

Technologie tworzenia stron i aplikacji WWW

wykład 4 – RWD

dr inż. Grzegorz Rogus



Responsive Web Design

Składowe nowoczesnej aplikacji webowej

czyli jak pisać dobry kod HTML i CSS ... i budować dobre strony internetowe

PROJEKTOWANIE
RESPONSYWNE

- elastyczny układ (layouts)
- media queries
- responsywne zdjęcia
- właściwe jednostki
- desktop-first a mobile-first

JAKOŚĆ
KODU

- Kod skalowany i łatwy do zarządzania
- czystość formy
 - łatwy do zrozumienia
 - łatwość rozbudowy
 - wielokrotnego użycia
 - organizacja struktury plików
 - nazewnictwo klas i struktur
 - struktura HTML

WYDAJNOŚĆ

- mniej zapytań HTTP
- mniej kodu
- kompresja kodu
- użycie preprocessora CSS
- mniej obrazów
- kompresja obrazów

Problem - wygląd strony na różnych nośnikach



Rozwiązania

- **Adaptive Design** – serwer zwraca spreparowaną stronę na podstawie informacji o urządzeniu klienckim.
Jeden adres URL.
- **Separate Mobile Site (.m)**- oddzielny adres URL dla stron mobilnych
- **Responsive Web Design (RWD)**



Nowoczesne aplikacje internetowe - FrontEnd

RWD Responsive Web Design [Reaktywne Projektowanie Stron] - jest to nowoczesne podejście do projektowania stron internetowych, w którym programista zapewnia, że niezależnie od tego na jakie urządzenie strona zostanie podana wyświetli się ona prawidłowo. W tym kontekście programista zapewnia dostosowanie się wyglądu strony, rozmiaru oraz układu strony do rozmiaru okna przeglądarki w którym będzie wyświetlana.

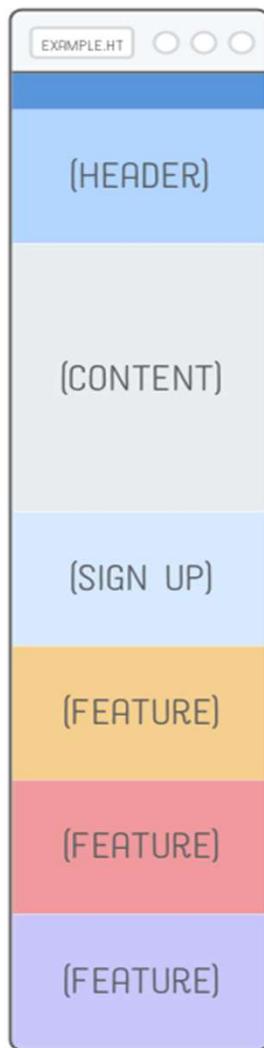
Jak to osiągnąć ?

Na przykład:



Idea RWD

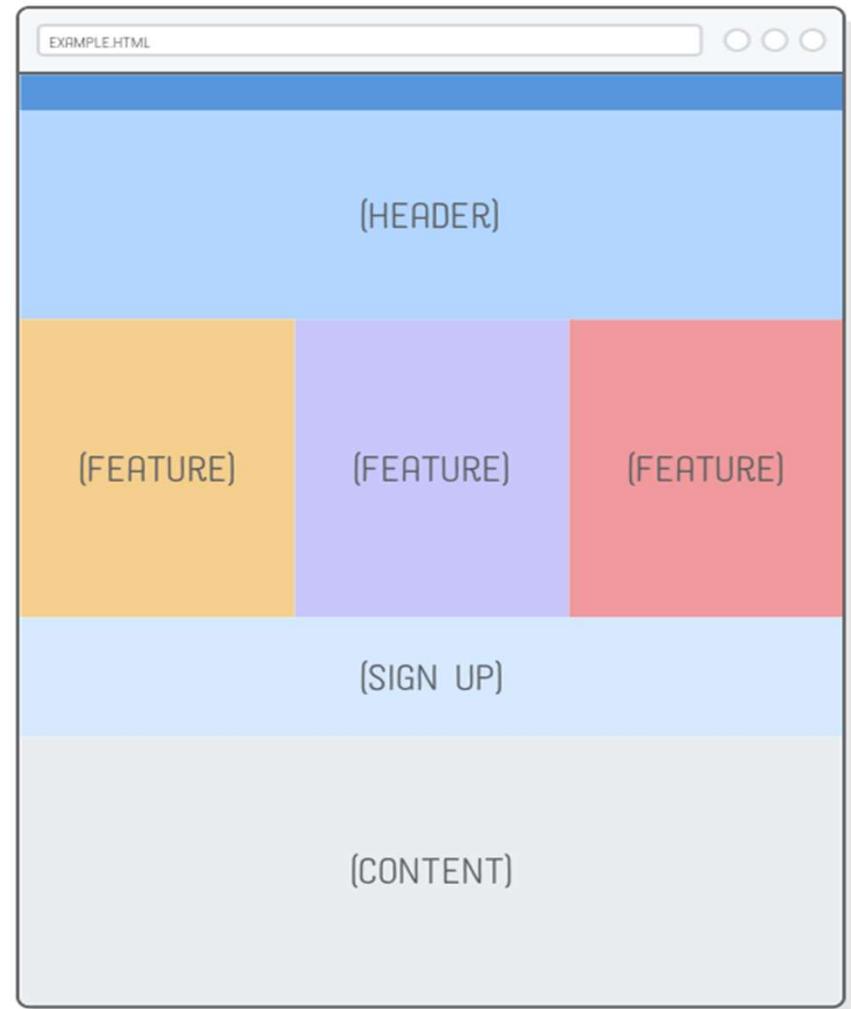
MOBILE



TABLET



DESKTOP



Responsive Design

- Definiowany przez 3 najważniejsze elementy
 - Flexible grid-based layout – układ siatkowy
 - style warunkowe oparte o tzw. Media queries (CSS3)
 - Zdjęcia o elastycznym rozmiarze (Images resize)
- www.alistapart.com/articles/responsive-web-design/

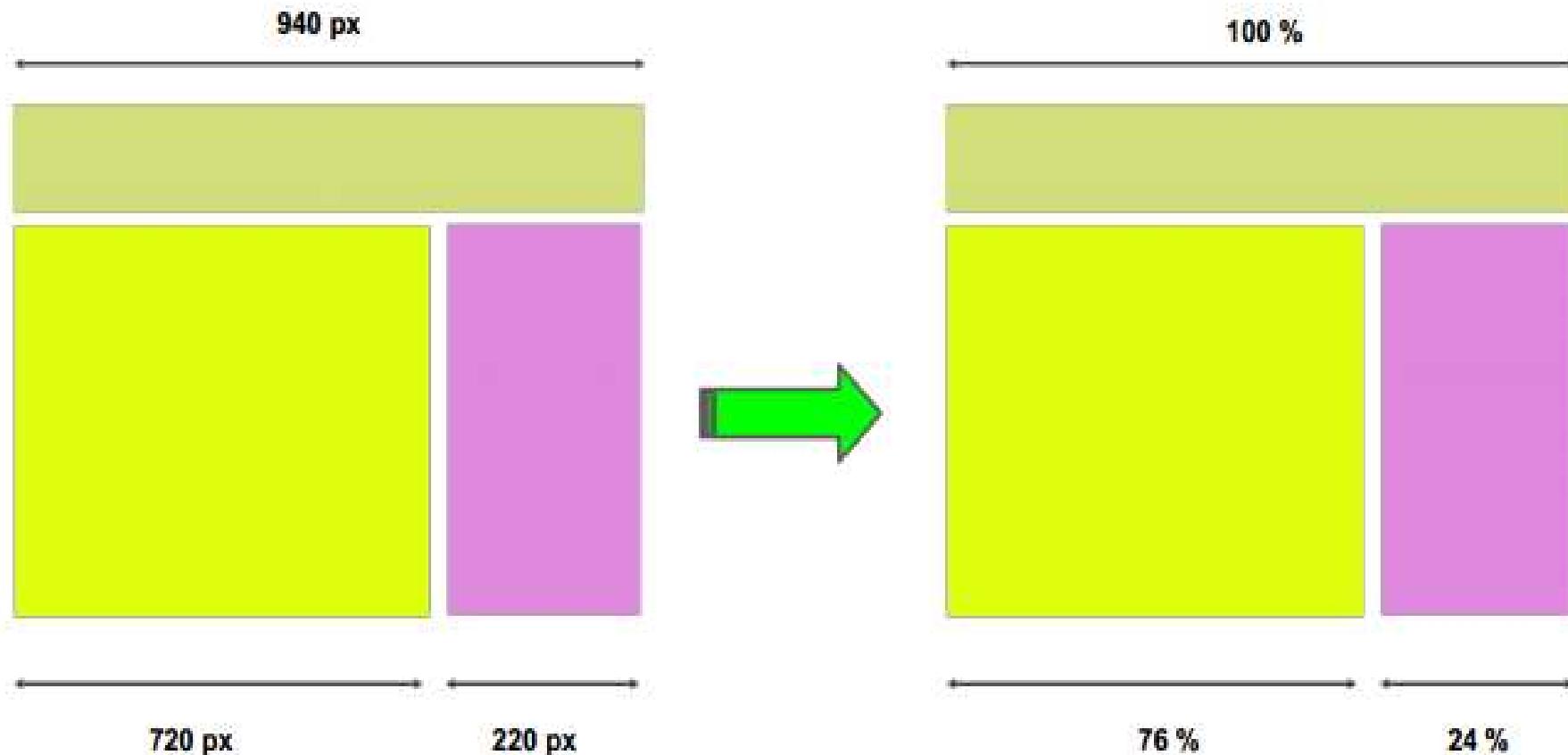
The screenshot shows the header and top portion of the article page. On the left is the A List Apart logo, featuring a large stylized 'A' above the words 'LIST apart' in a serif font, all set against a laurel wreath. Below the logo is the tagline 'FOR PEOPLE WHO MAKE WEBSITES'. To the right of the logo is a small black decorative element resembling a bird or a gear. The main title 'Responsive Web Design' is centered in a large, bold, reddish-brown sans-serif font. Below it, the author's name 'by ETHAN MARCOTTE' is written in a smaller green sans-serif font. At the bottom of the screenshot, there is a horizontal line followed by the text 'Published in: CSS, Layout, User Interface Design, Mobile, Mobile Design, Mobile Development' and two links: 'Discuss this article >' and 'Share this article >'. The background of the page is white.

1. Flexible grid-based layout

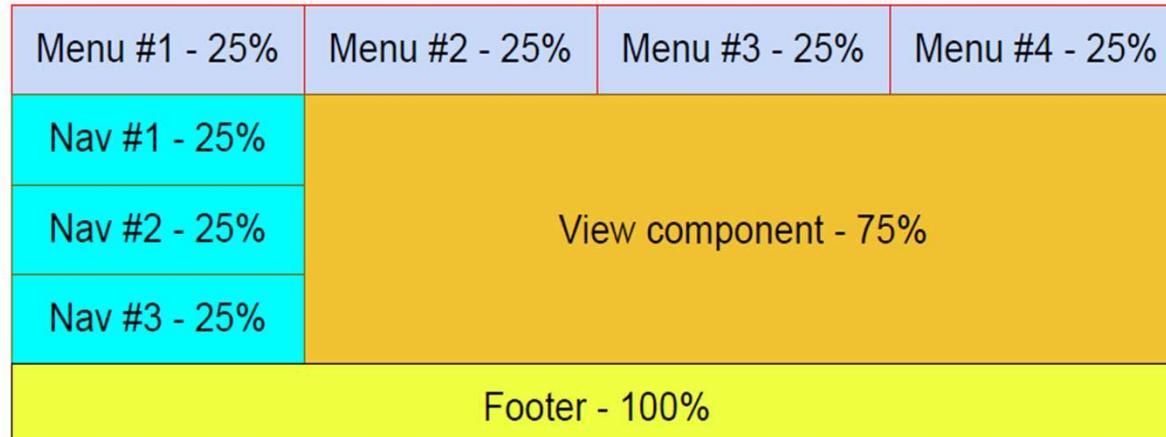
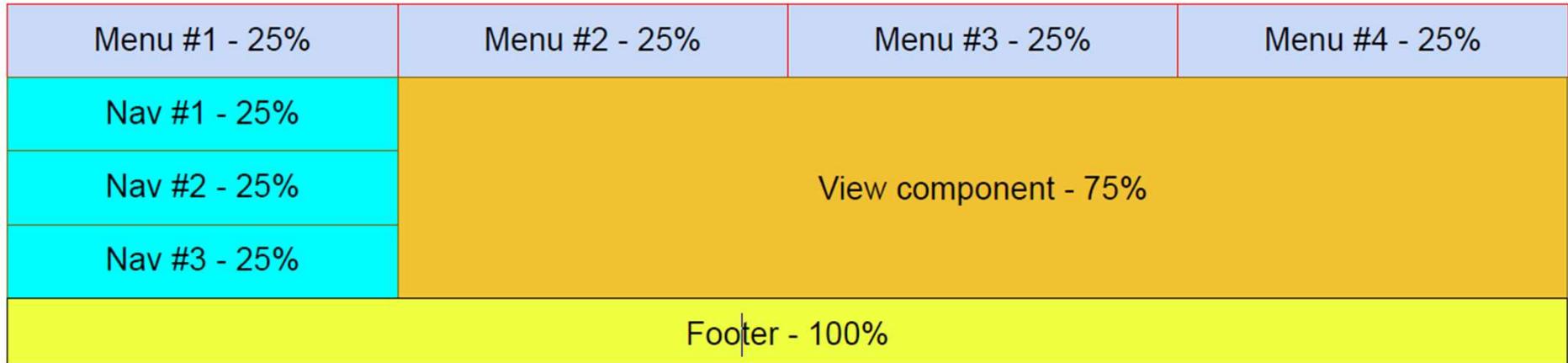
Chcąc zaprojektować "elastyczną" stronę dopasowującą się do całego ekranu powinniśmy zrezygnować z jednostek bezwzględnych wyrażonych np. w px i zastąpić je jednostkami względnymi w procentach %.

