



## 1. SIATKA BOOTSTRAP

### Podstawowa budowa layoutu strony

Całość elementów layoutu zawiera się w `div` z klasą `.container`.  
Możliwe jest stosowanie do 12 kolumn (klasy `.col-*-*`) umieszczonych w rzędach z klasą `.row`.

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

### „Płynny” layout strony

Dla layoutów o szerokości równej szerokości okna przeglądarki stosuje się klasę `.container-fluid`.

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

### System siatki — tabela

	Extra small Smartfony < 768 px	Small Tablety ≥ 768 px	Medium Urządzenia desktopowe ≥ 992 px	Large Urządzenia desktopowe ≥ 1200 px
Maksymalna szerokość elementu z klasą <code>.container</code>	auto	750px	970px	1170px
Prefiks klas kolumn	<code>.col-xs-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>
Sufiks klas kolumn	Każda klasa kolumn Bootstrap może mieć końcówkę o wartości: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 Wartość ta decyduje o szerokości kolumny			
Szerokość kolumn (przy 12 kolumnach w rzędzie)	Auto	~62px	~81px	~97px
Margines wewnętrzny kolumny (padding)	Łącznie 30 px, 15 px na stronę (prawa, lewa)			

### Przykład stosowania klas kolumn

```
<div class="col-xs-6 col-sm-12 col-md-4 col-lg-3">...</div>
```

Na ekranach < 768 px kolumna (`.col-xs-6`) ma szerokość równą 50% szerokości layoutu.  
Na ekranach ≥ 768 px kolumna (`.col-sm-12`) ma szerokość równą 100% szerokości layoutu.  
Na ekranach ≥ 768 px kolumna (`.col-md-4`) ma szerokość równą ~33% szerokości layoutu.  
Na ekranach ≥ 768 px kolumna (`.col-lg-4`) ma szerokość równą 25% szerokości layoutu.  
Poniższy przykład pokazuje budowę dwukolumnowej siatki, w której szerokość kolumn wynosi 50% szerokości layoutu dla wszystkich przedziałów rozdzielczości ≥ 768 px.  
Dla rozdzielczości < 992 px (`.col-sm-i.col-xs-`) kolumny będą miały szerokość 100% szerokości layoutu. Jest to zachowanie standardowe, występujące w momencie braku określenia zachowania kolumn w niższych rozdzielczościach. Kolumny w przedziale ≥ 1200 px, mimo że ich szerokość nie została określona klasą `.col-lg-`, pozostaną w szerokości polowy layoutu strony.

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

### Określanie widoczności elementów siatki

Dzięki posługiwaniu się klasami Bootstrap można określać, w jakich przedziałach rozdzielczości dany element ma być widoczny lub ukryty. Przedział określa się tak jak w przypadku rozmiaru kolumn.

#### Klasa `.visible-*-*`

Klasa `.visible-*-*` służy do określenia, w jakich przedziałach rozdzielczości obiekt będzie widoczny, jednocześnie pozostając ukrytym w pozostałych.

Klasa	Opis
<code>.visible-xs-*</code>	Element widoczny tylko na wyświetlaczach o szerokości < 768 px.
<code>.visible-sm-*</code>	Element widoczny tylko na wyświetlaczach o szerokości ≥ 768 px i < 992 px.
<code>.visible-md-*</code>	Element widoczny tylko na wyświetlaczach o szerokości ≥ 992 px i < 1200 px.
<code>.visible-lg-*</code>	Element widoczny tylko na wyświetlaczach o szerokości ≥ 1200 px.

Klasy `.visible-*-*` stosuje się wraz z właściwościami CSS `display` o wartościach `inline`, `block`, `inline-block`.

- `.visible-*-inline`
- `.visible-*-block`
- `.visible-*-inline-block`

#### Klasa `.hidden-*`

Klasa `.hidden-*` służy do określenia, w jakich przedziałach rozdzielczości obiekt będzie ukryty, jednocześnie pozostając widocznym w pozostałych.

Klasa	Opis
<code>.hidden-xs</code>	Element ukryty tylko na wyświetlaczach o szerokości < 768 px.
<code>.hidden-sm</code>	Element ukryty tylko na wyświetlaczach o szerokości ≥ 768 px i < 992 px.
<code>.hidden-md</code>	Element ukryty tylko na wyświetlaczach o szerokości ≥ 992 px i < 1200 px.
<code>.hidden-lg</code>	Element ukryty tylko na wyświetlaczach o szerokości ≥ 1200 px.

