



Bootstrap dla opornych

Paweł Grabiński

Wydział Fizyki i Astronomii

Uniwersytet Wrocławski

3 maja 2018



[Start Bootstrap](#) —

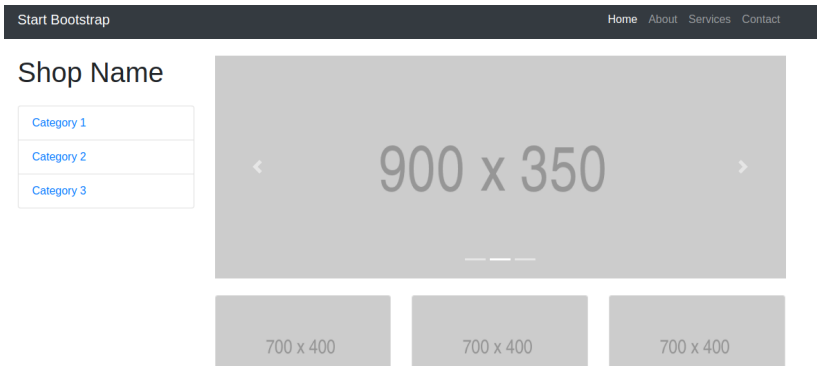
- [Home \(current\)](#)
- [About](#)
- [Services](#)
- [Contact](#)

Shop Name

[Category 1](#) [Category 2](#) [Category 3](#)

- 1.
- 2.
- 3.

900 x 350



Dodano do **<head>** Bootstrap CDN i odpowiednie klasy CSS.

Czym jest i co nam daje Bootstrap?

Framework CSS stworzony na potrzeby wewnętrzne Twittera zgodnie z myślą "mobile first", dzięki czemu jest od podstaw responsywny - dostosowuje się do ekranu na jakim oglądana jest strona. Udostępniony jako open source i regularnie updateowany. Najnowsza wersja jest oznaczona numerem 4.0 (4.X) i nie jest wstecznie kompatybilna łącznie z wersją 4.alpha.

Oferowane funkcjonalności:

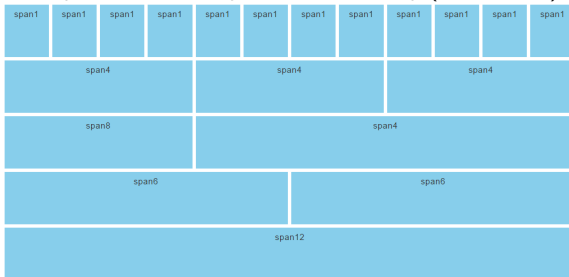
- ▶ Siatka (ang. Grid) - regularne rozmieszczenie zawartości strony w postaci bloków
- ▶ Gotowe klasy CSS - ustandaryzowane klasy stylu typografii, kolorów i modelu pudełkowego
- ▶ Gotowe komponenty CSS - gotowe rozwiązania stylowe, np. pasek nawigacji, przyciski
- ▶ Widżety JavaScript, np. karuzela, rozwijane menu

Responsywność uzyskiwana jest poprzez dostosowywania się styli (klas) w zależności od wielkości ekranu, jaki dostępny jest dla przeglądarki. Reakcje styli nie są ciągłe. Są rozbite na kilka ustalonych szerokości ekranu - breakpointów. W zależności od ekranu mamy dany rodzaj szerokości:

- ▶ xs (extra small) - małe ekrany, klasy nieresponsywne
- ▶ sm (small) - min. **576px**, klasy nieresponsywne i sm
- ▶ md (medium) - min. **768px**, klasy nieresponsywne, sm i md
- ▶ lg (large) - min. **992px**, wszystkie klasy poza xl
- ▶ xl (extra large) - min. **1200px**, wszystkie klasy

W zależności od zamierzonego celu, będziemy stosować klasy styli z suffixem do odpowiedniej wielkości ekranu.

Najważniejszą zaletą Bootstrapa jest możliwość geometrycznego zarządzania treścią poprzez siatkę (ang. grid).



Siatka to nie tylko rozwiązanie estetyczne. Dzięki zastosowaniu kolumn to, co na dużym ekranie wyświetla się jako sąsiadujące, na małych ekranach (np. smartfonów) będzie wyświetlało się jako ułożone kolejno pod sobą części jednej kolumny. Sprawia to, że strona jest naprawdę responsywna i czytelna na małych ekranach.