Pozwoli to uzyskać dopasowany układ do różnych szerokości ekranów.

Atrybuty *max-width* powinny być używane często zwłaszcza w przypadku elementów, które mogą nam "rozpychać layout" tj. obrazów, osadzonych odtwarzaczy video etc.



Układ elastyczny - przykład



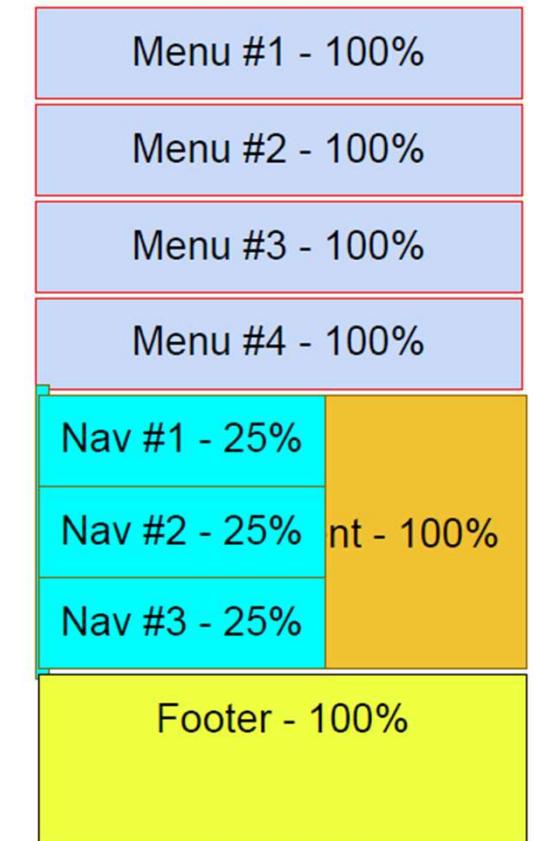
Ale czasami sam układ elastyczny nie wystarczy

Reguły CSS:

```
@media only screen and (min-width:768px) {  
/* layout do tabletów i desktopów */ }
```

```
@media only screen and (max-width: 767px)  
{ /* smartfony */ }
```

```
@media only screen and (max-width:767px)  
and (orientation: portrait) {  
/* smartfony orientacja portrait */ }
```

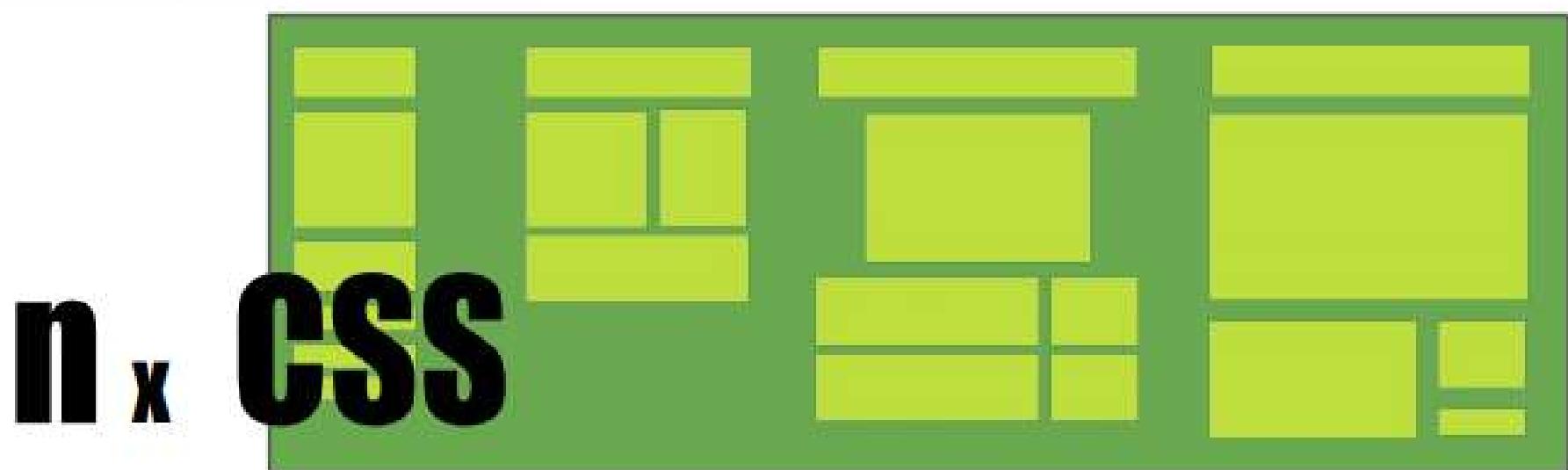


2. Style warunkowe



ZASADA:

W ogólności zasada jaka powinna przyświecać tworzeniu stron zgodnie z metodologią RWD jest utworzenie jednego pliku HTML, a dla formatowania jego wyglądu wiele “layoutów” w CSS które będą zawierać odpowiednie reguły wyświetlania strony w zależności od rozdzielczości ekranu.



RWD

HTML CONTENT



Reguła @media jest używana, aby definiować różne style dla różnych rodzajów mediów/urządzeń.

MOBILE CSS



MOBILE SITE

TABLET CSS



TABLET SITE

DESKTOP CSS



DESKTOP SITE

Zapytania mediów - media queries

- Technika CSS wprowadzona w CSS3.
- Używa reguły @media żeby dołączyć jakiś blok właściwości CSS jeśli zdefiniowany warunek jest prawdziwy.

Przykłady:

- jeśli okno przeglądarki jest mniejsze niż 500 pikseli to kolor tła zmieni się na jasnoniebieski
- ```
@media only screen and (max-width: 500px) {
 body {
 background-color: lightblue;
 }
}
```
- kiedy ekran (okno przeglądarki) stanie się mniejsze niż 768 pikseli, każda kolumna będzie miała szerokość 100%:

```
@media only screen and (max-width: 768px) {
 [class*="col-"] { /* dla mobile phones: */
 width: 100%;
 }
}
```

# Media Query – informacje

Zapytania mediów sprawdzają możliwości urządzenia, mogą być użyte do sprawdzenia wielu rzeczy, np.:

- szerokości i wysokości okna roboczego,
- szerokości i wysokości ekranu urządzenia
- orientacji (czy tablet/telefon jest poziomo czy pionowo?)
- rozdzielczości i wielu innych.

Zapytanie mediów standardowo składa się z:

- **typu lub grupy** medium,
- słowa kluczowego **and**
- **cechy** medium umieszczonej w nawiasie.

## Operatory logiczne

- AND
- przecinek
- NOT

# REGUŁY MEDIA QUERIES

```
@media warunek1, warunek2, (...), warunek-n {
 /* kod css dla urządzeń spełniających podane warunki */
}

@media screen {
 * { font-family: sans-serif }
}

@media screen and (min-width: 400px) and (max-width: 700px)
{ ... }

@media not screen and (color) {
 /* kod css dla urządzeń nie posiadających kolorowego ekranu */
}
```

# Inne przykłady

```
@media (pointer: coarse) and (min-width: 600px) {
 ...
}

@media (min-width: 900px), (orientation: landscape) {
 ...
}

@media not (orientation: landscape) {
 ...
}
```

# Typy mediów

- **all** - dla wszystkich typów urządzeń
- **print** - używane dla drukarek (dla podglądu wydruku)
- **screen** - dla ekranów komputera, tabletu, smartfonów itp.
- **speech** - dla czytników ekranu, które czytają stronę
- **aural** - **nieaktualne**, dla syntezatorów mowy i dźwięku
- **braille** - **nieaktualne**, dla urządzeń przeznaczonych dla niewidomych
- **embossed** - **nieaktualne**, dla drukarek brailla
- **handheld** - **nieaktualne**, dla bezprzewodowych urządzeń ręcznych
- **projection** - **nieaktualne**, dla prezentacji projektorowych
- **tty** - **nieaktualne**, dla dalekopisów, terminali z ograniczonymi możliwościami wyświetlania
- **tv** - **nieaktualne**, dla telewizora

# MEDIA QUERIES

Reguły media queries bazują na cechach nośnika, takich jak:

- Szerokość: width, min-width, max-width
- Wysokość: height, min-height, max-height
- Szerokość urządzenia: device-width, min-device-width, max-device-width
- Wysokość urządzenia: device-height, min-device-height, max-device-height
- Orientacja: orientation
- Proporcje obrazu: aspect-ratio, min-aspect-ratio, max-aspect-ratio
- Proporcje obrazu urządzenia: device-aspect-ratio, min-device-aspect-ratio, max-device-aspect-ratio
- Głębia koloru: color, min-color, max-color
- Paleta kolorów: color-index, min-color-index, max-color-index
- Urządzenia monochromatyczne: monochrome, min-monochrome, max-monochrome
- Rozdzielczość: resolution, min-resolution, max-resolution
- Technika wyświetlania: scan
- Siatka: grid

# Sposoby implementacji media queries?

## Trzy sposoby implementacji:

1. Użycie importu stylów zewnętrznych w pliku CSS - @import.

```
@import url(style600.css) screen and (min-width: 600px);
```

2. Użycie Media Queries bezpośrednio w arkuszu stylów

```
@media screen and (min-width: 600px){ #section{ width: 550px; } }
```

3. Użycie query jako atrybutu w stylu zewnętrzny

```
<link rel="stylesheet" type="text/css" media="screen and (max-device-width: 600px)" href="style600.css" />
```

# REGUŁA MEDIA QUERIES

```
<head>
```

```
 <meta name="viewport" content="width=device-width, initial-
scale=1">
```

```
</head>
```

- width=device-width – ustawia szerokość strony adekwatną do szerokości wyświetlacza urządzenia
- initial-scale=1 – ustawia poziom powiększenia podczas pierwszego uruchomienia strony.

Wyłączenie zoomowania dla urządzeń mobilnych:  
**zasada mobile-first**

```
<meta name = 'viewport' content='width=device-width, initial-scale=1,
maximum-scale = 1'>
```

Meta-tag *viewport* zapewnia poprawne dostosowanie szerokości oraz brak powiększenia. Jeżeli zapomnisz go umieścić, strona będzie prezentować się źle.

