**BootsTrap**

* Este un HTML,CSS si JavaScript Framework pentru a face siteuri responsive
* Bootstrap contine in mare parte foarte multe clase CSS
* Bootstrap cere sa avem o anumita structura in paginile noastre HTML
* Bootstrap e mobile first, deci se bazeaza ca intai facem siteul pe mobile
* Putem sa instala clasele sale si tot ce are de aici:

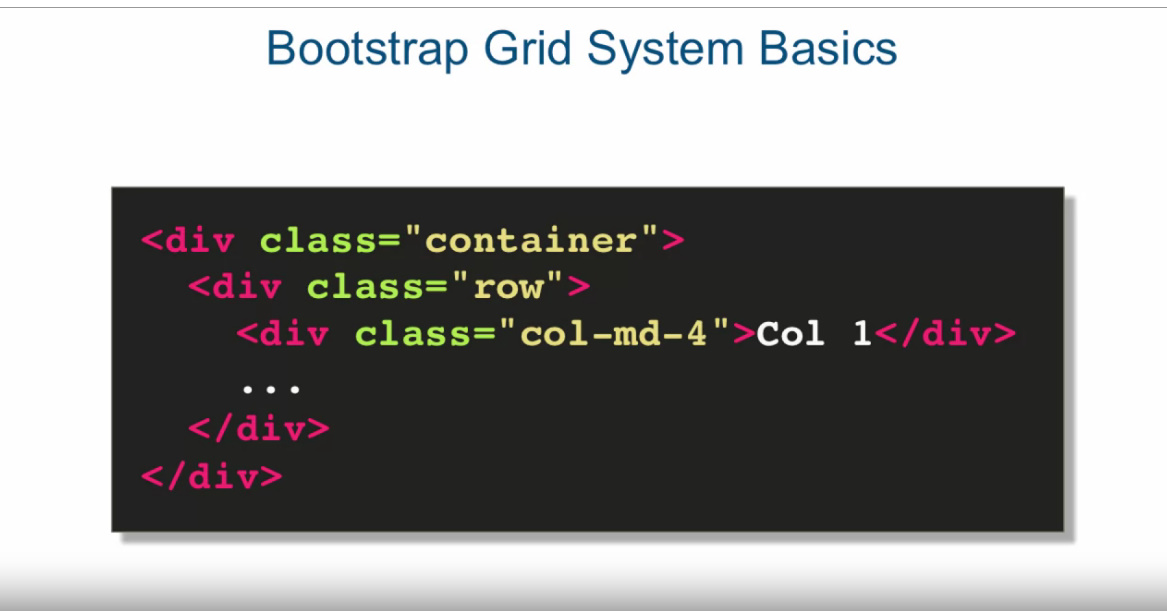
[Download · Bootstrap (getbootstrap.com)](https://getbootstrap.com/docs/4.0/getting-started/download/)

* Sau putem sa il includem direct in proiect, remote ca aici:

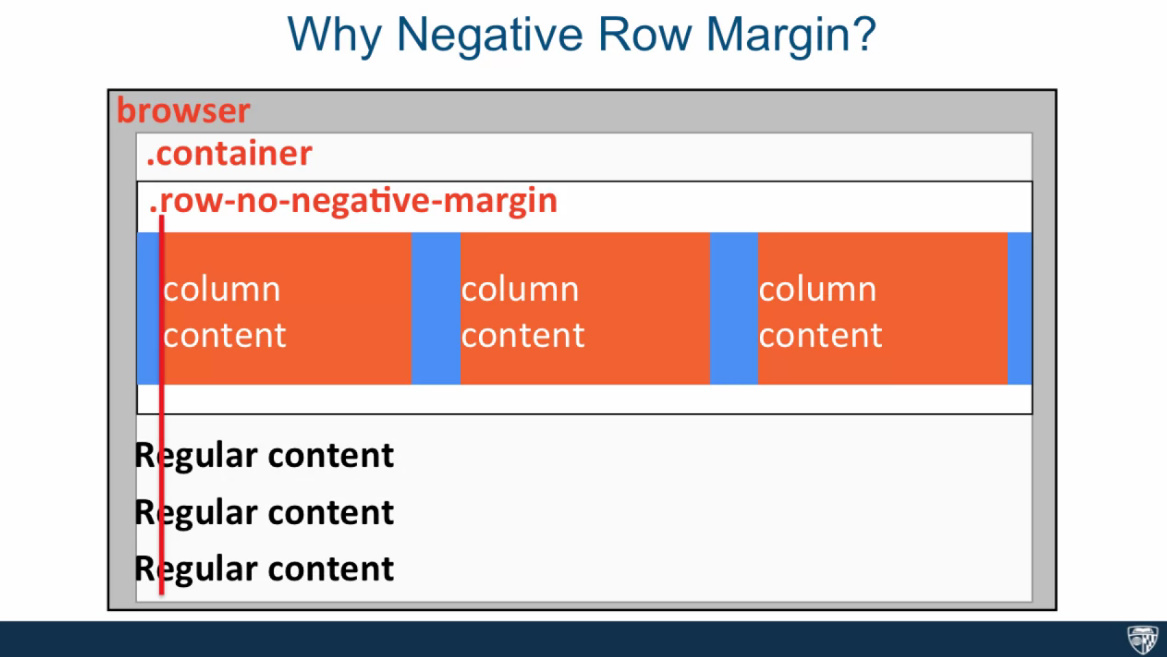
[Get started with Bootstrap · Bootstrap v5.3 (getbootstrap.com)](https://getbootstrap.com/docs/5.3/getting-started/introduction/#quick-start)

* Avem grija sa punem <link> pentru bootstrap inaintea la CSS al nostru, ca CSS al nostru sa poata rescrie chestii care exista in bootstrap
* Bootstrap scrie multe clase si inclusiv si proprietati pentru h1,h2 etc.\

