

Pillole CSS: come realizzare un layout fisso a due colonne?

Con l'introduzione dei [framework CSS come 960gs](#), o [Blueprint](#), realizzare layout con CSS è diventato davvero semplice. L'unica pecca che affligge questi framework è anche il loro punto di forza: nascondere le implementazioni all'utente, permettendo di realizzare layout complessi utilizzando la giusta combinazione di classi CSS.

Per un professionista però è importante capire quali sono i concetti fondamentali per la costruzione dei layout. Oggi vedremo uno dei metodi per **realizzare un layout fisso**.

Cos'è un layout fisso?

Un layout si dice fisso quando tutti gli elementi che lo compongono hanno una larghezza fissata a priori. Di solito l'unità di misura utilizzata per definire un layout fisso è il *pixel* (px). I layout fissi si oppongono ai layout "fluidi o liquidi", in cui le larghezze dei vari componenti sono definite utilizzando le percentuali (%), permettendo al layout di "adattarsi" alla finestra che lo contiene.

Più strade portano allo stesso risultato

Ci sono molti modi per realizzare un layout fisso: utilizzando il posizionamento assoluto, ad esempio, oppure i margini negativi. In questo esempio utilizzerò i **float**, per il semplice motivo che è il metodo più semplice e flessibile.

Layout, ovvero organizzare il contenuto, non l'aspetto

Quando si realizza un layout (X)HTML + CSS molto spesso si fa l'errore di pensare prima all'aspetto visivo ("Quante colonne? Di che colore? Che font uso?") piuttosto che al contenuto. La fretta di giungere ad un risultato "visivamente corretto" porta all'introduzione di molto markup aggiuntivo e inutile. Partendo dal contenuto, invece, si aggiungono man mano gli elementi necessari, garantendo la massima semplicità.

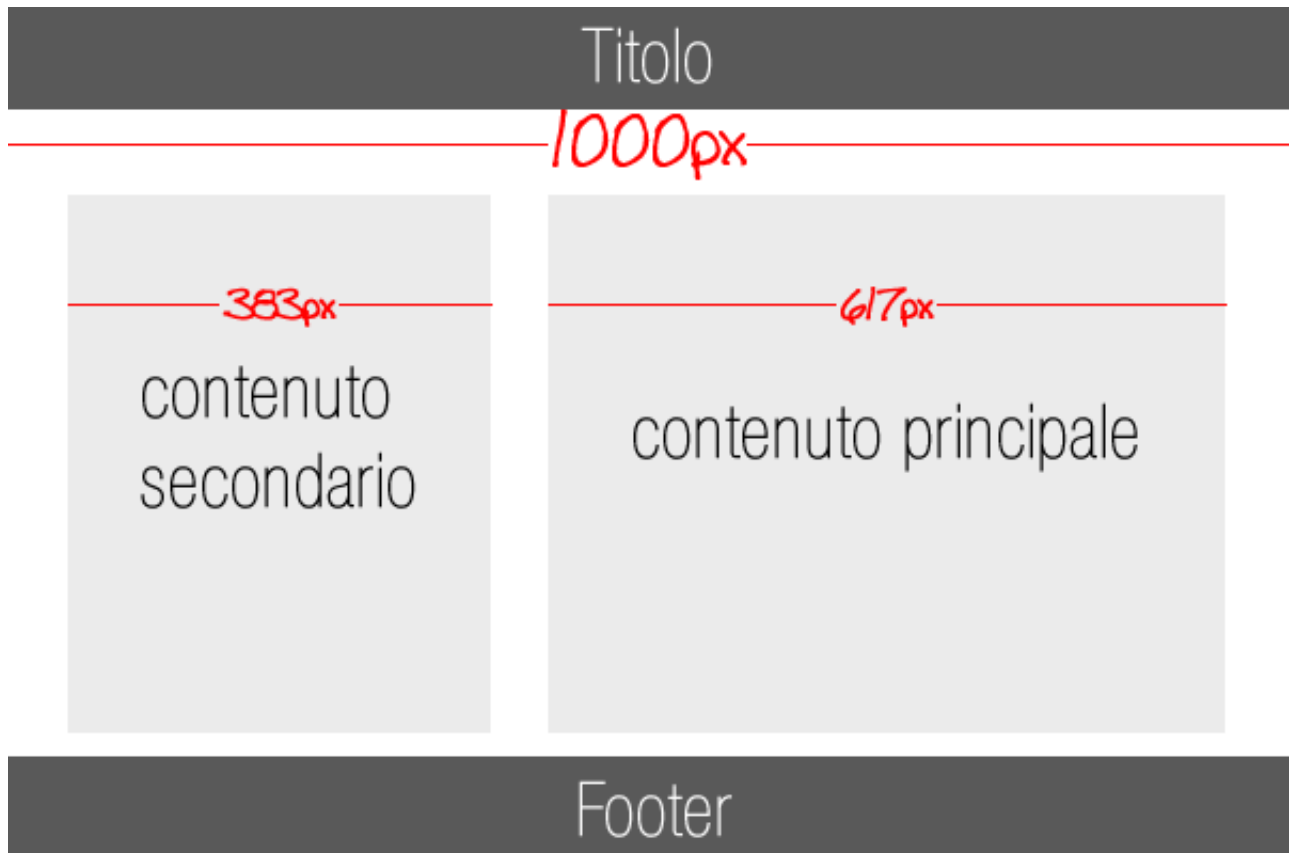
Immaginiamo di voler realizzare una pagina per una ricetta. La prima domanda che devi porti è: **"Quali sono gli elementi di cui ho bisogno?"**. Una ricetta è composta fondamentalmente di due parti: un elenco con gli ingredienti necessari, e le istruzioni di realizzazione. Quindi, in (X)HTML, verrebbero resi così:

Le istruzioni vengono rese tramite una lista ordinata, mentre gli ingredienti utilizzano una lista non ordinata. Aggiungiamo un titolo alla pagina, una nota di copyright e [il DOCTYPE](#):

Come puoi vedere [nell'esempio](#) la pagina ha senso anche senza nessuno stile applicato.

Realizzare il layout

Ora che abbiamo definito i componenti base del nostro layout, possiamo passare all'aspetto visivo. È bene sempre avere un'idea di massima di ciò che si vuole realizzare, quindi **prepariamo uno "schema guida" o wireframe**, come questo:



Quindi vogliamo che la larghezza totale sia di 1000 pixel: le sezioni "titolo" e "footer" si estenderanno per tutta la lunghezza, mentre il contenuto principale e quello secondario avranno una larghezza di 617 pixel e 383 pixel, rispettivamente. (Se sei curioso di sapere perché ho scelto questi numeri, ti consiglio di leggere ["La sezione aurea"](#) su Wikipedia)

Il primo passo verso la costruzione del layout è quello di racchiudere i vari elementi in "div" creando quindi delle sezioni semanticamente significative:

Abbiamo quindi aggiunto i "div" con ID "testata, ricetta, ingredienti e note_legali". **Questo è markup semantico:** infatti tali nomi riescono a dare il senso del possibile contenuto del tag molto

meglio di "sopra, destra, sinistra, sotto", riuscendo quindi a separare completamente l'aspetto visivo dal codice HTML.

Una volta completato il markup della nostra pagina, è tempo di lavorare con i CSS. Apri dunque un nuovo file nel tuo editor preferito e salvalo come "stile.css". Come specificato nel wireframe, dobbiamo fissare una larghezza di 1000 pixel per il nostro layout e [centrarlo](#) :

Aggiungiamo poi un piccolo bordo al div "testata" e "note_legali", per separare visivamente le due sezioni:

[Ecco il nostro risultato intermedio](#). Ora non ci resta che realizzare le sezioni "ricetta" e "ingredienti" e, come detto precedentemente, utilizzeremo i float per affiancare i due elementi. Una volta definite le dimensioni, "flottiamo" a sinistra il div "ingredienti", e a destra il div "ricetta":

Quando si utilizza il float, gli elementi vengono esclusi dal normale flusso della pagina: per ripristinarlo, bisogna applicare la proprietà "clear" ad un elemento che segue gli elementi flottati. Nel nostro esempio, il div "note_legali" è un perfetto candidato:

Con queste semplici regole CSS abbiamo realizzato un layout fisso a due colonne, con codice valido e semantico: qui puoi vedere il [risultato finale](#).

E se volessi invertire le due colonne?

Dato che non abbiamo utilizzato markup presentazionale, per scambiare le due colonne è sufficiente invertire le direzioni del float dei due elementi "ricetta" e "ingredienti", in questo modo:

Se avessimo utilizzato ID come "destra" e "sinistra", invece avremmo dovuto modificare anche il codice (X)HTML per riflettere i cambiamenti all'aspetto presentazionale della nostra pagina.

Conclusioni

In questo articolo abbiamo visto come realizzare un layout fisso a due colonne di base. Nella prossima "pillola" lo estenderemo al caso di tre colonne, e faremo alcune considerazioni sull'uso dei margini e del padding laterali. Alla prossima!