# Desktop-First vs Mobile-First

# Desktop-First

```
.example-section {
display: flex;
width: 60%;
}
.example-section__text {
font-size: 32px;
}
@media (max-width: 768px) {
.example-section {
flex-direction: column;
width: 80%;
}
```

# Mobile-First

```
.example-section {
 display: flex;
 flex-direction: column;
 width: 80%;
}
.example-section__text {
 font-size: 24px;
}
@media (min-width: 768px) {
.example-section {
 flex-direction: row;
```

# Jak ustalić breakpointy?

- Podział popularnych urządzeń na kilka grup
- BP zgodne z 'wymaganiami' designu

# Typowe ustawienia

Label	Layout Width
Smartphones	480px and below
Portrait Tablets	480px to 768px
Landscape Tablets	768px to 940px
Default	940px and up
Large Screens	1210px and up

# PUNKTY PRZEŁAMANIA (RWD BREAKPOINTS)

```
/*===== Mobile First Method =====*/
/* Custom, iPhone Retina */
@media only screen and (min-width : 320px) { ... }

/* Extra Small Devices, Phones */
@media only screen and (min-width : 480px) { ... }

/* Small Devices, Tablets */
@media only screen and (min-width : 768px) { ... }

/* Medium Devices, Desktops */
@media only screen and (min-width : 992px) { ... }

/* Large Devices, Wide Screens */
@media only screen and (min-width : 1200px) { ... }
```

# Optymalizacja zdjęć/grafik

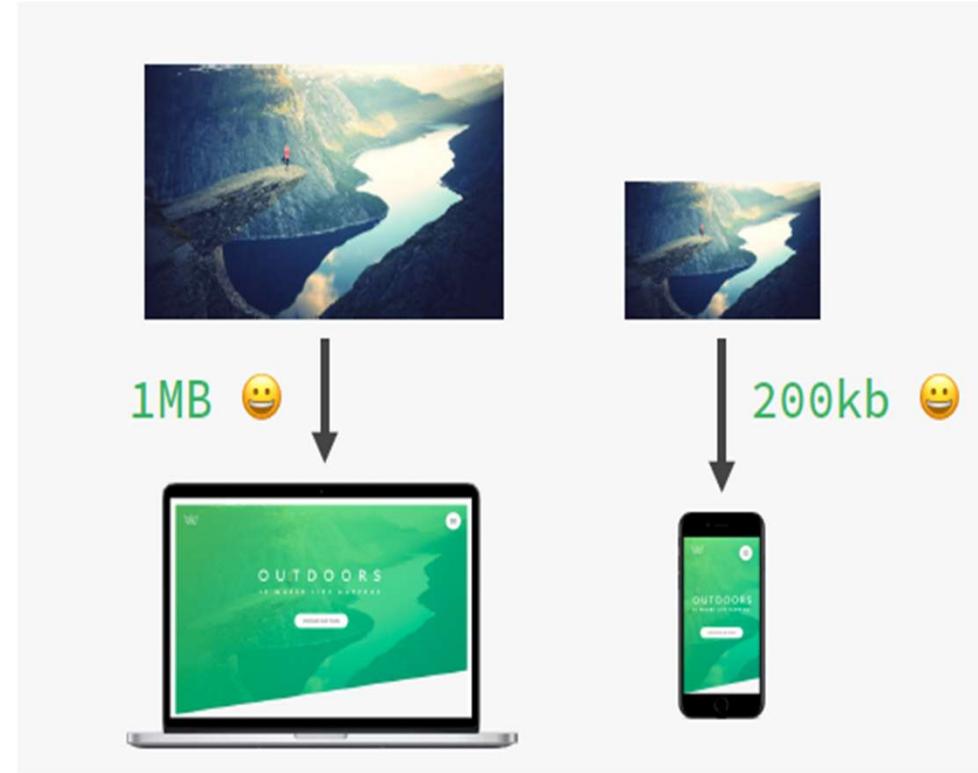
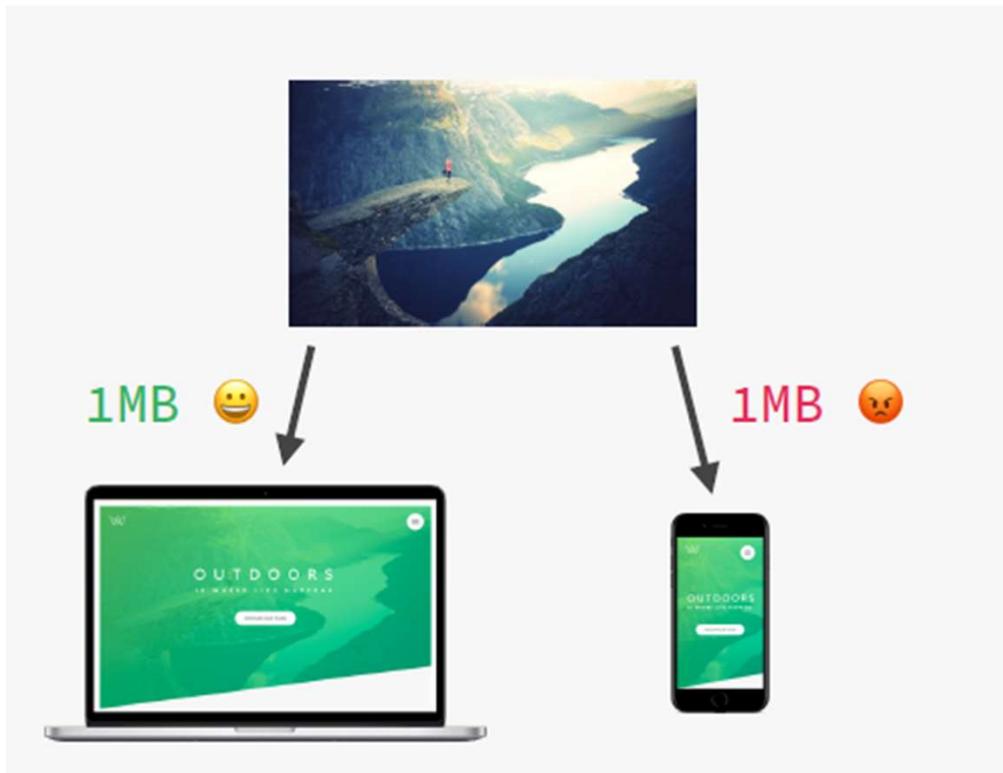
- Kompresja plików

zdjecia.jpg	736.6 KB	Finished	267.7 KB	<a href="#">download</a>	-64%
zdjecia@2x.jpg	2.5 MB	Finished	329.8 KB	<a href="#">download</a>	-87%
zdjecia@3x.jpg	5.0 MB	Finished	499.4 KB	<a href="#">download</a>	-90%

- Różne pliki dla różnych ekranów
- Skalowanie zdjęć i filmów video w celu dopasowania do rozmiaru obszaru na ekranie

```
img { width: 100%; height: auto; }
video { width: 100%; height: auto; }
```

# Różne pliki dla różnych ekranów



<https://responsivebreakpoints.com/>

# Zalecenia

- Używaj obrazów, które najlepiej pasują do cech wyświetlacza. Weź pod uwagę rozmiar ekranu, rozdzielcość urządzenia i układ strony.
- Gdy chcesz określić różne obrazy wyświetlane w zależności od cech urządzenia (tzn. dostosować grafikę), użyj elementu **picture**.
- Użyj atrybutu **srcset** i deskryptora x w elemencie **img**, by przy wyborze obrazu podpowiedzieć przeglądarce, której rozdzielcości najlepiej użyć.

# Grafika w zależności od gęstości pikseli

```

```

# Grafika w zależności od rozmiaru ekranu

```

```

**sizes** są podobne do zapytań o media, opisujące, ile miejsca zajmuje obraz w rzutni.

- jeśli rzutnia jest większa niż 1200px, obraz ma dokładnie 800px
- jeśli rzutnia ma wielkość między 1200px a 600px, obraz zajmuje 48% rzutni
- jeśli rzutnia jest mniejsza niż 640px, obraz zajmuje 580 px

**srcset** po prostu mówi przeglądarce, jakie obrazy mamy dostępne i jakie są ich rozmiary.

- img/foto.jpg ma szerokość 480 pikseli,
- img/foto2.jpg ma szerokość 700 pikseli,

# Grafika w zależności od 'wizji artystycznej'

```
<picture>
<source media="(max-width: 799px)"
srcset="img/foto.jpg">
<source media="(min-width: 800px)"
srcset="img/foto2.jpg">

</picture>
```

Responsywny obrazek <PICTURE, SOURCE>

# Picture – nowy znacznik

<~~picture~~<sup>Zastosowanie 1:</sup> Dopasowanie formatu

```
<source srcset="lokalizacja pliku w formacie 1"
type="format 1">
 <source srcset="lokalizacja pliku w formacie 2"
type="format 2">
 (...)

</picture>
```

```
<picture>
<source srcset="zdjecie.webp" type="image/webp">
<source srcset="zdjecie.jxr" type="image/vnd.ms-photo">

</picture>
```

Responsywny obrazek <PICTURE, SOURCE>

# Picture – nowy znacznik

Zastosowanie 2: Dopasowanie kadru

<picture>

    <source srcset="lokalizacja pliku 1" media="zapytanie mediów 1">

    <source srcset="lokalizacja pliku 2" media="zapytanie mediów 2">

    (...)

</picture>

<picture>

<source srcset="duze-zdjecie.jpg" media="(min-width: 1024px)">

<source srcset="srednie-zdjecie.jpg" media="(min-width: 400px)">



</picture>

# Techniki implementacji

Progressive Enhancement  
(Stopniowe ulepszanie)

vs

Graceful Degradation (Wdzięczna Degradacja)

# Progressive Enhancement – Stopniowe Ulepszanie

- W pierwszej kolejności skupienie na gorszych warunkach – starsza przeglądarka, wolny internet
- Ograniczenie wykorzystania najnowszych technologii, np. CSS Grid
- Stopniowe dodawanie 'fajerwerków'
- Związane z podejściem mobile-first

Internet Explorer >> Chrome

# Graceful Degradation – Wdzięczna Degradacja

- Założenie idealnych warunków użytkowania – nowoczesna przeglądarka, szybki internet
- Korzystanie z nowoczesnych technologii
- Zapewnienie minimum funkcjonalności w gorszych warunkach
- Związane z podejściem desktop-first

Chrome >> Internet Explorer

# Co wybrać?

To zależy!

- Target
- Wymagania biznesowe
- Deadline

# Jak zapewnić kompatybilność wsteczną?

- Fallbacki w CSS
- Feature query – @supports
- Polyfills
- Transpilers, np. Babel (JS)
- Informacja: 'Zmień przeglądarkę'

```
@supports (właściwość: wartość) { // CSS
}
.modal-overlay {
 opacity: .8;
}
@supports (backdrop-filter: blur(5px)) {
 .modal-overlay {
 backdrop-filter: blur(5px);
 opacity: .3;
 }
}
```

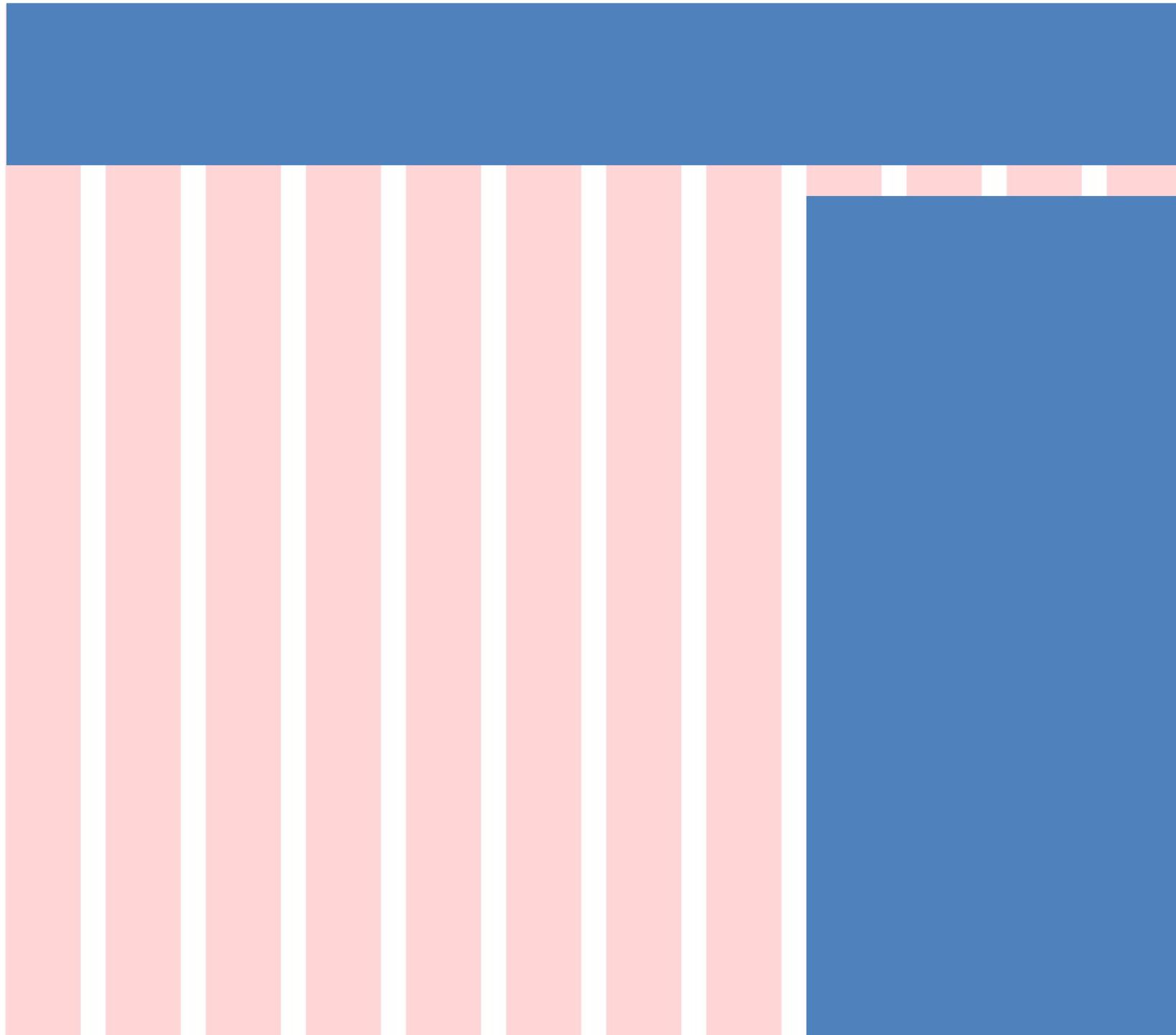
# System siatkowy - Grid System

- Umożliwia poprawne rozmieszczenie zawartości strony WWW.
- Dzieli on ekran na rzędy i kolumny, które następnie można wykorzystać do projektowania różnych układów treści.
- Po zdefiniowaniu rzędów i kolumn możemy zdecydować o rozlokowaniu poszczególnych elementów HTML.