Osiągamy to następującymi klasami:

- ▶ *container* - klasa potrzebna do zaimplementowania siatki, zawiera pełną samodzielną część treści, ma zadaną szerokość dla wszystkich breakpointów,
- ▶ *container-fluid* - wersja responsywna - dostosowuje się do całej dostępnej szerokości,
- ▶ *row* - klasa wiersza służąc do rozróżnienia treści w pionie, wiersze dzielimy dalej na kolumny,
- ▶ *col* - klasa kolumnowa, siatka ma szerokość 12, wszystkie nadmiarowe elementy zostaną wyświetlone jako następny wiersz, podstawowa klasa *col* dzieli szerokość ekranu równo pomiędzy wszystkie kolumny danego wiersza
- ▶ *col-sm-2*, *col-md-4*, *col-lg-9* - responsywna klasa kolumnowa nadaje szerokość dla danego breakpointa i wyżej, suma szerokości musi równać się 12 lub mniej, żeby pozostały w jednym wierszu.

```
<div class="container-fluid">
  <div class="row" style="height: 150px">
    <div class="col-sm-4 bg-success">
      Wiersz 1 - Kolumna 1
    </div>
    <div class="col-sm-4 bg-primary">
      Wiersz 1 - Kolumna 2
    </div>
    <div class="col-sm-4 bg-info">
      Wiersz 1 - Kolumna 3
    </div>
  </div>
  <div class="row" style="height: 150px">
    <div class="col-sm-3 bg-danger">
      Wiersz 2 - Kolumna 1
    </div>
    <div class="col-sm-3 bg-warning">
      Wiersz 2 - Kolumna 2
    </div>
    <div class="col-sm-3 bg-danger">
      Wiersz 2 - Kolumna 3
    </div>
    <div class="col-sm-3 bg-warning">
      Wiersz 2 - Kolumna 4
    </div>
  </div>
</div>
```

Przykładowy kod dla kontenera zawierającego dwa wiersze podzielone na kolumny.

Podział zadziała dla ekranów **sm** i większych. Dla mniejszych wszystkie kolumny zostaną wyświetlone jako wiersze.

Pierwszy wiersz podzielony jest na 3 kolumny o szerokości $4 - 3 \times 4 = 12$.

Drugi wiersz analogicznie.

Klasy *bg-sth* odpowiadają za kolor tła (patrz dalej).

Kod z powyższego slajdu daje nam poniższy podział geometryczny.



Dla ekranów **xs** zobaczylibyśmy kolejno wiersze W1-K1, W1-K2, W1-K3, W2-K1, W2-K2, etc.

Klasy reprezentujące większość podstawowych opcji CSS:

- ▶ Typografia: grubość czcionki, wyrównanie, rozmiar
- ▶ Kolory: tekstu, tła, ramek, przycisków
- ▶ Model pudełkowy: marginesy, wypełnienie (padding)
- ▶ Pozycje (fixed, sticky&float)
- ▶ Rozmiary
- ▶ Display

Możliwości są ograniczone w porównaniu do wszystkich zdolności CSSa, ale jest to zgodne z zamierzeniem Twittera, by ustandaryzować style w swoich projektach. Dla postronnego użytkownika zysk jest w postaci mniejszej ilości decyzji w trakcie szybkiego prototypowania (tworzenia szkicu) strony lub aplikacji. Klasy zawarte w Bootstrapie można modyfikować.

Klasy typograficzne Bootstrapa dają dostęp do szybkiego i intuicyjnego nadawania stylów różnym znacznikom.

Przykładem jest klasa *lead* używana do znaczników *p*, która w intuicyjny sposób eksponuje paragraf tekstu tak, jak lead w artykule.

Lorem ipsum dolor sit amet, consectetur adipisicing elit.
ipsum? Necessitatibus dicta quaerat, vitae dolorum, dol

Lorem ipsum dolor sit amet, consectetur adipiscing
voluptatem quam ab expedita eum, ipsum? N

```
<!-- PARAGRAPHS -->  
<p>Lorem ipsum dolor sit am
```

```
<!-- LEAD -->  
<p class="lead">Lorem ipsum
```



Klasy wielkości nagłówkowych *hX*.