**Gryd system**

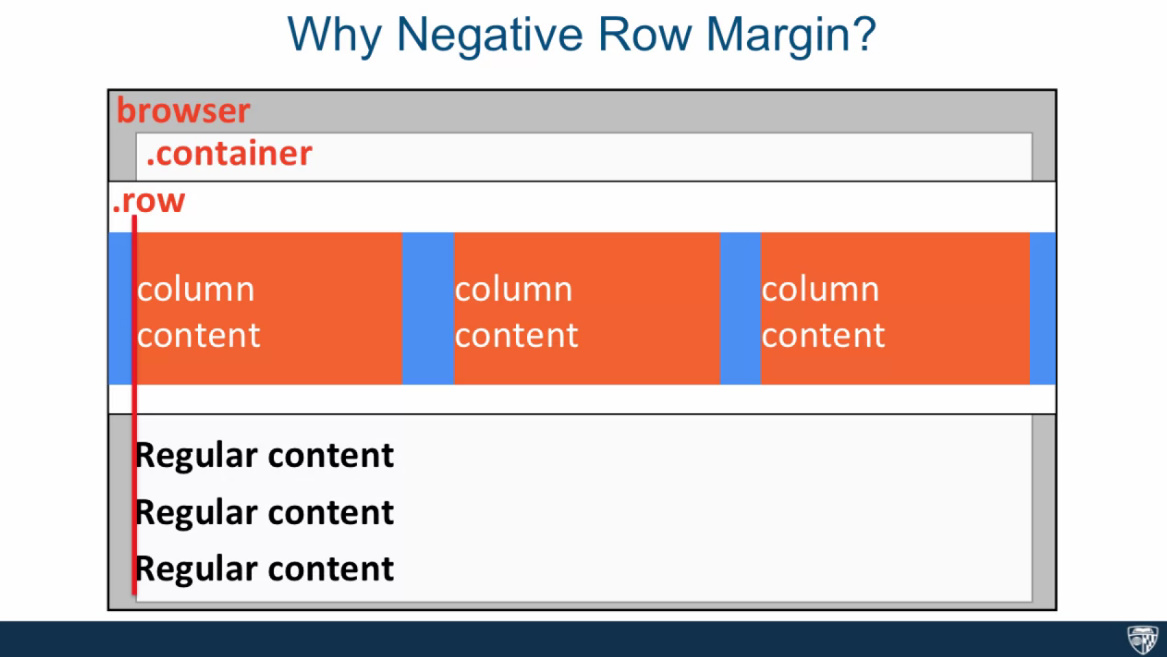


* Acest tip de structura este cel mai bun si folosit
* Orice bootstrap grid trebuie sa fie intr-un **container** sau **container-fluid(care creaza cele 12 coloane pentru noi)**
* **container-fluid** – ofera width max
* **container** – fixed width, dar tot responsive
* Un container e ca un div, doar ca il el se pun elementele pentru bootstrap
* Urmatorul component este **row**. Aceasta clasa creaza grupe orizontale de coloane intr-o linie. Asta inseamna ca coloanele interactioneaza una cu alta ca un grup dar independent de coloanele din alt row.
* Totodata, row class adauga negative left/right margins

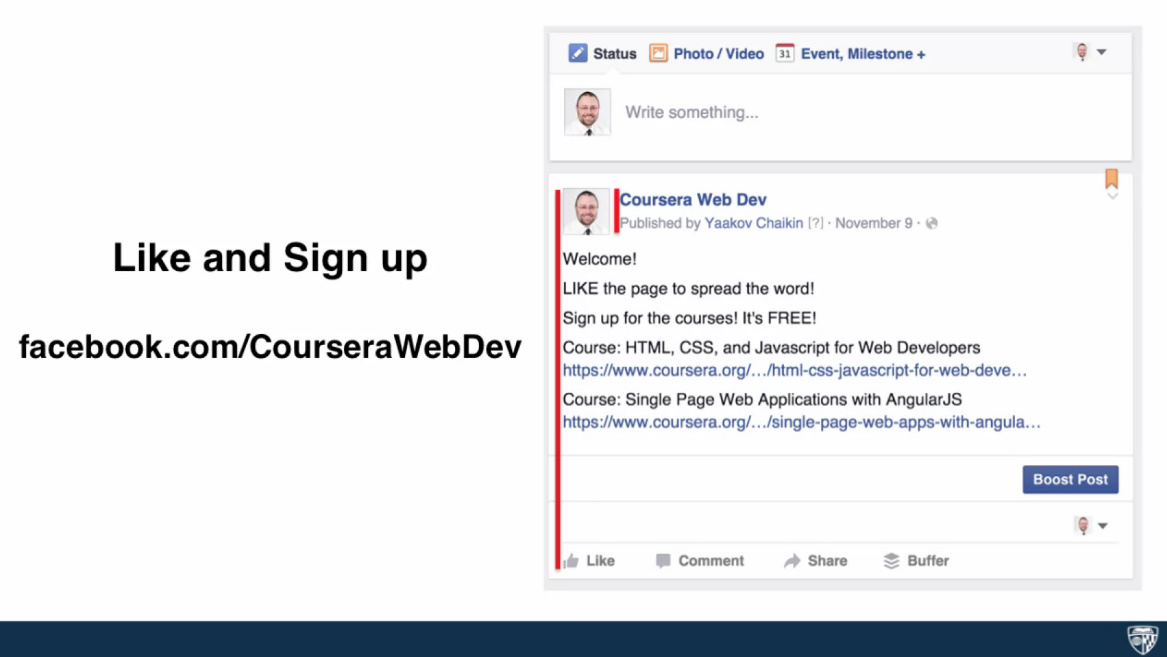


Sa zicem ca luam clasa .row-no-negative-margin

1. Vedem in primul rand ce este container. Este insasi pagina noastra pe care bootstrap o cntroleaza cu 12 coloane create
2. Row este un rand cu content adaugata unde se afla coloanele
3. Fiecare row are columns, dar vedem ca daca nu se adauga negative left/right margins, coloanele nu incep de acolo unde si textul, ci se duc mai inainte



.row va face ca linia sa iasa putin din container, de aia si are margin negativa, dar coloanele asa incep odata cu textul si se termina tot cu el, deci la inceput de laturile verticale



* **Coloana** – se foloseste formatul COL-SIZE-SPAN si spune anume cate coloane va ocupa tagul dat din cele 12

**Fara col randurile se vor pune unul peste altul!**

* **Size** este pentru dimensiunea ecranului(extra-small,small,large,medium)