# Zaplanowanie kompozycji szablonu



- Umożliwia poprawne rozmieszczenie zawartości strony WWW.
- Dzieli on ekran na rzędy i kolumny, które następnie można wykorzystać do projektowania różnych układów treści.
- Po zdefiniowaniu rzędów i kolumn możemy zdecydować o rozlokowaniu poszczególnych elementów HTML



# TWITTER BOOTSTRAP

Twitter Bootstrap to jeden z najbardziej popularnych framework'ów do tworzenia responsywnych i przyjaznych urządzeniom mobilnym stron internetowych.



**Bootstrap to framework oparty o język HTML, kaskadowe arkusze stylów CSS oraz język skryptowy JavaScript. To zestaw gotowych rozwiązań, bibliotek i narzędzi, które można wykorzystać przy tworzeniu stron internetowych albo ich prototypów.**

# Inne Frameworki

- **Foundation** - <https://foundation.zurb.com>, drugi najpopularniejszy, używany np. przez Facebook'a
- **Semantic UI** - <https://semantic-ui.com>, innowacyjny, jego struktura i przyjęte nazewnictwo elementów jest bardzo uporządkowane logicznie, semantycznie
- **Pure.css** - <https://purecss.io>, lekki, posiada modułowa budowę, napisany w czystym CSS, udostępnia tylko podstawowe style, elementy wyglądów, taki szkielet
- **Skeleton** - <http://getskeleton.com> - maksymalnie prosty responsywny szkielet aplikacji
- **HTML5 Boilerplate** - <https://html5boilerplate.com>
- **Cascade Framework** - <http://www.cascade-framework.com>
- **W3.CSS** - <https://www.w3schools.com/w3css> - mniejszy, prostszy w użyciu

## ZALETY

- Łatwy w użytku: wymaga jedynie podstawowej wiedzy o HTML i CSS.
- Responsywny: CSS dostosowany do smartfonów, tabletów i urządzeń desktopowych.
- Podejście mobile-first: style dla telefonów tworzą podstawę frameworka;
- Kompatybilny: oferuje wsparcie i kompatybilność pomiędzy wszystkimi popularnymi przeglądarkami.

## WADY

- Nadmierna standaryzacja: wszyscy używają Bootstrap, więc strony stają się podobne;
- Rozmiar: w wersji pełnej główny plik css po zminimalizowaniu zajmuje 120 kb;
- Brak krytyczyzmu – nie zawsze potrzeba tak dużej biblioteki do realizacji konkretnego zadania.
- Konflikty między różnymi bibliotekami, np. jQuery-ui.

# Gdzie szukać informacji

- <http://getbootstrap.com/>
- <https://kursbootstrap.pl/>
- <https://www.w3schools.com/bootstrap4/default.asp>

# Bootstrap - instalacja

Mamy kilka możliwości wykorzystania Bootstrapa:

- pobranie skompilowanych i zminifikowanych plików CSS i JS - wrzucamy do projektu i używamy
- samodzielna komplilacja plików źródłowych (Sass i JavaScript) – większe możliwości dostosowania biblioteki ale trzeba się tym pobawić
- wykorzystanie biblioteki dostępnej poprzez CDN:

```
<link rel="stylesheet"
 href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
...
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"> </script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"> </script>
```

- instalacja z menagera pakietów, np.: \$ npm install bootstrap

# Bootstrap w Angularze

```
npm install bootstrap
npm install jquery
npm install popper.js
```

## Użycie angular.json

w sekcji build i potem w sekcji options odnajdujemy styles i tam umieszczamy link do Boot

```
"styles": [
 "node_modules/bootstrap/dist/css/bootstrap.css",
 "src/styles.css"
],
```

Jeżeli używasz Sass-a to zmieniamy nieco ścieżkę, ponieważ pliki scss znajdują się w inn

```
"styles": [
 "node_modules/bootstrap/scss/bootstrap.scss",
 "src/styles.scss"
],
```

Natomiast w sekcji script umieszczamy linki do plików js:

```
"scripts": [
 "node_modules/jquery/dist/jquery.slim.js",
 "node_modules/popper.js/dist/umd/popper.js",
 "node_modules/bootstrap/dist/js/bootstrap.js"
]
```

# Bootstrap w Angularze

Warto wtedy rozważyć i zainstalować dodatkową bibliotekę jak:

[Ngx-Bootstrap](#) lub [Ng-Bootstrap](#).

# Minimalizacja (ang. minification) skryptów

Bootstrap dostępny jest w dwóch wersjach:

- pełnej,
- po minifikacji.

```
var a = [];
for(var i = 0; i < 20; i++) {
 a[i] = i;
}
```

```
for(var i=0,a=[];i<20;i++){a[i]=i;}
```

Minifikacja (ang. **minification**) – „proces mający na celu zmniejszenie kodu źródłowego poprzez usunięcie niepotrzebnych znaków bez zmianiania jego funkcjonalności”.

Minifikacja polega głównie na:

- Usunięciu tabulacji, spacji, znaków nowej linii (oraz innych białych znaków),
- Usunięciu komentarzy,
- Skracania do minimum nazw zmiennych, funkcji, bądź innych nazwanych bloków kodu.

Minifikacja nie jest ściśle związana z zaciemnianiem kodu (ang. *obfuscation*)

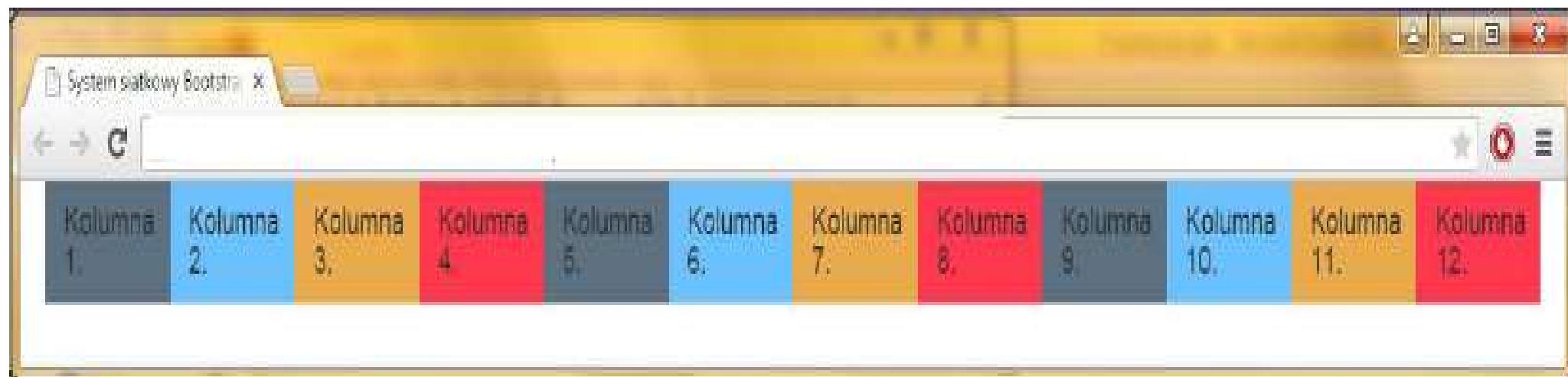


**System siatkowy Bootstrapa**

**BOOTSTRAP GRID SYSTEM**

# Bootstrap Grid

System siatkowy dzieli ekran na kolumny – do 12 w każdym rzędzie, a szerokość kolumny jest uzależniona od wielkości ekranu, na którym jest wyświetlana strona.



col1	col1	col1	col1	col1	col1	col1	col1	col1	col1	col1	col1	col1
col 4				col 4				col 4				
col 4		col 8										
col 6						col 6						
col 12												

12 Columns

6 Columns

6 Columns

4 Columns

4 Columns

4 Columns



# KLASY SIATKI GRID

Siatka jest responsywna i automatycznie zmienia się wraz z rozmiarem ekranu – kolumny się zwężają lub wchodzą pod siebie w zależności od potrzeb i ustawień.

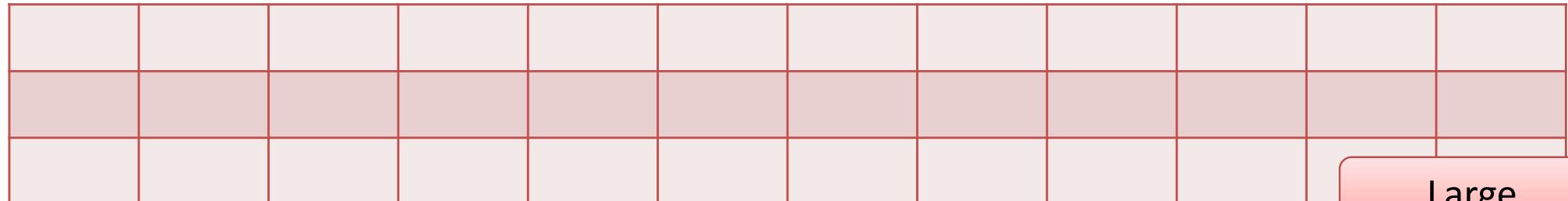
System grid operuje na ~~5~~ klasach:

- xs (dla telefonów)
- sm (dla tabletów)
- md (dla urządzeń desktopowych)
- lg (dla dużych rozdzielcości)
- xl (dla bardzo dużych rozdzielcości)

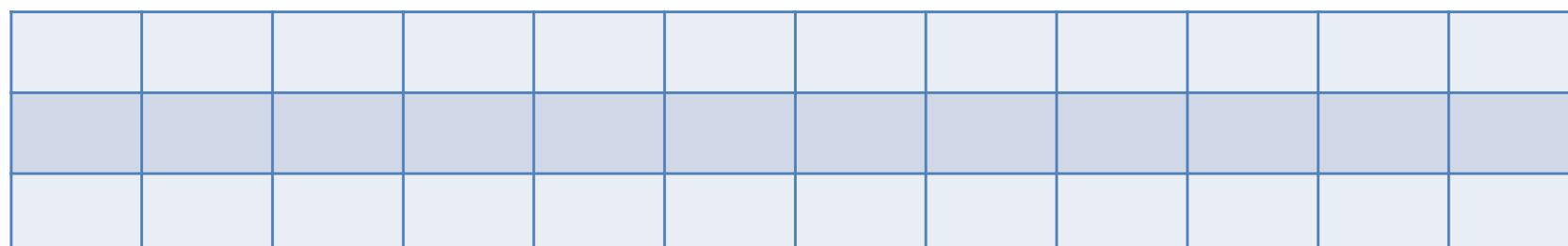


	Extra small $<576px$	Small $\geq 576px$	Medium $\geq 768px$	Large $\geq 992px$	Extra large $\geq 1200px$
maks szer. kon.	brak (auto)	540px	720px	960px	1140px
prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-

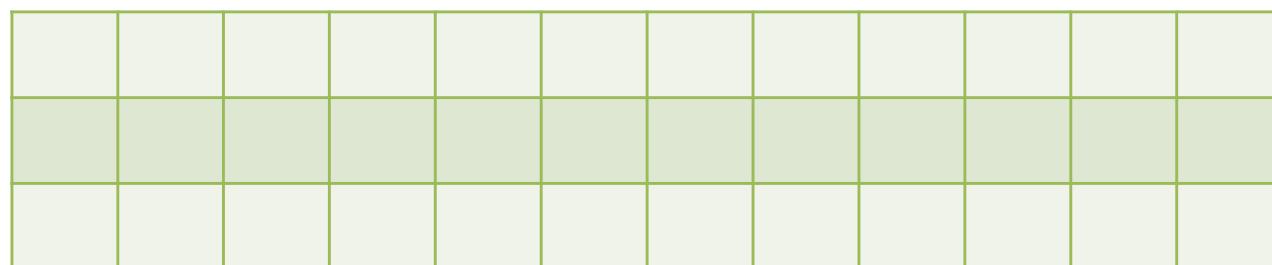
# Grids w Bootstrap



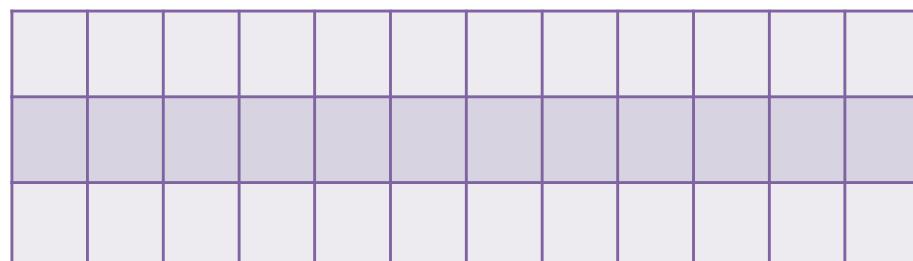
Large



Medium



Small



Extra Small

# Koncepcja Grid

- Wiersz & kolumna
- Wiersz
  - Horyzontalny kontener (12 column)
- Columny sa nazwane
  - col-<grid size>-<content size>
  - Przykład:
    - col-lg-8: large, używa 8 kolumn
    - col-sm-4: small, używa 4 kolumn
- Możliwa jest kombinacja różnych klas

```
<div class="row">
 <div class="col-lg-4">
 </div>
</div>
```

```
<div class="row">
 <div class="col-md-6">
 </div>
</div>
```

```
<div class="row">
 <div class="col-sm-6">
 </div>
</div>
```

```
<div class="row">
 <div class="col-xs-6">
 </div>
</div>
```

class='col-md-12'

class='col-md-6'

class='col-md-6'

class='col-md-4'

class='col-md-4'

class='col-md-4'

# Tworzenie prostej siatki

## Jak to zrobić?

# CONTAINER I CONTAINER-FLUID

Bootstrap wymaga elementu container, który otacza zawartość strony.