Mogą być stosowane do znaczników innych od nagłówkowych.

Heading One

Heading Two

Heading Three

Heading Four

Heading Five

Heading Six

```
<!-- HEADINGS -->
<h1>Heading One </h1>
<h2>Heading Two</h2>
<h3>Heading Three</h3>
<h4 class="h2">Heading Four</h4>
<h5>Heading Five</h5>
<h6 class="h1">Heading Six</h6>
```

Wielkości możemy kontrolować także przy pomocy klas *display*.

Display Heading 1

Display Heading 2

Display Heading 3

Display Heading 4

```
<!-- DISPLAY CLASSES -->
<h1 class="display-1">Display Heading 1</h1>
<h1 class="display-2">Display Heading 2</h1>
<h1 class="display-3">Display Heading 3</h1>
<h1 class="display-4">Display Heading 4</h1>
```



Klasy grubości tekstu znane z CSS.

Bold Text

Normal Text

Italic Text

```
<!-- STYLE CLASSES -->  
<p class="font-weight-bold">Bold Text</p>  
<p class="font-weight-normal">Normal Text</p>  
<p class="font-italic">Italic Text</p>
```

Możemy także zmieniać wielkość liter.

make lowercase

MAKE UPPERCASE

Make Capitalized

```
<!-- TEXT TRANSFORMS -->  
<p class="text-lowercase">MAKE LOWERCASE</p>  
<p class="text-uppercase">make uppercase</p>  
<p class="text-capitalize">make capitalized</p>
```

Mamy dostępne typowe formatowania tekstu.

```
<!-- ALIGNMENT -->  
<p class="text-left">Text aligned left</p>  
<p class="text-center">Text aligned center</p>  
<p class="text-right">Text aligned right</p>  
<p class="text-justify">Text justified</p>
```

Klasy cytatów blokowych.

Po oczach go Boo, po oczach!

— Minsc BG2

```
<!-- BLOCKQUOTE WITH FOOTER -->  
<blockquote class="blockquote text-right">  
  <p>Po oczach go Boo, po oczach!</p>  
  <footer class="blockquote-footer">  
    Minsc <cite title="BG2">BG2</cite>  
  </footer>  
</blockquote>
```

Klasy korespondujące z własnością position.

```
<!-- FIXED TOP -->  
<h3 class="fixed-top">Fixed Top</h3>  
<!-- FIXED STICKY -->  
<h3 class="sticky-top">Sticky</h3>  
<!-- FIXED BOTTOM -->  
<h3 class="fixed-bottom">Fixed Bottom</h3>
```

Klasy korespondujące z własnością float.

```
<!-- FLOATS -->  
<div class="float-left">Float left</div><br>  
<div class="float-right">Float right</div><br>  
<div class="float-none">Float none</div><br>
```

Ponieważ Bootstrap jest pisany z myślą "mobile first", to wprowadzono responsywne klasy formatowania i floatów dostosowujące się do wielkości ekranu.

```
<!-- RESPONSIVE ALIGN -->
```

```
<p class="text-sm-right">Right aligned on small or larger</p>  
<p class="text-md-right">Right aligned on medium or larger</p>  
<p class="text-lg-right">Right aligned on large or larger</p>  
<p class="text-xl-right">Right aligned on xl or larger</p>
```

```
<!-- RESPONSIVE FLOATS -->
```

```
<div class="float-sm-right">Float right on small or wider</div><br>  
<div class="float-md-right">Float right on medium or wider</div><br>  
<div class="float-lg-right">Float right on large or wider</div><br>  
<div class="float-xl-right">Float right on extra large or wider</div>
```


Mamy do dyspozycji różne wartości klasy własności display.

Hello Goodbye

```
<!-- TURN BLOCK TO INLINE -->  
<h1 class="d-inline bg-success">Hello</h1>  
<h1 class="d-inline bg-success">Goodbye</h1>
```

Block

```
<!-- TURN INLINE TO BLOCK -->  
<span class="d-block bg-success">Block</span>
```

Hello Hello
This is inline This is inline

```
<!-- INLINE BLOCK -->  
<div class="d-inline-block bg-success">  
  <h3>Hello</h3>  
  This is inline  
</div>  
<div class="d-inline-block bg-success">  
  <h3>Hello</h3>  
  This is inline  
</div>
```

```
<!-- NONE DISPLAY -->
```

```
<span class="d-none">None</span>
```



Istnieje 9 klas kolorystycznych dla tekstu i tła korespondujących do odpowiednich wydarzeń (żółty - ostrzeżenie, zielony - sukces, etc.)

