i vostri CSS riferendovi ai browser più avanzati *prima* di provarli sugli altri, non dopo.**

Se disegnate un sito testandolo su un browser pieno di bug, il vostro codice si baserà sin dall'inizio sulla visualizzazione *fallata* di quel browser. Quando poi testerete il sito su un browser che rispetta maggiormente gli standard, sarete frustrati dalle imperfezioni del layout.

Partite quindi dalla perfezione, e poi correggete con qualche *hack* per i browser più *scarsi*. Avrete un codice standardizzato sin dall'inizio, e le correzioni non saranno così numerose.

Fare questo oggi significa partire da Mozilla, Safari, oppure Opera.

### 3. **Assicuratevi che l'effetto desiderato esista davvero.**

Vi sono delle estensioni CSS specifiche per alcuni browser, che non compaiono nelle specifiche ufficiali.

Se state cercando di applicare l'estensione filter o lavorate per formattare le barre di scorrimento, vi affidate a del codice proprietario che funzionerà solo su Internet Explorer. Se il validatore vi dice che il codice non è definito, molto probabilmente è codice proprietario e non avrà un comportamento consistente nei diversi browser.

### 4. **Quando vi affidate ai float per l'impaginazione, assicuratevi che anche l'attributo clear funzioni correttamente.**

L'uso di float è complicato, e non sempre segue la via che vorreste voi. Se avete un float il cui comportamento si estende oltre il bordo dell'elemento che lo contiene, o non si comporta semplicemente come vi aspettate, controllate la correttezza del codice. Sull'argomento potete consultare il tutorial di Eric Meyer.

**5. I margini collassano: utilizzate padding o border per evitarlo.**

Potreste ritrovarvi dello spazio bianco dove non dovrebbe essercene, oppure non trovarlo dove servirebbe. Se usate l'attributo margin, il problema sarà verosimilmente il *collassare* dei margini stessi. [Andy Budd](#) vi spiega cosa vi potete aspettare.

**6. Evitate di assegnare ad un elemento una lunghezza fissa e gli attributi padding/margin.**

Internet Explorer 5 interpreta male il box model, cosa che porta davvero molti problemi. Ci sono [modi per superarli](#), ma è sempre meglio evitare il problema applicando il padding all'elemento genitore anziché al figlio che assume la lunghezza fissa.

**7. Evitate la comparsa di contenuto privo di stile in Internet Explorer.**

Se utilizzate soltanto la tecnica @import per *agganciare* il contenuto al foglio di stile, prima o poi incorrerete nell'inconveniente tipico di IE: dell'HTML non formattato che compare per alcuni istanti prima di *ricevere* il contenuto del foglio di stile. Tutto [questo può essere evitato](#).

**8. Non vi affidate a min-width su IE.**

IE non lo supporta: interpreta però, fino a un certo punto, width come se fosse min-width; giocandoci un pò potete ottenere quindi lo stesso risultato.

**9. Quando avete dei dubbi, diminuite le lunghezze.**

A volte dei problemi di approssimazione causano errori come  $50\% + 50\% = 100.1\%$ . Questo scompagina i layout in alcuni browser; provate a diminuire i 50% fino a 49%, o anche 49.9%.

**10. Il contenuto non si visualizza correttamente in IE?**

Potreste essere affetti dal *Peekaboo bug*, specialmente se i problemi insorgono quando passate con il mouse sopra un link. Controllate *Position is everything* per risolvere il problema.

**11. Fate attenzione ad applicare gli stili ai link se usate le àncore.**

Se usate un'àncora classica nel codice (<a name="anchor">) vi accorgete che assume le pseudoclassi :hover e :active. Per evitarlo, avrete bisogno di

usare anche id per le àncore, oppure applicare gli stili con una sintassi un pò più arcana: :link:hover, :link:active

**12. Ricordate la regola dell'*Amore/Odio* (LoVe/HAtE in inglese) per i link.**

Quando specificate le pseudoclassi per i link, fatelo sempre in quest'ordine: Link, Visited, Hover, Active. Altri ordinamenti non funzioneranno in modo consistente su tutti i browser. Provate anche ad utilizzare :focus, e modificate l'ordine in LVHFA (come suggerito da Matt Haughey).

**13. Ricordate i bordi *problematici* (TRouBLEd in inglese).**

Le forme abbreviate per specificare gli attributi border, margin e padding hanno un ordine preciso: in senso orario a partire dall'alto oppure Top, Right, Bottom, Left.

Quindi margin: 0 1px 3px 5px; vorrà dire nessun margine alto, 1 pixel di margine destro, e così via.

**14. Specificate delle unità per i valori non a zero.**

I CSS richiedono di specificare le unità di misura per tutti le quantità relative a font, margini, dimensioni (l'unica eccezione è line-height, che non richiede un'unità di misura). Non c'è da fidarsi del comportamento di nessun browser quando le unità di misura non sono specificate. Zero è comunque zero, quale che sia l'unità, per questo potete evitare di inserirla. Esempio: padding: 0 2px 0 1em; .

**15. Provate diverse dimensioni dei font.**

Browser moderni come Mozilla e Opera vi permettono di ridimensionare il testo senza curarsi dell'unità di misura che avete usato. Alcuni utenti avranno un carattere di default più grande o più piccolo del vostro; verificate il comportamento del layout con la gamma più ampia possibile di dimensioni del carattere.

**16. Testate con il CSS in linea; pubblicate con il CSS importato.**

Lavorando con il foglio di stile *dentro* il sorgente HTML eliminate ogni possibile errore di cache. Questo è molto importante con alcune versioni di browser per Mac. Prima di pubblicare il sito però, spostate il foglio di stile in un file esterno collegato tramite @import o il tag link.

**17. Aggiungete i bordi che non si vedranno per facilitare il debug del layout.**

Una regola universale come `div {border: solid 1px #f00; }` può esservi utilissima per identificare i problemi di layout. Allo stesso modo, aggiungere temporaneamente dei bordi ad elementi specifici può aiutare a scovare i problemi di sovrapposizione e di spazio bianco non richiesto che altrimenti sarebbero difficili da intravedere.

**18. Non usate virgolette singole (') nei percorsi delle immagini.**

Quando specificate un'immagine di sfondo, non usate le virgolette singole. Non sono necessarie, e IE non le sopporta.

**19. Non collegate a fogli di stile vuoti usati per occupare il posto di futuri CSS (ad esempio fogli di stile per palmari o per la stampa).**

IE5 per Mac stenta sul foglio di stile vuoto e aumenta così il tempo di caricamento della pagina. Mettete invece almeno una regola (o anche solo un commento) nel foglio di stile in modo che Explorer per Mac non stenti.

Estratto da:

<http://www.demetriopolimeno.com/>