**large**(LG) – 1200px+, deci coloanele vor lua tot spatiul si vor sta una sub alta

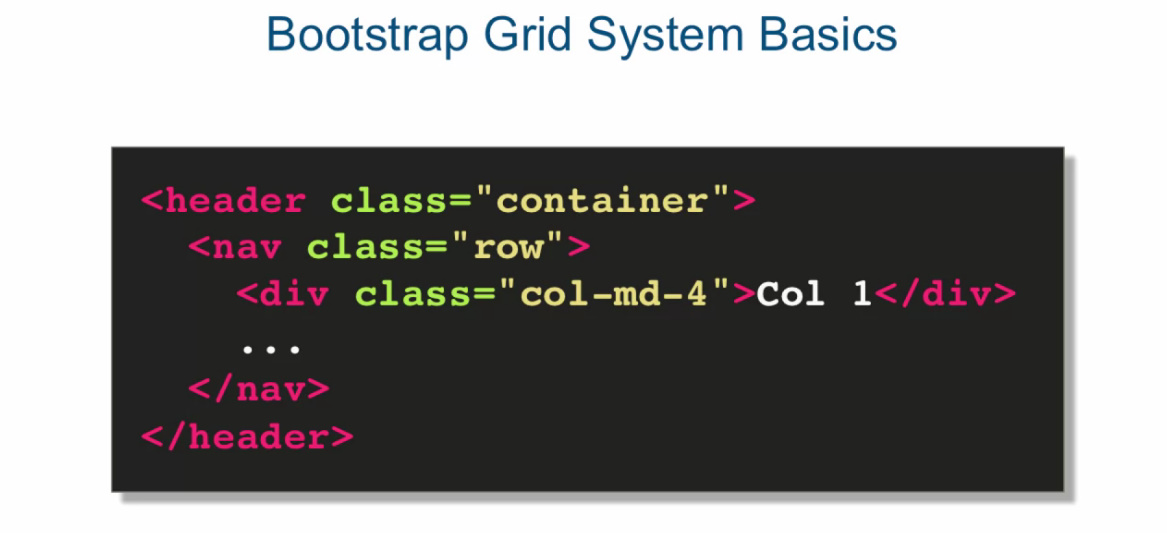
**medium**(MD) –

**small**(sm)

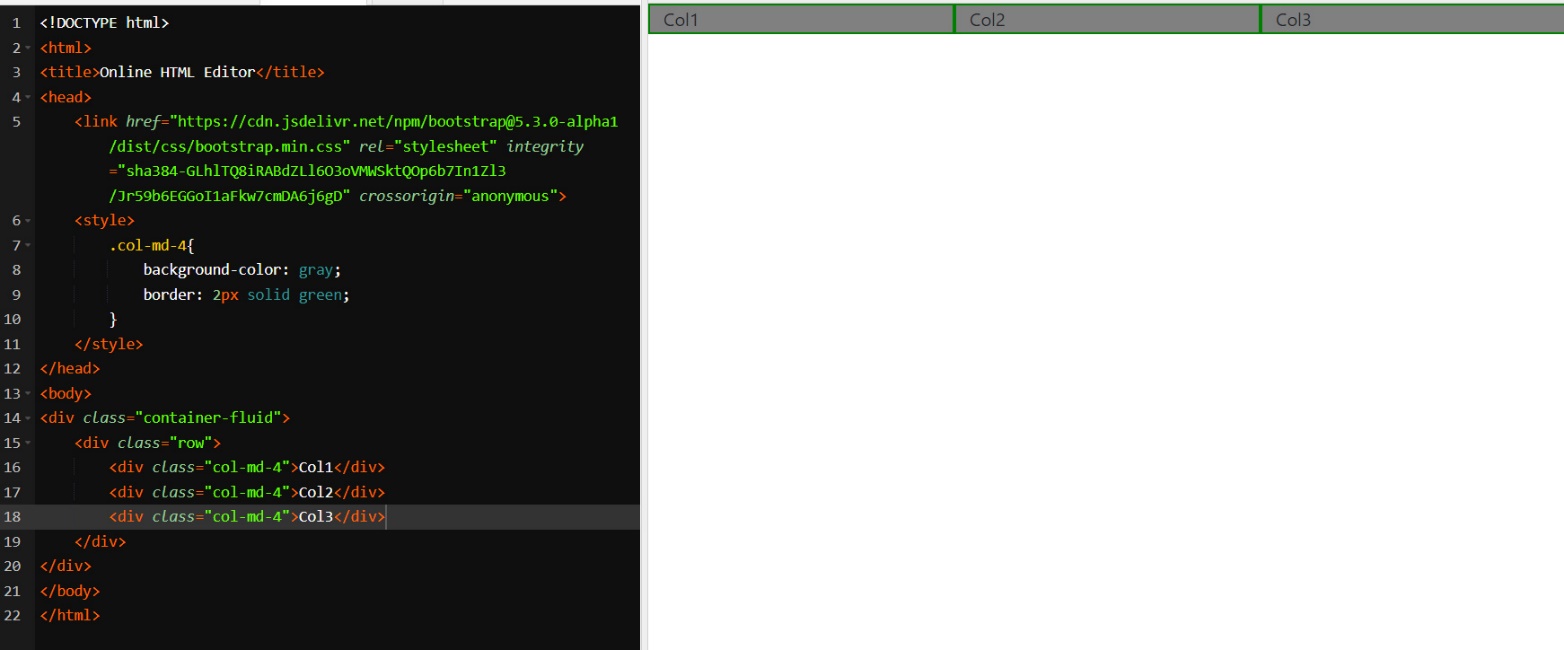
**extra-small**(xs)

