

Framework CSS: Grid System 960. Cos'è?

Da qualche anno a questa parte si è molto parlato di **Framework CSS**: Blueprint, Grid System 960, Elastic, ecc... in molti dicono che possono essere un buon punto di partenza per lo sviluppo dei propri lavori.

Ma è davvero così? Oggi vedremo nel dettaglio l'utilizzo di uno di questi **framework CSS: il Grid System 960**.

Cos'è un framework CSS?

Un **framework css** è uno strumento che fornisce allo sviluppatore un'architettura ben definita, basata su regole ben precise già scritte che consentono di **realizzare applicazioni web senza dover partire da zero**.

E cos'è il framework CSS Grid System 960?

Il **framework CSS Grid System 960** non è altro che un insieme di fogli di stile con **regole già pronte** che considerano il contenitore generale disposto su una griglia. Il framework mette a disposizione due griglie: una formata da 12 colonne e un'altra da 16. In entrambe le griglie ovviamente la larghezza del contenitore generale resta sempre di 960px.

Perché scegliere una dimensione di 960px?

Tutti i monitor moderni supportano ormai una risoluzione minima di 1024x768 pixel. **Il numero 960 è divisibile** per 2, 3, 4, 5, 6, 8, 10, 12, 15, 16, 20, 24, 30, 32, 40, 48, 60, 64, 80, 96, 120, 160, 192, 240, 320 e 480. Questo rende molto più flessibili le combinazioni relative alle **dimensioni delle colonne** che si possono realizzare.

Nell'immagine che segue puoi vedere un esempio di due siti realizzati utilizzando le due griglie messe a disposizione **dal framework Grid System 960**: il primo utilizza la **griglia a 12 colonne** mentre il secondo fa uso della **griglia a 16 colonne**.

Come si utilizza il framework CSS Grid System 960?

Dopo aver scaricato il [framework](#), la prima cosa da fare è caricare i file necessari per lavorare con la nostra griglia:

Un piccolo consiglio che voglio darti: non apportare modifiche direttamente al file 960.css perché altrimenti queste verrebbero perse nel momento in cui, in futuro, si eseguirebbe un update del framework.

Quindi creiamo un foglio di stile separato da quelli del framework che chiameremo "style.css" il quale conterrà le regole che desideriamo assegnare al sito che stiamo sviluppando e carichiamo anch'esso:

Come scelgo il contenitore/griglia da utilizzare?

Adesso dobbiamo scegliere il tipo di griglia che vogliamo utilizzare tra le due messe a disposizione dal framework, **griglia a 12 colonne** (.container_12) o **griglia a 16 colonne** (.container_16). Abbiamo già accennato al fatto che le due griglie hanno entrambe una **dimensione di 960px**, l'unica differenza è che la griglia a 12 colonne ne divide la dimensione per 12 mentre quella a 16 la suddivide per 16.

La classe che permette di scegliere il tipo di griglia è la seguente:

oppure:

Come stabiliamo la dimensione delle colonne?

Quando desideriamo dividere il nostro contenitore in più colonne basterà semplicemente utilizzare la classe `.grid_XX`, dove "XX" rappresenta il numero di colonne (della griglia scelta come contenitore) di cui sarà composta la colonna che vogliamo realizzare.

Per esempio, se desideriamo realizzare due colonne (menù laterale e contenuti) nel layout del sito che stiamo sviluppando basterà semplicemente indicarlo così:

La classe "*grid_7*" indica che la colonna menù laterale dovrà avere una dimensione pari a 7 colonne della griglia scelta.

Come puoi notare, la somma della prima colonna (*grid_7*) con la seconda colonna (*grid_5*) è esattamente 12, il totale delle colonne che compongono la griglia del contenitore che abbiamo scelto (*container_12*). In questo modo non è necessario conoscere la larghezza che dovrà avere ogni singola colonna è diventa molto più semplice **dimensionare le nostre colonne** durante lo sviluppo.

Guardiamo un altro esempio, questa volta ci è stato richiesto di suddividere il nostro layout in tre colonne di uguale dimensione, senza dover fare complessi calcoli per specificare le dimensioni in pixel che ciascuna colonna dovrà avere, basterà applicare un po' di matematica per ottenere quanto segue:

Anche qui come puoi notare la somma delle tre colonne *grid_4* è esattamente 12 ($4 + 4 + 4$).

Esistono margini tra una colonna e l'altra?

Per impostazione predefinita le colonne hanno un certo margine tra loro. **Ogni classe *grid_XX* ha 10px di margine** sia a destra che sinistra. Questo comporta che due colonne adiacenti hanno tra di loro un margine totale di 20px rendendo più armoniosa e leggibile la presentazione dei contenuti.

Nel caso in cui desideriamo che la prima colonna e/o l'ultima colonna del nostro layout non abbiano nessun margine basta aggiungere la classe "alpha" per non assegnare il margine sinistro alla colonna desiderata o la classe "omega" per evitare di assegnare il margine destro.

Vediamone un esempio:

Nell'esempio riportato sopra la prima colonna non avrà margine sinistro e l'ultima colonna non avrà nessun margine destro.

E se si desidera lasciare uno o più spazi vuoti prima o dopo una colonna?

Se si desidera **inserire uno o più spazi vuoti prima o dopo una colonna** basta avvalersi delle

classi "*prefix_XX*" o "*suffix_XX*" che servono proprio a specificare il numero di colonne (*_XX*) che si vogliono lasciare vuote. Vediamone un esempio:

Nell'esempio riportato sopra abbiamo lasciato uno spazio vuoto ("*prefix_1*": pari alla dimensione di una colonna della griglia scelta) prima di definire una colonna di dimensione *grid_7*, seguita da una colonna con dimensione *grid_3* e uno spazio vuoto subito dopo ("*suffix_1*").

Da notare che la somma delle due colonne in aggiunta agli spazi vuoti che abbiamo lasciato all'inizio della prima colonna e subito dopo la seconda (*grid_7 + grid_3 + prefix_1 + suffix_1*) risulta essere sempre 12.

Conclusioni

L'utilizzo di un **framework CSS**, come hai avuto modo di vedere, può portare notevoli risparmi di tempo durante lo sviluppo dei tuoi progetti. Ovviamente nulla vieta di personalizzare il framework secondo le tue effettive esigenze in quanto queste soluzioni non potranno mai essere universali, soprattutto per quel che riguarda la tipografia dei tuoi lavori. Quindi il consiglio è di non fermarti al semplice utilizzo di un framework ma di spingerti oltre, personalizzandolo fino a crearne uno tutto tuo da adoperare come punto di partenza per i tuoi futuri lavori.