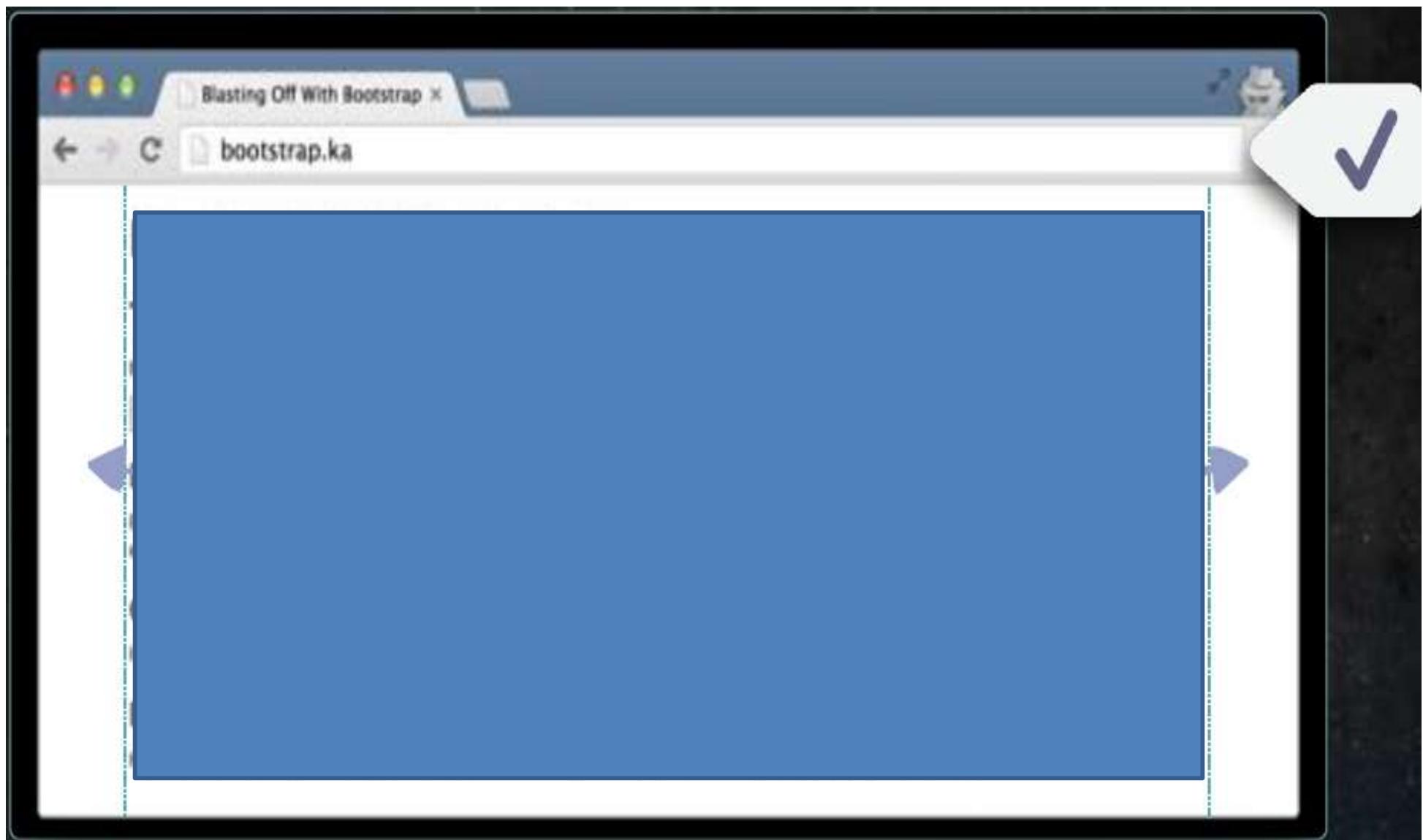
Element ten służy do określania, jakiej szerokości będzie strona.

- **.container** - strona będzie konkretnej szerokości zależnej od urządzenia;
- **.container-fluid** – strona będzie wypełniała całą szerokość ekranu urządzenia.

**Ważne:** Elementów o class="container" nie można zagnieździć.



# Container



# Rząd w kontenerze

```
<div class="container">
 <div class="row">
 </div>
</div>
```

Tworzymy rząd – warstwę zawierającą kolumny `<div class="row">`.

W rzędzie tworzymy kolumny – suma numerów w kolumnach powinna wynosić maksimum 12 `<div class="col-sm-8">`.

```
<div class="row">
 <div class="col-*-*"></div>
</div>
<div class="row">
 <div class="col-*-*"></div>
 <div class="col-*-*"></div>
 <div class="col-*-*"></div>
</div>
```

# Bootstrap Grid

```
<div class="container">
```

```
<div class="row">
```

```
<div class="container">
```

```
<div class="row">
```

1

2

3

4

5

6

7

8

9

10

11

12

```
<div class="container">
```

```
 <div class="row">
```

```
 <div class="col-sm-5">
```

1

2

3

4

5

6

7

8

9

10

11

12

```
<div class="container">
```

```
 <div class="row">
```

```
 <div class="col-sm-5">
```

1

2

3

4

5

```
 <div class="col-sm-7">
```

6

7

8

9

10

11

12

## PRZYKŁAD – 3 KOLUMNY

```
<div class="row">
 <div class="col-sm-4">col 4</div>
 <div class="col-sm-4">col 4</div>
 <div class="col-sm-4">col 4</div>
</div>
```



col 4                    col 4                    col 4

## PRZYKŁAD – 2 RÓŻNE KOLUMNY

Dwie kolumny dla desktopów i po jednej dla tabletów.

```
<div class="row">
 <div class="col-sm-12 col-md-4">col 4</div>
 <div class="col-sm-12 col-md-8">col 8</div>
</div>
```

Dekstop:



col 4                    col 8

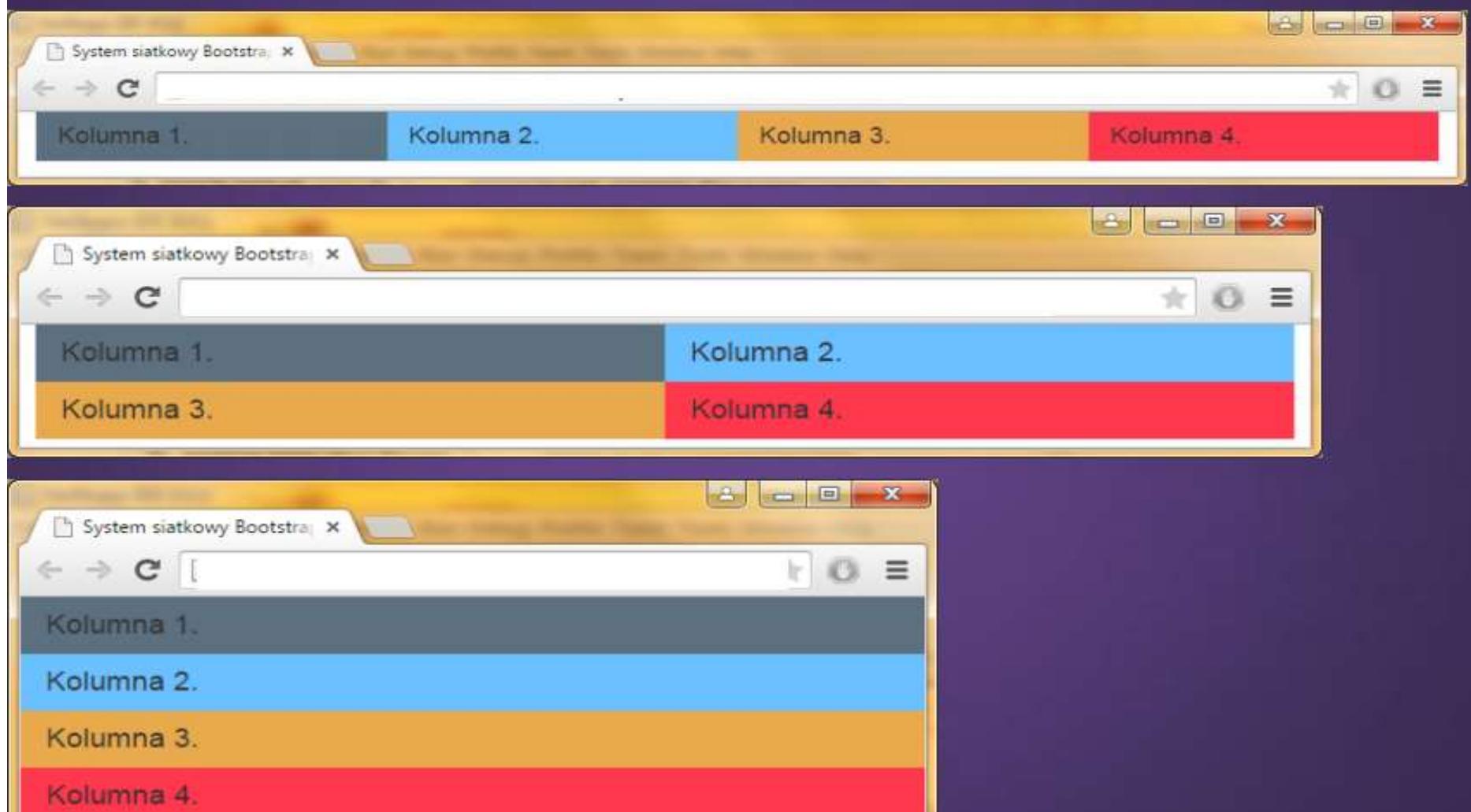
Tablet:



col 4

col 8

```
<div class="col-xs-12 col-sm-6
col-md-3">
```



# Siatka zagnieżdzona

```
<div class="container">
```

```
 <div class="row">
```

```
 <div class="col-sm-5">
```

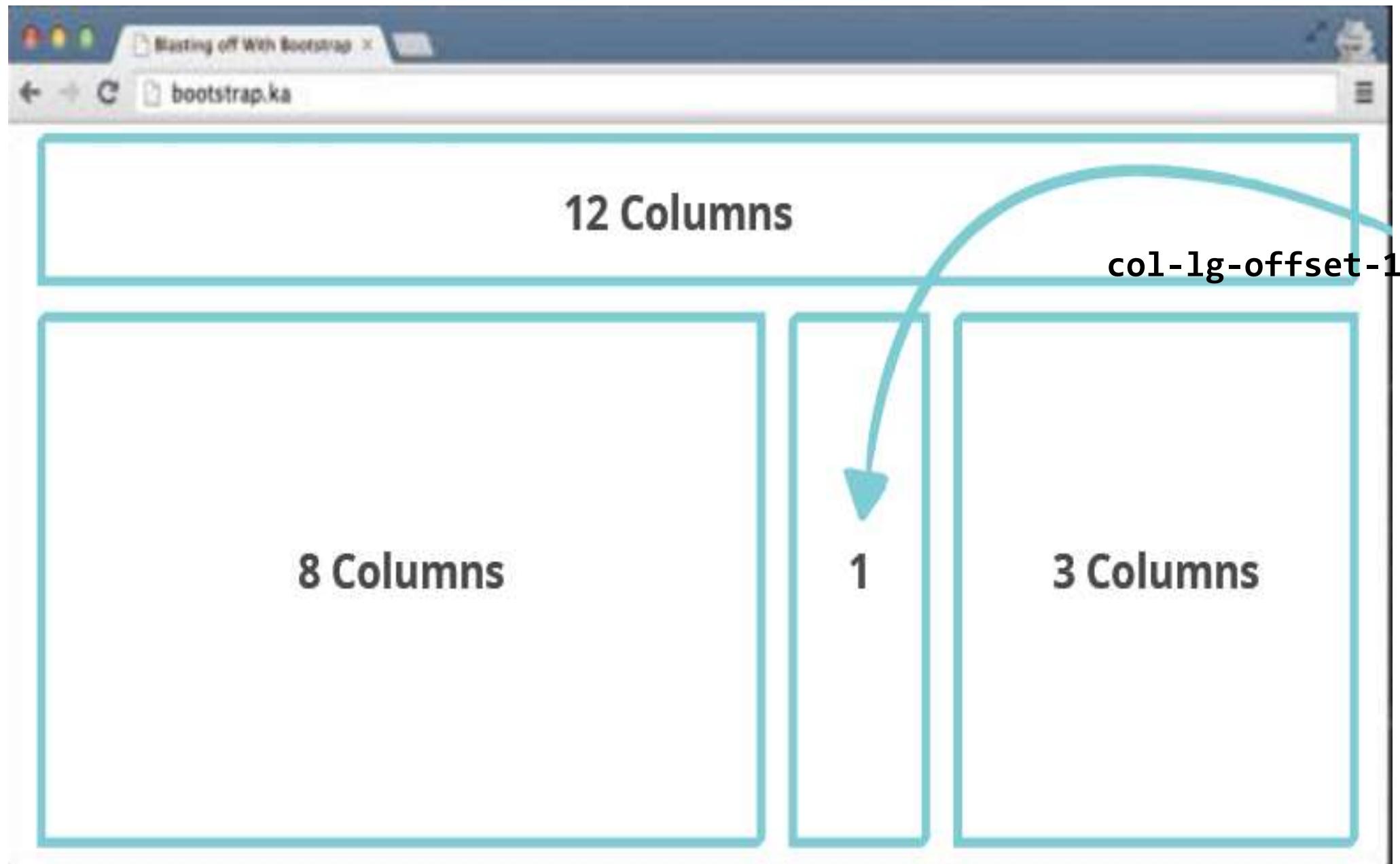
1 2 3 4 5

```
 <div class="col-sm-7">
```

```
 <div class="row">
```

```
 <div class="col-sm-9">
```

```
 <div
 class="col-
 sm-3">
```



# Ukrywanie zawartości

## Blasting Off With Bootstrap

### The Fastest Way to Space

Make your way to space in the comfort of your own rocket, elevator or transporter.

[Take the Tour](#) [Book Tickets Now](#)



### Book Today!

Even if you're traveling tomorrow, you can still get tickets today. We have a number of conveniently located ports around the globe to service everyone.

### Go Anywhere

If you need to get to space today, why not try out a transporter? Despite the claims, there are have been no deaths in the last 6 weeks!

### RocketBus®

For cheapest fares, catch the next RocketBus® to the stars. Cheaper on your wallet, and easiest way to make friends.

## Blasting off With Bootstrap

### The Fastest Way to Space

Make your way to space in the comfort of your own rocket, elevator or transporter.

[Take the Tour](#) [Book Tickets Now](#)

### Book Today!

Even if you're traveling tomorrow, you can still get tickets today. We have a number of conveniently located ports around the globe to service everyone.

### Go Anywhere

If you need to get to space today, why not try out a transporter? Despite the claims, there are have been no deaths in the last 6 weeks!

### RocketBus®

For cheapest fares, catch the next RocketBus® to the stars. Cheaper on your wallet, and easiest way to make friends.



```
<div class = „d-md-none .d-lg-block“>

</div>
```

# Ukrywanie zawartości

Rozmiar ekranu

## Ukryte cały czas

Ukryte tylko na xs

Ukryte tylko na sm

Ukryte tylko na md

Ukryte tylko na lg

Ukryte tylko na xl

## Widoczne on all

Widoczne tylko na xs

Widoczne tylko na sm

Widoczne tylko na md

Widoczne tylko na lg

Widoczne tylko na xl

Klasa

## .d-none

.d-none .d-sm-block

.d-sm-none .d-md-block

.d-md-none .d-lg-block

.d-lg-none .d-xl-block

.d-xl-none

## .d-block

.d-block .d-sm-none

.d-none .d-sm-block .d-md-none

.d-none .d-md-block .d-lg-none

.d-none .d-lg-block .d-xl-none

.d-none .d-xl-block

# Dlaczego Bootstrap?

## powód #1 mnogość funkcji

Designed for everyone, everywhere.

Need reasons to love Bootstrap? Look no further.

### Built for and by nerds

Like you, we love building awesome products on the web. We love it so much, we decided to help people just like us do it easier, better, and faster. Bootstrap is built for you.

### 12-column grid

Grid systems aren't everything, but having a durable and flexible one at the core of your work can make development much simpler. Use our built-in grid classes or roll your own.

### Growing library

Despite being only 10kb (gzipped), Bootstrap is one of the most complete front-end toolkits out there with dozens of fully functional components ready to be put to use.

### HTML5

Built to support new HTML5 elements and syntax.

### CSS3

Progressively enhanced components for ultimate style.

### For all skill levels

Bootstrap is designed to help people of all skill levels—designer or developer, huge nerd or early beginner. Use it as a complete kit or use to start something more complex.

### Responsive design

With Bootstrap 2, we've gone fully responsive. Our components are scaled according to a range of resolutions and devices to provide a consistent experience, no matter what.

### Custom jQuery plugins

What good is an awesome design component without easy-to-use, proper, and extensible interactions? With Bootstrap, you get custom-built jQuery plugins to bring your projects to life.

### Open-source

Built for and maintained by the community via [GitHub](#).

### Cross-everything

Originally built with only modern browsers in mind, Bootstrap has evolved to include support for all major browsers (even IE7!) and, with Bootstrap 2, tablets and smartphones, too.

### Styleguide docs

Unlike other front-end toolkits, Bootstrap was designed first and foremost as a styleguide to document not only our features, but best practices and living, coded examples.

### Built on LESS

Where vanilla CSS falters, LESS excels. Variables, nesting, operations, and mixins in LESS makes coding CSS faster and more efficient with minimal overhead.

### Made at Twitter

Brought to you by an experienced [engineer](#) and [designer](#).

# Dlaczego

## powód #2: popularność

The screenshot shows a browser window displaying the GitHub 'Popular Forked' page at <https://github.com/popular/forked>. The page features a navigation bar with links for Explore, Gist, Blog, Help, and a user profile for kegill. Below the navigation is a search bar and a tab bar with 'Explore' (selected), 'Repositories' (highlighted in orange), 'Languages', and 'Timeline'. Underneath is a sub-tab bar with 'Popular Forked' (selected) and 'Popular Starred'. The main content area displays four repository cards:

- bootstrap**: JavaScript, 48,206 stars, 14,386 forks. Description: Sleek, intuitive, and powerful front-end framework for faster and easier web development. Last updated 15 hours ago.
- Spoon-Knife**: This repo is for demonstration purposes only. Comments and issues may or may not be responded to. Last updated a year ago.
- homebrew**: Ruby, 12,090 stars, 5,856 forks. Description: The missing package manager for OS X. Last updated 4 hours ago.
- rails**: Ruby, 18,082 stars, 5,424 forks. Description: Ruby on Rails. Last updated 2 hours ago.

# Dlaczego

powód #3: Wsparcie dla wszystkich? przeglądarki

Also MSIE and Opera



# Dlaczego

## powód #4: Grid System



- 960 Grid System <http://960.gs/>
- Blue Print CSS <http://www.blueprintcss.org/>
- Golden Grid System <http://goldengridsystem.com/>

# Dlaczego

## powód #5: Components

- Buttons:



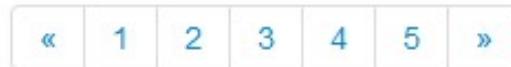
- Tabs:



- Breadcrumb:

Home / Library / Data

- Pagination:



- Alerts:

Warning! Best check yo self, you're not looking too good.

x

- Progress bar:



# Dlaczego

## powód #6: Javascript/jQuery

Transitions (required for any animation)

Modals

Dropdowns

Scrollspy

Toggable tabs

Tooltips

Popovers (requires Tooltips)

Affix

Alert messages

Buttons

Collapse

Carousel

Typeahead

# Dlaczego

## Powód #7: Custom (kustomizacja)

### Scaffolding

- Normalize and reset
- Body type and links
- Grid system
- Layouts

### Components

- Button groups and dropdowns
- Navs, tabs, and pills
- Navbar
- Breadcrumbs
- Pagination
- Pager
- Thumbnails
- Alerts
- Progress bars
- Hero unit
- Media component

### Base CSS

- Headings, body, etc
- Code and pre
- Labels and badges
- Tables
- Forms
- Buttons

### Miscellaneous

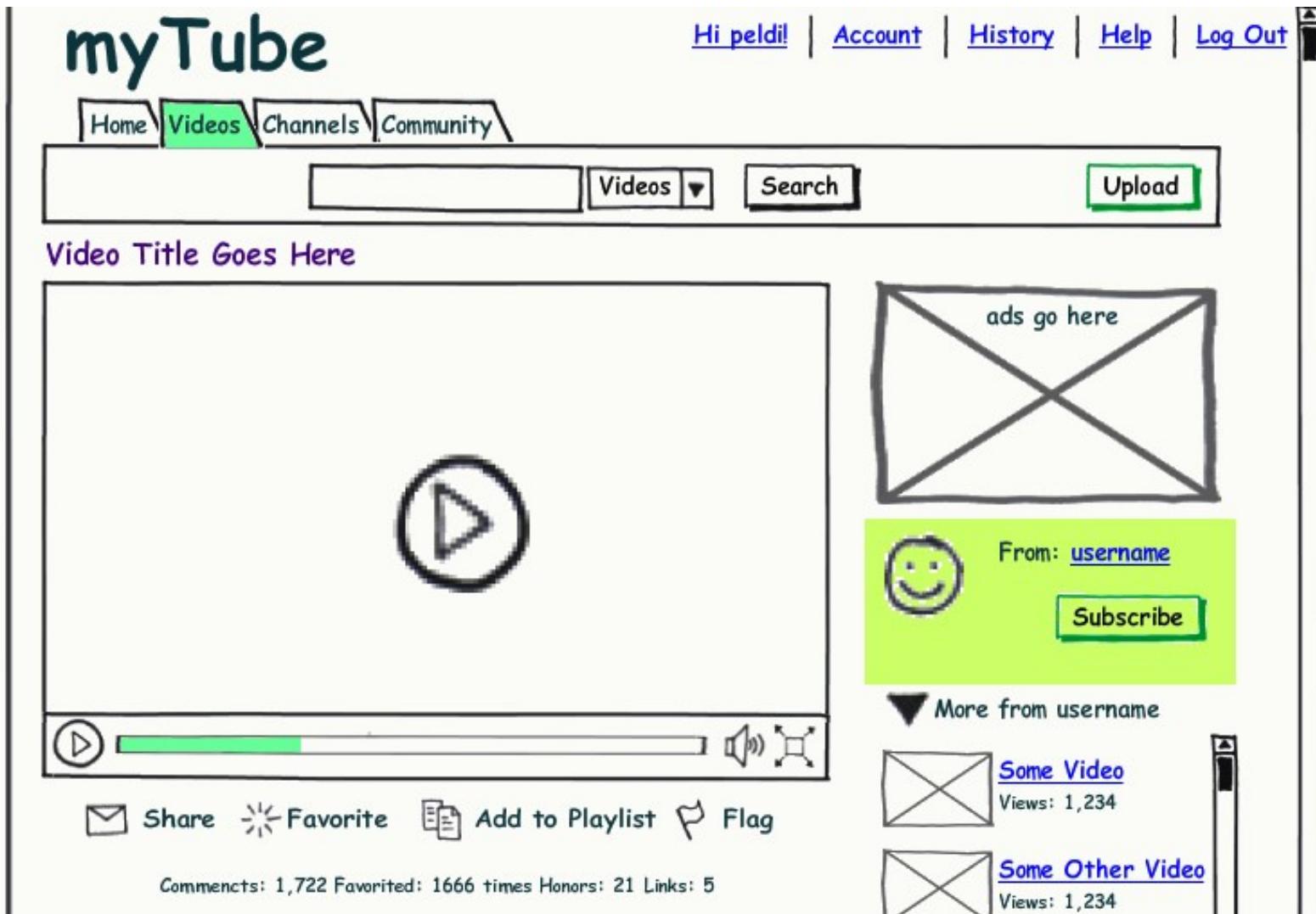
- Media object
- Wells
- Close icon
- Utilities
- Component animations

