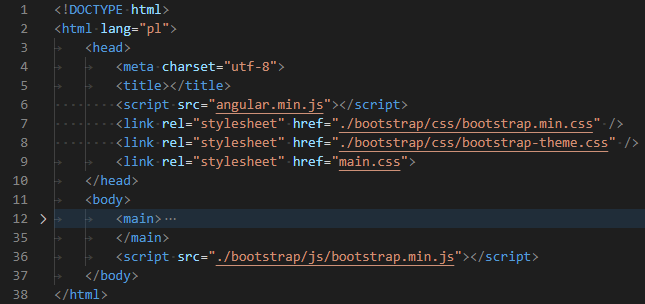
# Framework Bootstrap – GRID (siatka strony vs. Responsywność)

Zasadność używania siatki bootstrapowej:



(Podział skalowalności rozdzielczości ekranu przy różnych typach urządzeń stacjonarnych i mobilnych)

Przygotowujemy szablon strony z pobraną biblioteką bootstrap.

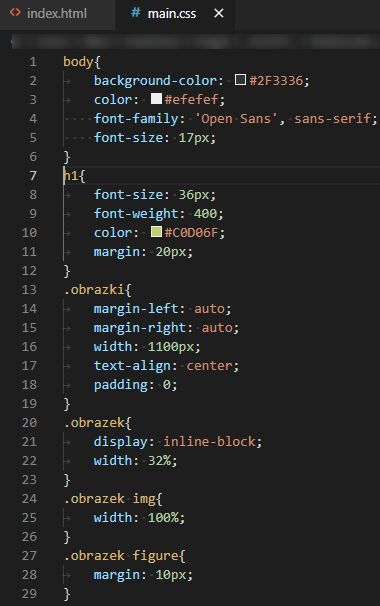


Wpisujemy dodatkowy skrypt wyświetlający nasze pliki graficzne:

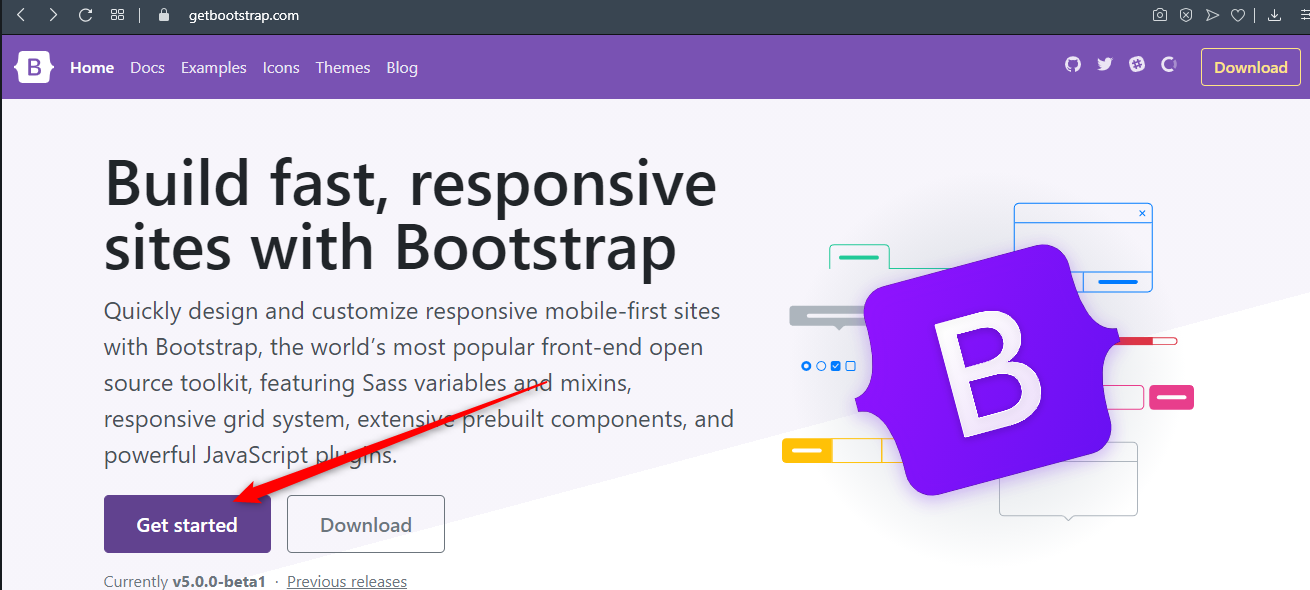


Pliki należy umieścić w odpowiednik katalogi i nadać im odpowiednie nazwy.

Układamy elementy za pomocą CSS:



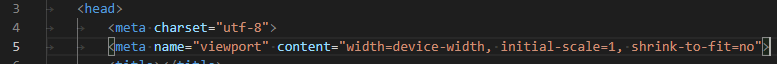
Dodatkowo warto podłączyć dodatkowe dwie biblioteki z (biblioteki te można też pobrać i używać lokalnie):



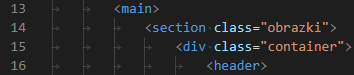
Są to jQuery oraz popper.js : (poniżej wiersze zostały zawinięte)



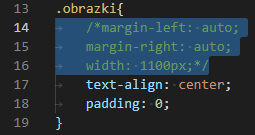
W celu umożliwienia skalowania rozdzielczości należy dołączyć znacznik meta umożliwiający wprowadzenie tej funkcjonalności;:



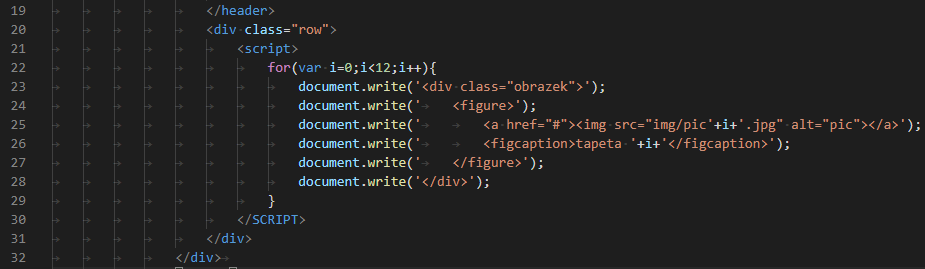
Kolejnym krokiem jest otagowanie siatki kontenerem. Bootstrap ma przygotowany CSS obsługujący tą klasę.

 przenosimy jednocześnie klasę „obrazki” do <section>

Teraz bootstrap będzie pilnować szerokości więc przerabiamy w main.css selektor klasy:

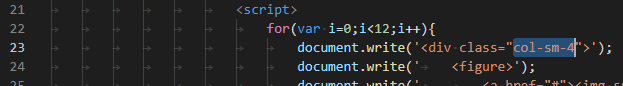


Teraz wszystkie obrazki wrzucamy do wiersza GRIDowego, czyli wstawiamy do dodatkowego DIVa.



W tym momencie strona nie powinna się już utożsamiać z poziomym paskiem przewijania.

W tym momencie w kontenerze mamy wiersz do którego możemy włożyć elementy o klasie autoskalującej w bootstrapie. Np. może to być klasa o nazwie .col-sm-4 (rozmiar mały z trzema elementami ułożonymi na szerokości – czyli każdy element zajmuje cztery kolumny). W main.css możemy pominąć klasą ‘obrazek’.



Znaczniki <img> przestały się zachowywać w oczekiwany sposób. Aby to naprawić do znacznika należy dodać zdefiniowaną w bootstrapie klasę img-fluid.



Układanie obrazków można poprawić rozszerzając siatkę w klasie pilnującej kolumn.





Gotowe

Klasa col- vs. Rozdzielczość ekranów

