

Da Psd a Xhtml: come trasformare il layout in XHTML+CSS?



Nella prima parte di questo articolo abbiamo visto [come esportare, utilizzando lo strumento "sezioni" di Photoshop](#), i vari componenti del nostro layout "[YIW Corporate](#)". Oggi ti "sporcherai le mani" con il codice: vedremo **come realizzare il markup per tale layout, e il CSS associato**.

Bando alle chiacchiere: cominciamo!

Il markup

Una volta terminata l'esportazione di tutti gli elementi dal file sorgente (nel nostro caso un file PSD realizzato con Photoshop) è tempo di pensare al **markup che sarà la spina dorsale del nostro layout**. Come spesso ricordato negli articoli di YIW, il contenuto di una pagina deve essere (almeno inizialmente) **tenuto separato dall'aspetto visuale**. Solo una volta definito, si potranno apportare aggiunte e modifiche per adattare il lavoro alla grafica.

Iniziamo, come al solito, con un documento vuoto, a cui colleghiamo un file CSS:

Partiamo dall'header del nostro layout: il logo sarà l'intestazione di massimo livello, mentre la navigazione la renderemo una lista non ordinata:

Passiamo alla parte centrale: un semplice paragrafo servirà allo scopo di pubblicizzare la nostra azienda:

Tocca ora al contenuto principale della nostra home page: la presentazione dei prodotti. Ci sono diversi approcci possibili: io ritengo che sia significativo utilizzare una lista (perché stiamo presentando appunto una lista di prodotti):

Una cosa interessante da notare è che gli elementi nella lista (titolo-immagine-descrizione) non rispettano l'ordine visuale (immagine-titolo-descrizione), giusto per ribadire che si ha la flessibilità, con i CSS (e un po' di pazienza), **di modificare l'ordine predefinito del browser a proprio piacimento**.

Il footer sarà rappresentato da due semplici paragrafi:

Aggiungendo dei div per rafforzare le divisioni tra le varie sezioni, [ecco il nostro markup di partenza](#), senza alcuno stile applicato:

Stili: diamo una forma ai contenuti

Ora non resta che applicare i CSS a questo markup, per **realizzare il nostro layout**. Per questo progetto utilizzeremo [960 grid system](#), dato che Sara ha progettato tutto il layout seguendo la griglia messa a disposizione da questo framework. Come già spiegato da Nando [nel suo articolo su 960gs](#), dobbiamo scaricare il framework dal sito ufficiale, e collegare i vari file CSS al nostro progetto, in modo da avere una solida base da cui partire.

Innanzitutto, applichiamo alcuni stili di default per il layout, come il font ed il colore dei link:

Ora sistemiamo la struttura del layout, seguendo la griglia: utilizzeremo quella formata da 12 colonne, quindi dovremo aggiungere la classe "*container_12*" agli elementi che costituiscono lo scheletro del nostro layout. Sicuramente il logo e la lista di navigazione dovranno estendersi per tutta la larghezza; anche alla lista dei prodotti dovrà essere aggiunta la classe. Per quanto riguarda il div "*featured*" e "*footer*", purtroppo dovremo aggiungere un div, poiché gli sfondi ripetuti si estendono ben oltre i 960 pixel della larghezza standard della griglia. Dunque il nostro markup si trasforma così:

Il prossimo passo è quello di aggiungere le classi agli elementi della lista contenenti i prodotti della società: dato che sono tre elementi affiancati, e abbiamo scelto una griglia di dodici colonne, ogni elemento dovrà essere largo quattro colonne. Questo effetto è ottenuto aggiungendo la classe "*grid_4*" agli elementi:

Come puoi notare, ho aggiunto anche le classi "*alpha*" e "*omega*" per caratterizzare il primo e l'ultimo elemento, in modo da azzerare rispettivamente il margine sinistro ed il margine destro. Abbiamo ora terminato le modifiche al markup: il nostro layout comincia a prendere forma, ma ci sono ancora molte cose da sistemare. Iniziamo dalla testata.

Header

Cominciamo col nascondere il testo del titolo, sostituendo il logo dell'azienda:

Ottenendo [questo risultato](#). Come puoi notare il logo, che dovrebbe essere un link alla home del sito, non viene visualizzato come tale, poiché abbiamo nascosto il testo, che rappresentava l'estensione del link. Per questo, dobbiamo definire manualmente una larghezza e un'altezza:

Rendendo in questo modo cliccabile tutta la superficie del logo.