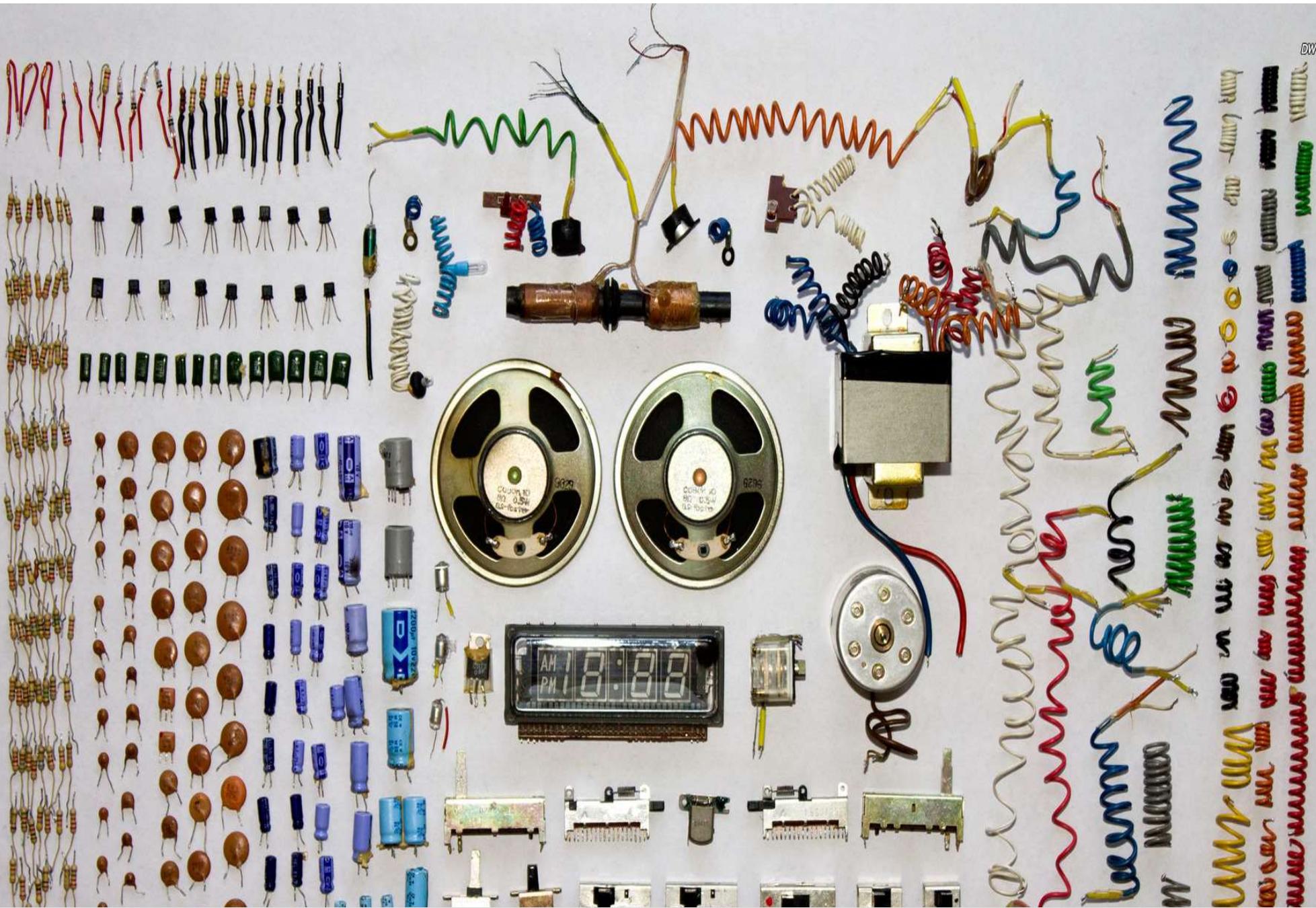
### Responsive

- Visible/hidden classes
- Narrow tablets and below (<767px)
- Tablets to desktops (767-979px)
- Large desktops (>1200px)
- Responsive navbar

# Dlaczego

## powód #8: Żywą Mock-Ups





# KOMPONENTY BOOTSTRAP

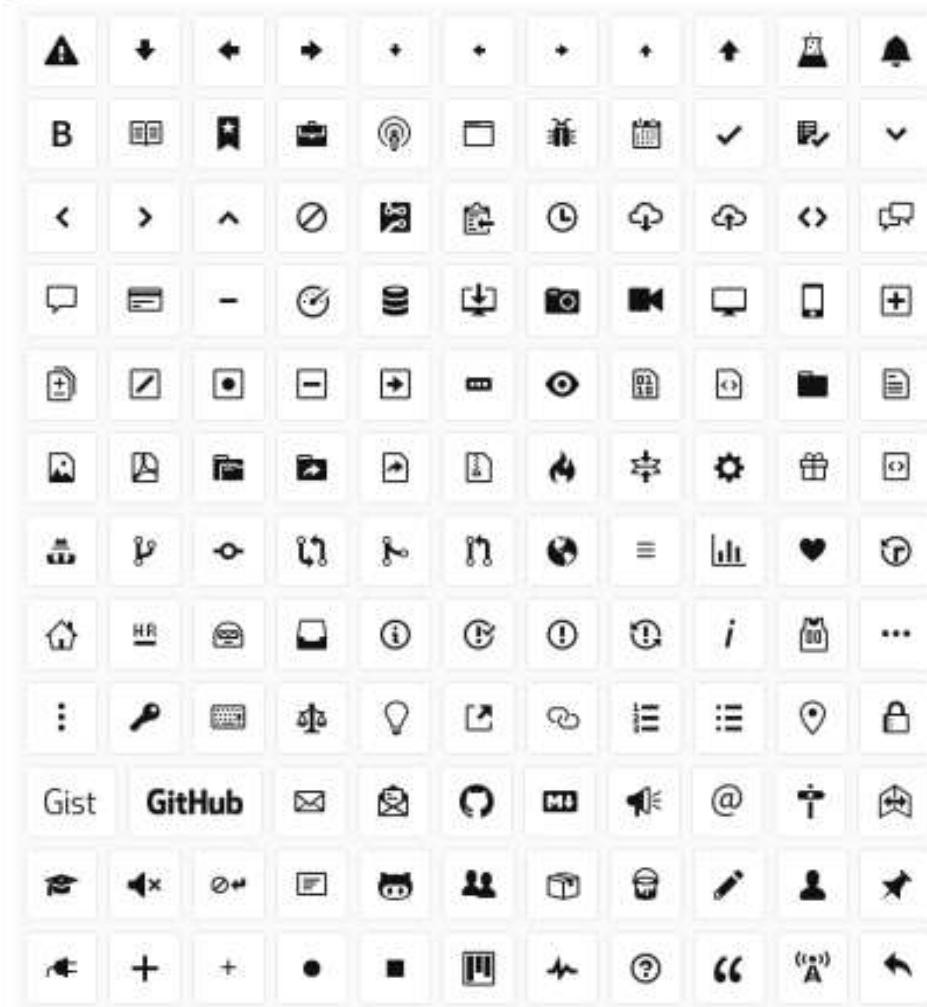
Lista komponentów z dokumentacji:

- powiadomienia, informacje ostrzegawcze (doc. *Alerts*),
- etykietki towarzyszące (plakietki, znaczki, symbole) (doc. *Badge*),
- okruszki, nawigacja okruszkowa (z Jasia i Małgorzatą, doc. *Breadcrumb*),
- przyciski (doc. *Buttons*),
- grupy przycisków (doc. *Button group*),
- karty (doc. *Cards*),
- **karuzela**, ze zdjęciami, slajdami tekstu (doc. *Carousel*),
- treść pokazywana/ukrywana, rozwijana zawartość (doc. *Collapse*),
- **listy rozwijane**, najczęściej listy odnośników (doc. *Dropdowns*),
- **kontrolki formularzy**, dużo ich... (doc. *Forms*),
- grupy kontrolek formularza (doc. *Input group*),

Lista komponentów z dokumentacji:

- **wielki baner**, telebim (doc. *Jumbotron*),
- **grupa elementów tworząca listę** (doc. *List group*),
- okna dialogowe (doc. *Modal*),
- **elementy nawigacyjne** (doc. *Navs*),
- **belka nawigacyjna**, nagłówek strony z nawigacją (doc. *Navbar*),
- paginacja, numeracja kolejnych stron (doc. *Pagination*),
- okienka wyskakujące (pop-up) po kliknięciu (doc. *Popovers*),
- paski postępu (doc. *Progress*),
- szpiedzy przewijania, komponenty automatycznej aktualizacji linków nawigacyjnych zależnie przewijanej treści (doc. *Scrollspy*),
- malutkie okienka wyskakujące (pop-up) po najechaniu na element (doc. *Tooltips*).

- Sam w sobie nie zawiera żadnego zestawu ikon, ale rekomenduje kilka z nich:
  - Open Ionic - <https://useiconic.com/open/>
  - Octicons - <https://octicons.github.com/>
- Preferowane są ikony w formacie SVG.



# POZOSTAŁE

- Okna modal
- Tooltips
- Popovers
- Bages
- Alerts
- Images
- Breadcrumbs
- Pagination
- I wiele innych

The screenshot displays a user interface design system with the following components:

- Popover top:** A callout bubble pointing upwards containing placeholder text: "Sed posuere consectetur est at lobortis. Aenean eu leo quam. Pellentesque ornare sem lacinia quam venenatis vestibulum."
- Table:** A grid of 8 columns and 5 rows labeled 4 through 8. Each cell contains placeholder text: "Column content".
- Image:** Three placeholder images: a square (140x140), a circle (140x140), and another square (140x140).
- Modals:** A large modal window titled "Small modal" with a close button (X) is centered over the grid.
- Tooltips:** Two tooltits are shown: "Tooltip on right" and "Tooltip on right" (bolded), both pointing to the right.
- Alerts:** Five horizontal alert bars:
  - Green: "Well done! You successfully read this important alert message."
  - Light blue: "Heads up! This alert needs your attention, but it's not super important."
  - Yellow: "Warning! Better check yourself, you're not looking too good."
  - Pink: "Oh snap! Change a few things up and try submitting again."

Nazwa

Podgląd

Button groups

Left Middle Right

Button  
dropdowns



Navigational  
tabs

Home Profile Messages

Navbar

Title Home Link Link

Labels

Success

Warning

# PRZYCISKI - TYPY

Bootstrap udostępnia 7 rodzajów przycisków:



```
<button type="button" class="btn btn-default">Default</button>
<button type="button" class="btn btn-primary">Primary</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-link">Link</button>
```

## PRZYCISKI – TYPY

Każdy z tych typów może (nie musi) otrzymać jeden ze stanów:

- .active – przycisk wygląda jak wciśnięty;
- .disabled – przycisk przestaje być klikalny.



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

```
<button type="button" class="btn btn-default active">
Button</button>
```

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

```
<button type="button" class="btn btn-default disabled">
Button</button>
```

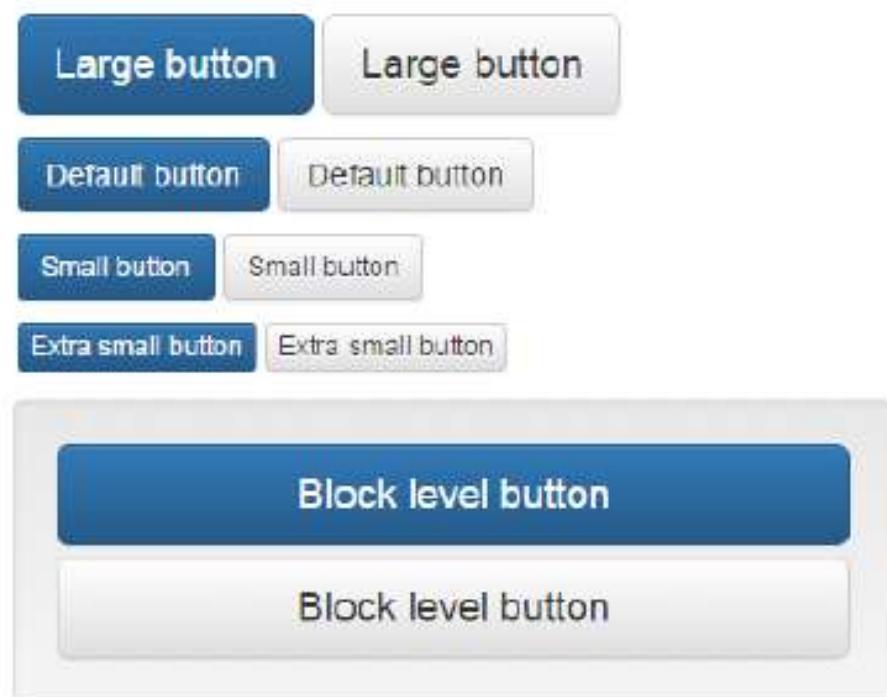
## PRZYCISKI - ROZMIARY

Ponadto można korzystać z kilku różnych rozmiarów przycisków.

- .btn-lg
- .btn-md
- .btn-sm
- .btn-xs
- .btn-block

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

```
<button type="button" class="btn btn-primary btn-lg btn-
block">Block level button</button>
```



# Karty

320 x 180

**Karta (card)** - elastyczny i rozszerzalny kontener zawartości.

## Tytuł karty

Treść naszej karty, czyli jakiś opis czy cokolwiek chcemy w niej zamieścić.