Text Primary Color

Text Secondary Color

Text Success Color

Text Info Color

Text Warning Color

Text Danger Color

Text Light Color

Text Dark Color

Background Primary Color

Background Secondary Color

Background Success Color

Background Info Color

Background Warning Color

Background Danger Color

Background Light Color

Background Dark Color

Background White Color

```
<!-- TEXT COLORS -->
```

```
<p class="text-primary">Text Primary Color</p>
```

```
<p class="text-secondary">Text Secondary Color</p>
```

```
<p class="text-success">Text Success Color</p>
```

```
<p class="text-info">Text Info Color</p>
```

```
<p class="text-warning">Text Warning Color</p>
```

```
<p class="text-danger">Text Danger Color</p>
```

```
<p class="text-light">Text Light Color</p>
```

```
<p class="text-dark">Text Dark Color</p>
```

```
<p class="text-white">Text White Color</p>
```

```
<!-- BACKGROUND COLORS -->
```

```
<div class="bg-primary">Background Primary Color</div>
```

```
<div class="bg-secondary">Background Secondary Color</div>
```

```
<div class="bg-success">Background Success Color</div>
```

```
<div class="bg-info">Background Info Color</div>
```

```
<div class="bg-warning">Background Warning Color</div>
```

```
<div class="bg-danger">Background Danger Color</div>
```

```
<div class="bg-light">Background Light Color</div>
```

```
<div class="bg-dark">Background Dark Color</div>
```

```
<div class="bg-white">Background White Color</div>
```

Regular

Primary

Secondary

Success

Info

Warning

Danger

Light

Dark

Klasa *border* jest podstawową klasą ramki. Możemy także nadawać im kolor przy pomocy klas *border-X*, gdzie X to jeden z podstawowych kolorów Bootstrapa. Możemy także zaokrąglić za pomocą klasy *rounded-KIERUNEK*.

```
<!-- BORDERS -->
<div class="border">Regular</div>
<div class="border border-primary rounded">Primary</div>
<div class="border border-success rounded-top">Success</div>
<div class="border border-danger rounded-left">Danger</div>
```

Bootstrapie zawiera klasy marginesów (*m*) i wypełnień (*p*).

Kierunek:

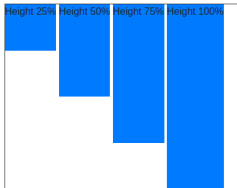
- ▶ l - lewo,
- ▶ p - prawo,
- ▶ t - góra,
- ▶ b - dół,
- ▶ x - lewo i prawo,
- ▶ y - góra i dół,
- ▶ brak - wszystkie

Dodatkowo wybieramy rozmiar od 0-5 lub *auto*, które symetryzuje ułożenie.

```
<!-- MARGINS -->
<span class="mx-5">
<h1 class="mb-2">Margin Bottom 2</h1>
<p class="m-5">Margin All Sides 5</p>
<div class="mx-auto">Center Align</div>

<!-- PADDINGS -->
<p class="p-5">Padding All Sides 5</p>
<span class="px-5">Padding Left & Right 5<
<h1 class="pb-4">Padding Bottom 4</h1>
```

By rozplanować elementy w pełni responsywnie możemy wykorzystać klasy procentowe wysokości i szerokości.



```
<!-- HEIGHT CLASSES -->
<div style="height:300px;border:1px solid #333;">
  <div class="bg-primary d-inline-block h-25">Height 25%</div>
  <div class="bg-primary d-inline-block h-50">Height 50%</div>
  <div class="bg-primary d-inline-block h-75">Height 75%</div>
  <div class="bg-primary d-inline-block h-100">Height 100%</div>
</div>
```



```
<!-- WIDTH CLASSES -->
<div class="w-25 bg-success p-2">Width 25%</div>
<div class="w-50 bg-success p-2">Width 50%</div>
<div class="w-75 bg-success p-2">Width 75%</div>
<div class="w-100 bg-success p-2">Width 100%</div>
```