Możliwe jest stosowanie jednocześnie kilku klas określających widoczność elementu:

```
<div class="visible-sm-block visible-md-block">...</div>
<div class="hidden-xs hidden-lg">...</div>
```

### Określanie elementów widocznych na wydrukach

Stosując klasy `.visible-print-*` i `.hidden-print`, można określić widoczność elementów na wydruku.

Klasa	Opis
<code>.visible-print-block</code>	Elementy blokowe nie są wyświetlane w przeglądarce, ale są widoczne na wydruku.
<code>.visible-print-inline</code>	Elementy liniowe nie są wyświetlane w przeglądarce, ale są widoczne na wydruku.
<code>.visible-print-inline-block</code>	Elementy liniowo-blokowe nie są wyświetlane w przeglądarce, ale są widoczne na wydruku.
<code>.hidden-print</code>	Elementy wyświetlają się w przeglądarce, ale nie są widoczne na wydruku.

### Czyszczenie float

Obiekty z klasami kolumn (`.col-*-*`) mają własność CSS `float: left`, co w niektórych przypadkach powoduje nieodpowiednie wyświetlanie się tych elementów. Problem ten rozwiązuje się, stosując obiekt z klasą `.clearfix`, poprzez umieszczenie go po serii kolumn, których suma szerokości w danym przedziale rozdzielczości wynosi 12. Klasę `.clearfix` stosuje się w parze z klasą `visible-*-block`.

```
<div class="container">
  <div class="row">
    <div class="col-sm-6 col-md-4">...</div>
    <div class="col-sm-6 col-md-4">...</div>
    <!-- suma szerokości w przedziale sm to 6+6=12 -->
    <div class="clearfix visible-sm-block"></div>
    <div class="col-sm-6 col-md-4">...</div>
    <!-- suma szerokości w przedziale md to 4+4+4=12 -->
    <div class="clearfix visible-md-block"></div>
    <div class="col-sm-6 col-md-4">...</div>
    <!-- suma szerokości w przedziale sm to 6+6=12 -->
    <div class="clearfix visible-sm-block"></div>
    <!-- suma szerokości w przedziale md to 4+4+4=12 -->
    <div class="clearfix visible-md-block"></div>
    <div class="col-sm-6">...</div>
    <div class="col-sm-6">...</div>
    <!-- stosowanie czyszczenia nie jest konieczne z powodu zamknięcia obiektu z klasą .row -->
  </div>
</div>
```

### Wcięcie kolumn

Z wykorzystaniem klasy `.col-*-offset` można określić wcięcie/margines z lewej strony kolumny (przesunąć kolumnę w prawo), o wartości równej dowolnej wielokrotności szerokości kolumny.

Kolumna o szerokości 6 z wcięciem o szerokości 3:

```
<div class="row">
  <div class="col-md-6 col-md-offset-3">...</div>
</div>
```

W efekcie powstaje kolumna wyśrodkowana względem strony:

3+6+3 (pozostały obszar z prawej strony kolumny) = 12.

### Zmiana kolejności wyświetlania kolumn

Korzystając z pary klas `.col-*-pull-*` i `.col-*-push-*`, można zmienić kolejność wyświetlania sąsiadujących ze sobą kolumn dla określonego przedziału rozdzielczości.

W klasach tych należy określić szerokość podobnie jak w przypadku klas kolumn (szerokość kolumn). Żeby sąsiadujące ze sobą kolumny mogły zamienić się miejscami, klasom `.col-*-pull-*` i `.col-*-push-*` określa się wartość równą wartości klasy kolumny (szerokości kolumny) sąsiedniej.

W poniższym przykładzie kolumny zamieniają się miejscami w przedziale md.

```
<div class="row">
  <div class="col-md-8 col-md-push-4">...</div>
  <div class="col-md-4 col-md-pull-8">...</div>
</div>
```

### Zagnieżdżanie kolumn

Kolumny można umieszczać (zagnieżdżać) w innych kolumnach. Kolumny znajdujące się wewnątrz innej muszą być umieszczone w elemencie z klasą `.row`.