* **Span** – cate coloane sa se ocupe din cele 12(1 – 12)

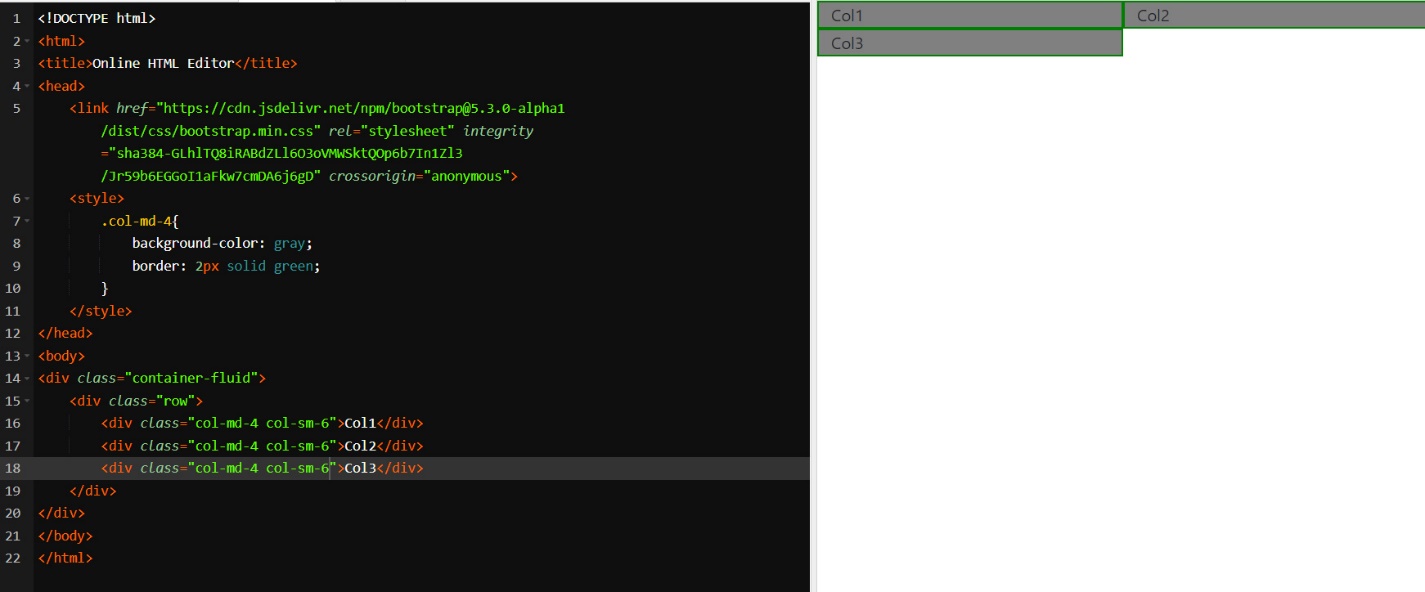
DIV nu e obligatriu, putem folosi orice



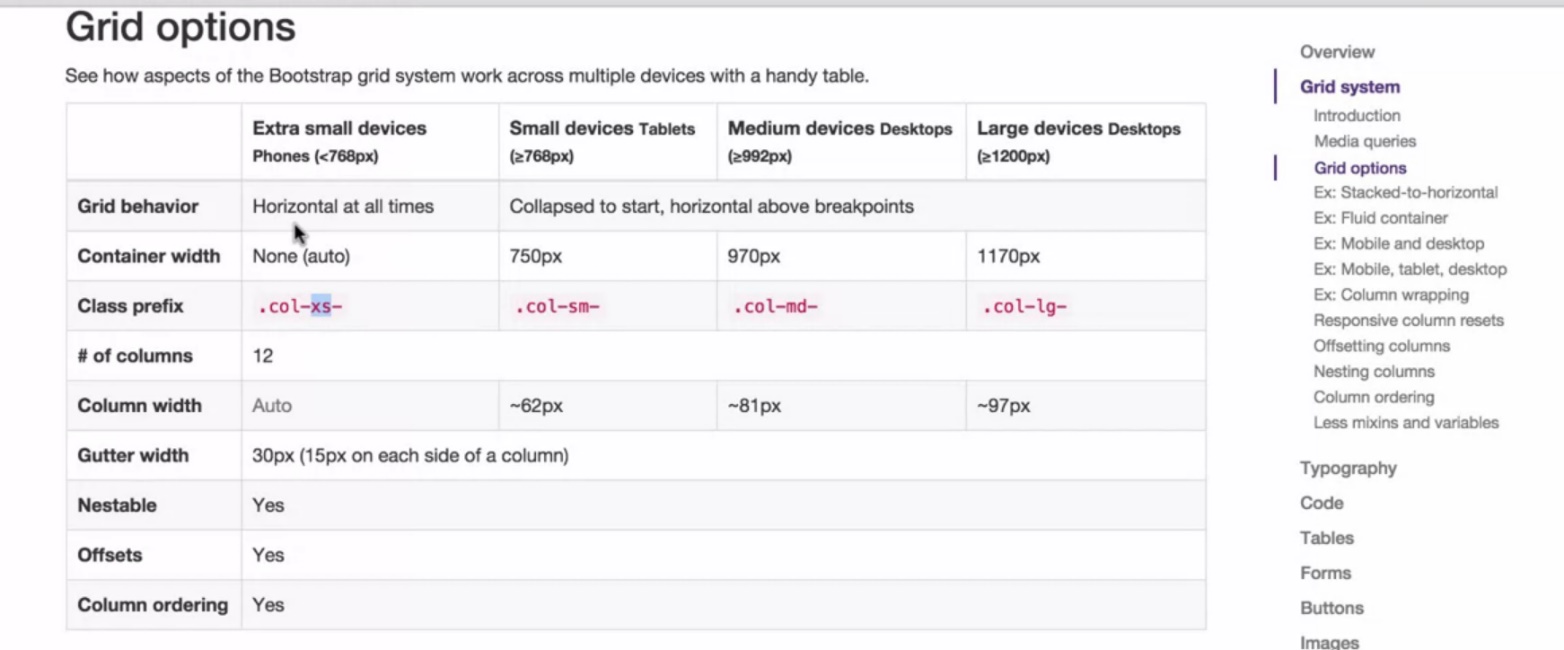
**Exemplu**



Avem acum un container, si in el un rand. Acum, in randul dat avem 3 divuri(ca si cum 3 coloane), si fiecare ocupa 4 coloane din cele 12 ce nu le vedem.dar exista. Deci fiecare coloana ia 1/3 din ecran



Aici deja se foloseste clasa col-sm-6, adica pentru ecran small, si deci fiecare div va fi o coloana ce va lua 6 coloane din cele 12 ce nu se vad, adica va ocupa jumatate de ecran



Collapsed to start ,horizontal above breakpoints – inseamna ca daca ecranul devine mai mic de valorile indicate, coloanele incep sa ocupe tot ecranul si sa stea una peste alta

.col-xs- deci va face ca coloanele mere sa stea una sub alta