Primary

Secondary

Success

Info

Warning

Danger

Light

Dark

[Link](#)

Klasa *btn* jest podstawową klasą przycisków. Możemy także nadawać im kolor przy pomocy klas *btn-X*, gdzie X to jeden z podstawowych kolorów Bootstrapa. Istnieje wiele innych klas modyfikujących wygląd przycisków - warto zajrzeć do dokumentacji.

```
<!-- BUTTONS -->
```

```
<button type="button" class="btn btn-primary">Primary</button>
```

```
<button type="button" class="btn btn-secondary">Secondary</button>
```

```
<button type="button" class="btn btn-success">Success</button>
```

```
<button type="button" class="btn btn-info">Info</button>
```

```
<button type="button" class="btn btn-warning">Warning</button>
```

```
<button type="button" class="btn btn-danger">Danger</button>
```

```
<button type="button" class="btn btn-light">Light</button>
```

```
<button type="button" class="btn btn-dark">Dark</button>
```

```
<button class="btn btn-link">Link</button>
```

Navbar

Home About Services Contact

Prosty komponent
CSS do budowania
czytelnych menu.
Istnieje wiele
rozszerzeń jak
rozwijalne listy
linków czy pola
wyszukiwania.

```
<!-- SIMPLE NAVBAR -->
<nav class="navbar navbar-expand-sm navbar-dark bg-dark mb-3">
  <div class="container">
    <a class="navbar-brand" href="#">Navbar</a>
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link" href="#">Home</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">About</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Services</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Contact</a>
      </li>
    </ul>
  </div>
</nav>
```

Prosty komponent poprawiający wygląd zwykłych list.

My List Item One
My List Item Two
My List Item Three
My List Item Four
My List Item Five

```
<!-- LIST GROUP -->  
<ul class="list-group">  
  <li class="list-group-item">My List Item One</li>  
  <li class="list-group-item">My List Item Two</li>  
  <li class="list-group-item">My List Item Three</li>  
  <li class="list-group-item">My List Item Four</li>  
  <li class="list-group-item">My List Item Five</li>  
</ul>
```




Komponent poprawiający wygląd formularzy.

Name

Email address

Your email will not be shared with anyone

```
<!-- FORM -->
<form>
  <div class="form-group">
    <label for="name">Name</label>
    <input class="form-control" type="text" id="name" placeholder="Enter name">
  </div>
  <div class="form-group">
    <label for="email">Email address</label>
    <input class="form-control" type="email" id="email" placeholder="Enter email">
    <small class="form-text text-muted">
      Your email will not be shared with anyone
    </small>
  </div>
</form>
```

Komponent dający możliwość dodawania okien komunikatów, które mogą zostać zamknięte po naciśnięciu x. Idealne do reakcji na pozytywne lub nieudane wypełnienie formularza i tym podobnych.

Coś poszło nie tak!



```
<!-- DISMISSABLE ALERT -->
<div class="alert alert-danger alert-dismissible fade show">
  <button type="button" class="close" data-dismiss="alert">
    <span>&times;</span>
  </button>
  <strong>Dismissable</strong> Blog post added
</div>
```

Komponent poprawiający
wygląd tabel. Posiada wiele
klas estetyki takich jak
obramowane komórki,
wiersze o zmiennych
kolorach, zmianę koloru po
najechnaniu na wiersz, etc.

#	First Name	Last Name	Email
1	John	Doe	jdoe@gmail.com
2	Will	Johnson	will@yahoo.com

```
<!-- BORDERED TABLE -->
<table class="table table-bordered">
  <thead>
    <tr>
      <th>#</th><th>First Name</th>
      <th>Last Name</th><th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th><td>John</td>
      <td>Doe</td><td>jdoe@gmail.com</td>
    </tr>
    <tr>
      <th scope="row">2</th><td>Will</td>
      <td>Johnson</td><td>will@yahoo.com</td>
    </tr>
  </tbody>
</table>
```

Komponent dający możliwość graficznego odseparowania treści.
W zależności od klas stylistycznych przyjmuje różne formy -
wyświetlania tekstu, cytatów, obrazków, etc.