Button z linkiem

```
1 <div class="card">
2
3 <div class="card-block">
4 <h4 class="card-title">Tytuł karty</h4>
5 <p class="card-text">Treść naszej karty, czyli jakiś opis czy cokolwiek chcemy w niej zamieścić.</p>
6 Button z linkiem
7 </div>
8 </div>
```

# Pasek nawigacji – komponent navbar

Tworzymy element div i przypisujemy mu dwie klasy: *navbar* i *navbar-default*.

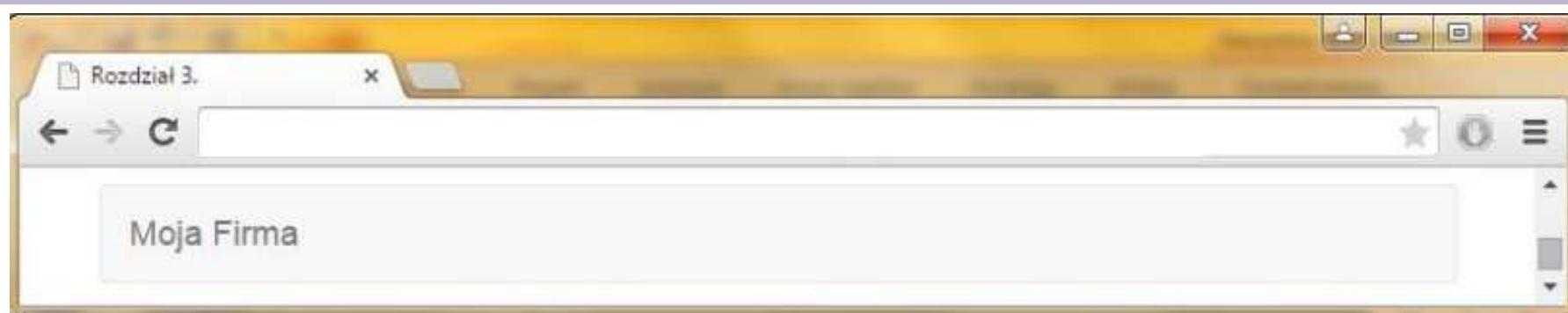
Wewnątrz elementu klasy navbar tworzymy kolejny div klasy **container-fluid**.

```
<div class="navbar navbar-default">
 <div class="container-fluid">
 </div>
</div>
```

Wstawiamy div klasy **navbar-header**

```
<div class="navbar-header">
 <button type="button" class="navbar-
 toggle" data-toggle="collapse" data-
 target="#my navbar-content">

 </button>
 <a class="navbar-brand"
 href="#">Moja Firma
</div>
```



## NAWIGACJA – DODATKOWE KLASY

- .navbar-inverse
- .navbar-fixed-top
- .navbar-fixed-bottom
- .navbar-right

```
<button type='button' class='navbar-toggle'>
 Toggle navigation
 <i class='glyphicon glyphicon-align-justify'></i>
</button>
```

```
<button type='button' class='navbar-toggle'>
 Toggle navigation

</button>
```

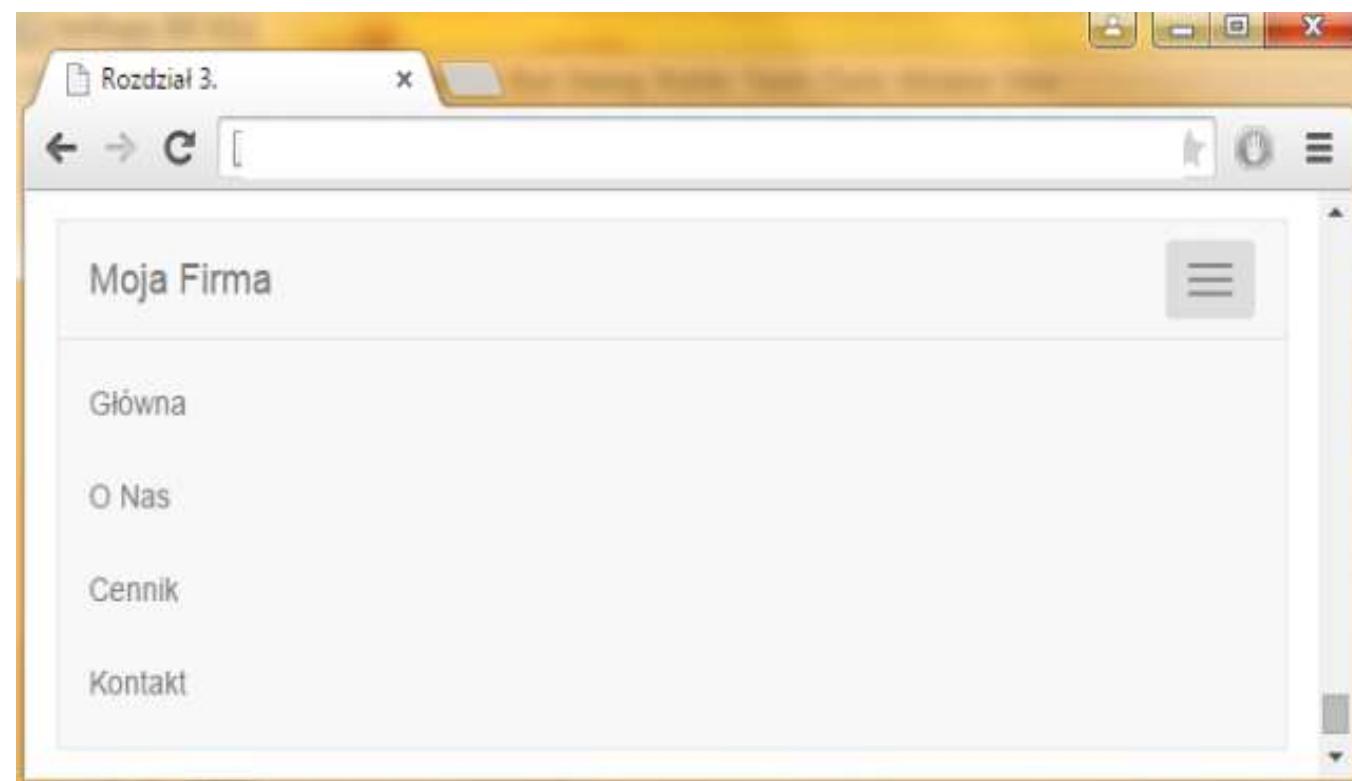
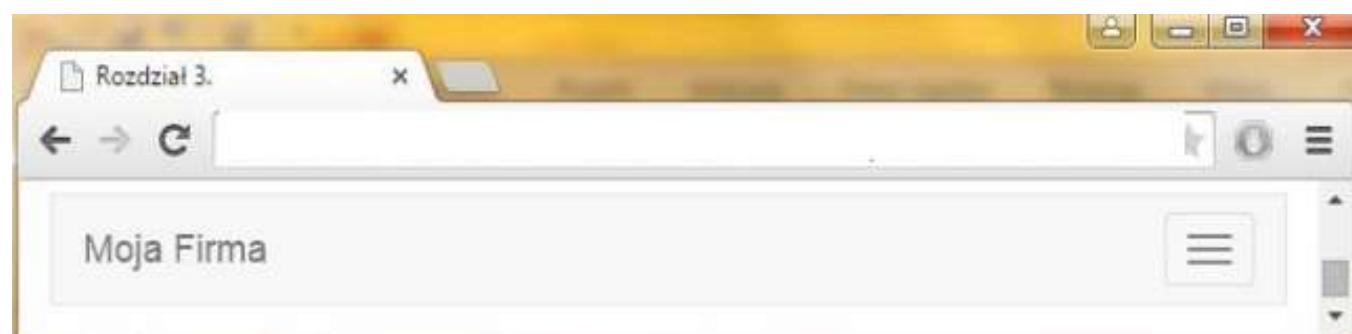
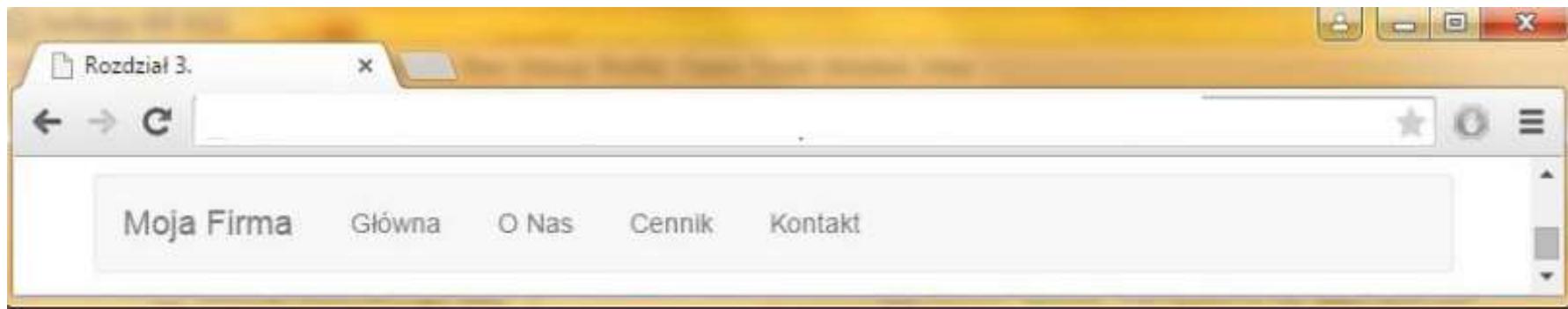
Tworzymy kolejny div, który jest rodzeństwem elementu navbar-header – temu elementowi przypisujemy dwie klasy: **collapse** i **navbar-collapse**.

```
<div class="collapse navbar-collapse" id="mynavbar-content">
</div>
```

```

 Główna
 O Nas
 Cennik
 Kontakt

```



# Obramowanie - borders



```



```



```



```

# FORMULARZE

Bootstrap udostępnia trzy rodzaje formularzy:

- Formularz wertykalny (domyślnie)
- Formularz liniowy
- Formularz horyzontalny

Wszystkie trzy rodzaje korzystają z następujących znaczników.

- `<form role="form">`
- `<div class="form-group">` - otacza etykietę i kontrolkę formularza
- `.form-control` – klasa dodana do `<input>`, `<textarea>` oraz `<select>`

# PRZYKŁAD – FORMULARZ WERTYKALNY

Email address

Email

Password

Password

File input

Nie wybrano pliku.

Example block-level help text here.

Check me out

# PRZYKŁAD – FORMULARZ LINIOWY

<form class="form-inline">

Elementy ułożone do lewej strony.

Name

Jane Doe

Email

jane.doe@example.com

## PRZYKŁAD – FORMULARZ HORYZONTALNY

```
<form class="form-horizontal">
```

Elementy ułożone z zastosowaniem systemu siatki Bootstrapa.

.form-group zachowuje się jak .row.

Email

Email

Password

Password

Remember me

Sign in

# Dropdowns

```
<div class="dropdown">
 <button class="btn btn-default dropdown-toggle" type="button"
id="dropdownMenu1"
 data-toggle="dropdown">
 Select a value

 </button>
 <ul class="dropdown-menu" role="menu">
 <li role="presentation"><a role="menuitem" tabindex="-1"
href="#">1
 <li role="presentation"><a role="menuitem" tabindex="-1"
href="#">2
 ...

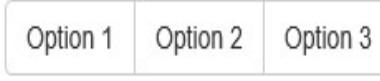
</div>
```

Dropdown



# Button groups

```
<div class="btn-group" role="group">
 <button type="button" class="btn btn-
default">Option 1</button>
 <button type="button" class="btn btn-
default">Option 2</button>
 <button type="button" class="btn btn-
default">Option 3</button>
</div>
```



The code block shows the HTML structure for creating a button group. It consists of a `<div>` element with the class `btn-group` and the role `group`. Inside this div, there are three `<button>` elements, each with the class `btn` and `btn-default`. The buttons are labeled "Option 1", "Option 2", and "Option 3". The resulting visual representation is a horizontal row of three light blue-outlined buttons with white backgrounds and black text.

# Nav

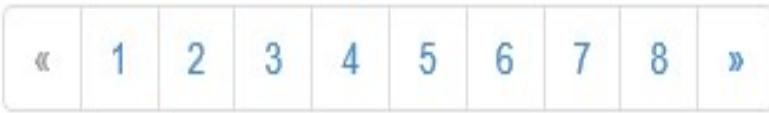
```
<ul class="nav nav-tabs">
 <li role="presentation" class="active">Home
 <li role="presentation">Profile
 <li role="presentation">Messages

```



Home    Profile    Messages

# Pagination



```
<nav>
 <ul class="pagination">
 <li class="disabled">

 «

 1
 2
 ...

 »

 </nav>
```