Passiamo ora a sistemare i link del menu: la prima cosa da fare è rendere la lista orizzontale:

Come puoi vedere il paragrafo inferiore viene ad affiancarsi ai link. Per eliminare questo sgradevole effetto aggiungiamo la classe "*clearfix*" alla lista dei link nel markup, in questo modo:

E fissiamo un'altezza per tale elemento: ho deciso di impostarla a 42 pixel poiché tale è l'altezza del gradiente che andremo a inserire in seguito:

Non resta ora che applicare gli stili per i singoli link:

Abbiamo impostato colori e grandezza del font, e i margini. Le regole più importanti di questo gruppo sono quelle che fissano la larghezza e l'altezza (ricavate dalle dimensioni dell'immagine di sfondo per i link selezionati) e la dimensione della 'line-height': questa viene impostata allo stesso valore dell'altezza, facendo sì che il testo sia centrato anche verticalmente.

Non resta adesso che inserire lo sfondo e l'header può dirsi completato:

Questo è il [risultato intermedio](#).

Sezione "featured"

Il prossimo passo è quello di realizzare la sezione intermedia, contenente il paragrafo di presentazione e la foto del team. Questa sezione può essere realizzata in più modi, alcuni più semplici altri più sperimentali. Discutendone in redazione, ad esempio, è stata proposta la realizzazione di un'unica immagine di sfondo molto lunga e poi posizionare il paragrafo opportunamente. Per questo esempio, però, ho pensato di presentare una realizzazione più "sperimentale", utilizzando più immagini ed il posizionamento relativo.

Intanto, le immagini di cui avremo bisogno sono due: una che sarà ripetuta orizzontalmente, e quella con il team che verrà posizionata a destra del paragrafo introduttivo. Iniziamo col posizionare la prima immagine, come sfondo del div "`#featured`":

Passiamo poi a modificare l'aspetto e la formattazione del paragrafo, cambiando il colore predefinito, la dimensione e la larghezza:

Passiamo ora alla parte "sperimentale" di questo layout, ovvero il posizionamento dell'immagine del team. La cosa più semplice è porla come background al div "`#slogan`":

Questo è [il risultato ottenuto](#): come puoi vedere, ci sono problemi nell'allineamento dell'immagine e del paragrafo, che vi si sovrappone. Per ovviare a questo inconveniente, occorrerebbe "spingere" a destra il div "`#slogan`": questo effetto si può ottenere utilizzando il **posizionamento relativo**. Quindi il blocco precedente diventerà:

Abbiamo dunque spostato tutto il div "`#slogan`" a destra di 200 pixel: così facendo, però, anche il paragrafo è stato spostato. Dobbiamo rimettere "a posto" questo elemento. Il ragionamento alla base è lo stesso, solamente che ora dobbiamo riportare **200 pixel a sinistra** il div "`#slogan-inner`":

Questa volta abbiamo utilizzato un valore negativo di 200 pixel, per ripristinare il corretto posizionamento del paragrafo, aggiungendo allo stesso tempo delle regole per i margini e per il padding.

Contenuto principale e il footer

Non resta che occuparci delle descrizioni dei tre prodotti: dato che abbiamo utilizzato le classi di 960gs per realizzare le colonne, non dobbiamo che riposizionare le immagini. Ti ricordo che nel layout originale si trovano prima del titolo, mentre nel markup noi le abbiamo posizionate **dopo** per rispettare l'ordine semantico. La soluzione è molto semplice: impostiamo la proprietà "position:absolute" per le immagini, e spostandole in alto: in questo modo vengono estratte dal

flusso della pagina e vanno a coprire il testo, che quindi dovrà essere spostato in basso tramite padding. In codice:

A queste proprietà, vanno aggiunte quelle per modificare l'aspetto visivo, diventando quindi:

L'ultimo passo da compiere è quello di aggiungere lo sfondo al footer, e centrare il testo:

A questo punto possiamo dire [di aver ultimato il nostro layout](#).

Conclusioni

In questi due articoli ho voluto mostrare un po' il modo di ragionare che ci può essere alla base della costruzione di un layout partendo da un file psd. Chiaramente, come ho già ripetuto molte volte, non esiste **IL metodo migliore**. Un punto importante, secondo me, è il fatto che dallo stesso layout grafico, sviluppatori diversi possono ottenere un buon risultato seguendo strade diverse, applicando metodologie più o meno all'avanguardia, o strani hack dal codice oscuro.

Credo che sia il bello di questo lavoro: le possibilità di miglioramento sono pressoché illimitate, e sta agli sviluppatori rendere diverso ogni *coding* piuttosto che adagiarsi sulle tecniche già imparate e consolidate. Innovare è la parola d'ordine secondo me. E noi, che viviamo in questo media cangiante che è il Web, non possiamo che essere innovatori in prima linea.