Impaginiamo i contenuti in maniera intelligente con Bootstrap 3

Per chi non lo conoscesse, **Twitter Bootstrap** è uno dei più popolari e potenti **framework front-end** in circolazione; mette a disposizione classi **CSS** predefinite che permettono di semplificare enormemente lo sviluppo delle nostre pagine web.

Probabilmente, avrete già letto altri articoli che ne parlano su questo blog.

Ma, come tutti i prodotti validi, gli aggiornamenti corrono velocemente e con la **nuova versione Bootstrap** ha deciso di apportare alcuni cambiamenti che necessitano di un nuovo studio.

In questo articolo parleremo di **come utilizzare** l'ottimo sistema di **griglie** offertoci da Bootstrap. Se volete iniziare a impaginare i contenuti in maniera intelligente, sfruttando tutti i vantaggi che può offrire un framework così potente, allora questo è l'articolo che stavate cercando.

I concetti principali

Bootstrap offre un sistema di griglie **mobile first,** suddiviso su 12 colonne che cambiano di dimensione in basa alla grandezza del dispositivo sulle quali vengono visualizzate.

Un sistema di griglie permette di realizzare il **layout** di una pagina suddividendola in **righe** a loro volta suddivise in **colonne (12** nel caso di bootstrap).

Le cose principali da sapere prima di poter utilizzare il *grid system* di **Bootstrap** sono:

- Le righe (di classe *.rows)* vengono utilizzate per creare **gruppi orizzontali** di **colonne** e solo le colonne dovrebbero essere figlie dirette delle righe.
- Le righe devono essere posizionate all'interno di contenitori aventi la classe *.container* (larghezza fissata) oppure *.container-fluid* (larghezza totale della finestra), questo serve per ottenere corretti allineamento e *padding.*
- Le colonne vengono create assegnando i numeri da **1 a 12 assegnati** a **determinati prefissi,** ad esempio tre colonne uguali potrebbero avere la classe *.col-md-4*

Nell'ultimo punto ho utilizzato di proposito la parola "potrebbero", il prefisso cambia infatti in base alla grandezza del dispositivo che stiamo prendendo in considerazione. Vediamo cosa significa.

Mediaquery e breakpoint

Bootstrap utilizza quattro indicatori per riferirsi ai break point delle media query:

- **xs** (extra small)
- **sm** (small)
- **md** (medium)
- **lg** (large)

Il loro utilizzo per la griglia è riassunto in questa tabella

	Dispositivi extra piccoli Cellulari (<768px)	Dispositivi piccoli Tablet (≥768px)	Dispositivi medi Desktop (≥992px)	Dispositivi grandi Desktop (≥1200px)
Larghezza .container	Auto	750px	970px	1170px
Prefisso classe	`.col-xs-`	`.col-sm-`	`.col-md-`	`.col-lg-`
Larghezza colonne	Auto	60px	78px	95px
Larghezza padding	30px (15px per ogni lato della colonna)			

Un po' di esempi pratici

É bene sapere che i prefissi assegnati agli elementi della griglia hanno effetto anche sui **breakpoint maggiori** se questi non sono esplicitamente definiti. Questo significa che se assegnamo a un elemento solo la classe *.col-md-6* questo occuperà sei colonne sia su schermi medi **(md)** che su schermi larghi **(lg** è più grande di md e non è espressamente definito), ma non su quelli piccoli.

Faccio un semplice esempio pratico per spiegarmi meglio.
Prendiamo in esame il codice:

Il risultato è visibile qui.

Come potete vedere, questo codice restituisce delle **righe suddivise verticalmente** soltanto su **schermi desktop** (da md in poi), mentre su schermi più **piccoli** ogni elemento prenderà di **default** la **grandezza** di **massima** di **12 colonne,** facendo sì che tutti gli elementi finiscano uno **sotto** l'altro.

Se invece volessimo conservare le forma delle celle anche su schermi piccoli ci basterebbe utilizzare la classe **xs** al posto di md, in modo che la suddivisione dichiarata venga applicata dal break extra small in poi, cioè su tutti.

Ma ancora meglio **Bootstrap** ci permette di assegnare forme diverse alle colonne per diversi break point, semplicemente inserendo i prefissi dei i break che ci interessano, come possiamo vede qui.

Ho usato questo esempio per mostrarvi due cose importanti (che potete vedere dallo schermo di un **tablet,** di un cellulare o ovviamente ridimensionando la finestra):

- Le righe di Bootstrap accettano **l'overflow,** infatti potete vedere che nella prima riga ci sono dodici elementi da due colonne ognuno, una volta occupate le dodici colonne, gli elementi vanno a capo in maniera automatica in modo da avere sempre dodici colonne per riga.
- Le righe non devono necessariamente essere occupate totalmente.

Offset delle colonne

Come vi dicevo le righe non devono necessariamente essere riempite, ma non solo, possiamo anche **gestire** degli **spazi vuoti,** per distanziare gli elementi a nostro piacimento.

Per farlo viene utilizzata la **dichiarazione offset.** Come potete vedere qui, basta aggiungere l'offset all'elemento che deve essere spostato impostando il numero di colonne che deve saltare.

Gli indicatori xs, sm, md, lg *dell'offset* seguono le stesse regole di quelli delle colonne, anche con gli offset infatti, gli indicatori più piccoli si ripercuotono sui **breakpoint** maggiori se questi non sono esplicitamente definiti.

Colonne annidate

Bootstrap permette inoltre di **annidare righe** all'interno di elementi figli a una prima riga. In questo modo l'elemento contenitore si comporterà come un *.container* da dodici colonne (anche se è più piccola) e al suo interno funzionano tutti gli esempi visti in precedenza.

Vi propongo quest'ultimo esempio visualizzabile qui.

Come potete vedere l'elemento che contiene la seconda riga è lungo solo sei colonne, ma con la riga interna possiamo riferirci sempre ad un totale di dodici.

In conclusione

In questo articolo abbiamo visto come impaginare i contenuti grazie al sistema delle griglie offertoci da **Bootstrap,** che ci permette di spostare e annidare i contenuti in maniera intelligente e rapida.

Sono certo che alcuni (come ho fatto anche io a mio tempo) riterrano troppo ostico imparare le metodologie di Bootstrap, ma vi assicuro che se investiamo su di lui un po' del nostro tempo, saprà ripagarci risolvendo un sacco di problemi futuri in un batter d'occhio!

Vi invito quindi di iniziare a studiarlo a fondo. Per qualsiasi dubbio inerente al grid system, resto a completa disposizione nei commenti.

Mi piacerebbe inoltre leggere nei commenti quali altre **guide** a **bootstrap** vi piacerebbe leggere, così da dare **priorità** a quelli **più richiesti.**