Card Title

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aut, perspiciatis.

Read More

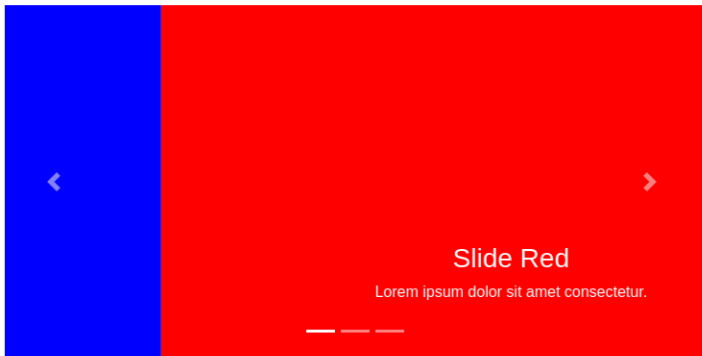
```
<!-- CARD WITH HEADER -->
<div class="card">
  <div class="card-header">
    My Card
  </div>
  <div class="card-body">
    <h4 class="card-title">Card Title</h4>
    <p class="card-text">Lorem ipsum dolor sit amet, consecte
      adipisicing elit. Aut, perspiciatis.</p>
    <a href="#" class="btn btn-info">Read More</a>
  </div>
</div>
```

Widżety w Bootstrapie są funkcjonalnościami napisanymi przy użyciu JavaScriptu - wymagają załączenia JQuery i pliku źródłowego .js od Bootstrapa:

- ▶ Carousel - karuzela (slider) dynamiczna galeria obrazów
- ▶ Popover - blok tekstu wyświetlający się po naciśnięciu przycisku
- ▶ Tooltip - blok tekstu wyświetlający się po najechaniu kursorem na pewien obszar
- ▶ Modal - okienka "dialogowe" pojawiające się nad treścią naszej strony, dobre do komunikatów lub okien logowania
- ▶ Collapse - rozwijalny obszar strony "wydłużający" treść

Wyjątkowo przydatnym i efektownym widżetem jest slider obrazków - dynamiczna galeria obrazków zmieniających się po ustalonym interwale.

Możemy dodać do niego przyciski kontrolne i podpisy.



Poniżej znajduje się kod do najprostszej wersji karuzeli bez przycisków kontrolnych i podpisów.

To oraz modyfikację dynamiki karuzeli można znaleźć w dokumentacji.

```
<!-- SIMPLE SLIDER -->
<div id="slider1" class="carousel slide mb-5" data-ride="carousel">
  <div class="carousel-inner" role="listbox">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
</div>
```



Istnieje jeszcze wiele innych klas i szczegółów związanych z Bootstrapem. Dlatego warto dobrze zapoznać się z dokumentacją i istniejącymi gotowymi szablonami.

Bootstrap jest standardem branży i najpopularniejszym frameworkiem CSS - alternatywy to Foundation, Skeleton czy Materialize. Istnieje wiele jego integracji z popularnymi frameworkami Java Scriptowymi jak Angular czy React.

Stwórz stronę dowolnego wydarzenia studenckiego.

Ma się składać z 3 podstron:

1. Strona główna zawierająca opis wydarzenia.

Wykorzystaj następujące elementy:

- ▶ klasę Jumbotron lub Card,
- ▶ tabelę wykorzystującą odpowiednie klasy,
- ▶ listę z odpowiednimi klasami.

2. Formularz rejestracyjny wykorzystujący:

- ▶ jak najwięcej różnych znaczników wejściowych,
- ▶ przycisk zapisu i czyszczenia formularza,
- ▶ alerty o zapisaniu i wyczyszczeniu.

3. Galerię zdjęć korzystającą z widżetu Carousel zawierającą minimalnie 3 grafiki z podpisami.

Strona ma posiadać pasek nawigacji, ma być napisana z użyciem siatki i być w pełni responsywna.



[Bootstrap 4.X Documentation](#)



[Bootstrap Crash Course by Traversy Media](#)



[Start Bootstrap: Free Templates](#)



[WrapBootstrap: Templates Marketplace](#)



[Bootstrap grid](#)

Paweł Grabiński

pawelrgrabinski@gmail.com

pgrabinski.pl