

# Bootstrap Grids

Bootstrap to biblioteka CSS, rozwijana przez programistów Twittera, wydawana na licencji MIT. Zawiera zestaw przydatnych narzędzi ułatwiających tworzenie interfejsu graficznego stron oraz aplikacji internetowych. Bootstrap bazuje głównie na gotowych rozwiązaniach HTML oraz CSS i może być stosowany m.in. do stylizacji elementów jak teksty, formularze, przyciski, wykresy, nawigacje i innych komponentów wyświetlanych na stronie. Biblioteka korzysta także z języka JavaScript. Jednym z głównych narzędzi biblioteki Bootstrap jest siatka (Grid system) pomagająca układać elementy i dostosowywać ich wygląd na różnych urządzeniach.

## 1. Utworzenie projektu

1. Utwórz na pulpicie folder o nazwie `Bootstrap Grids`.
2. Uruchom program `Visual Studio Code` i otwórz w nim folder `Bootstrap Grids`.
3. W folderze `Bootstrap Grids` utwórz plik o nazwie `index.html`.
4. W pliku `index.html` poprzez polecenie `!` wygeneruj strukturę dokumentu HTML.
5. Zmień tytuł dokumentu HTML na `Bootstrap Grids`.
6. Kod HTML po utworzeniu struktury dokumentu HTML i zmianie tytułu:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bootstrap Grids</title>
  </head>
  <body>
  </body>
</html>
```

## 2. Dodanie plików CSS i JS Bootstrapa do projektu

Pliki CSS i JS Bootstrapa można dołączyć z serwera CDN lub pobrać do lokalnego użytku, jeżeli zakładamy pracę offline.

Typy skompilowanych plików CSS:

- `bootstrap.css`, `bootstrap.min.css`, `bootstrap.rtl.css`, `bootstrap.rtl.min.css` - wszystkie dostępne narzędzia CSS
- `bootstrap-grid.css`, `bootstrap-grid.rtl.css`, `bootstrap-grid.min.css`, `bootstrap-grid.rtl.min.css` - tylko siatka Bootstrap i narzędzia flex
- `bootstrap-utilities.css`, `bootstrap-utilities.rtl.css`, `bootstrap-utilities.min.css`, `bootstrap-utilities.rtl.min.css` - formatowanie treści bez siatki Bootstrap, reboot i komponentów

- `bootstrap-reboot.css`, `bootstrap-reboot.rtl.css`, `bootstrap-reboot.min.css`, `bootstrap-reboot.rtl.min.css` - tylko reboot dla ujednolicenia wyglądu w przeglądarkach

Typy skompilowanych plików JS:

- `bootstrap.bundle.js`, `bootstrap.bundle.min.js` - Bootstrap razem z biblioteką Popper
- `bootstrap.js`, `bootstrap.min.js` - Bootstrap bez biblioteki Popper

Dodatkowe oznaczenia w nazwach plików CSS i JS:

- `min` - wersja zminifikowana, służąca do publikacji w Internecie
- `rtl` - wsparcie dla języków pisanych od prawej do lewej (right to left), jak hebrajski i arabski

## 2.1. Dołączanie plików Bootstrapa z serwera CDN

1. Dołącz do pliku `index.html`:

- plik `https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css` na końcu sekcji `<head>` poprzez polecenie `link:css`
- plik `https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js` na końcu sekcji `<body>` poprzez polecenie `script:src`

2. Kod HTML po dołączeniu plików CSS i JS z serwera CDN:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bootstrap Grids</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">
  </head>
  <body>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
"></script>
  </body>
</html>
```

## 2.2. Pobieranie i dołączanie plików Bootstrapa lokalnie

1. Pobierz skompilowane pliki CSS i JS Bootstrapa:

<https://github.com/twbs/bootstrap/releases/download/v5.3.3/bootstrap-5.3.3-dist.zip>

2. Rozpakuj plik zip.

3. Przenieś foldery `css` i `js` do folderu `Bootstrap Grids`.

4. Dołącz do pliku `index.html`:

- plik `css/bootstrap.min.css` na końcu sekcji `<head>` poprzez polecenie `link:css`
- plik `js/bootstrap.bundle.min.js` na końcu sekcji `<body>` poprzez polecenie `script:src`

5. Kod HTML po dołączeniu plików CSS i JS lokalnie:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bootstrap Grids</title>
    <link rel="stylesheet" href="css/bootstrap.min.css">
  </head>
  <body>
    <script src="js/bootstrap.bundle.min.js"></script>
  </body>
</html>
```

### 3. Punkty przerwania (breakpoints)

Bootstrap 5 zawiera sześć domyślnych punktów przerwania, czasami określanych jako warstwy siatki, do budowania responsywnego wyglądu stron i aplikacji internetowych.

Breakpoint	Infiks klasy	Wymiary
Extra small	<i>brak</i>	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥992px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px

### 4. Kontenery

Kontenery są najbardziej podstawowym elementem szablonu strony w Bootstrap i **są wymagane podczas korzystania z domyślnego systemu siatki Bootstrap**. Kontenery służą do przechowywania, uzupełniania i (czasami) centrowania zawartości w nich zawartej.

Bootstrap zawiera trzy różne typy kontenerów:

- `.container` - ustawia maksymalną szerokość w każdym responsywnym punkcie przerwania
- `.container-{breakpoint}` - szerokość 100% do określonego punktu przerwania
- `.container-fluid` - szerokość 100% we wszystkich punktach przerwania

1. Na końcu sekcji `<head>` dodaj kod formatujący wygląd nagłówków oraz kontenerów i kolumn Bootstrapa:

```
<style>
  h1, h2, h3, h4 {
    text-align: center;
  }

  [class*="container"] {
    background: gray;
    border: 1px solid black;
    text-align: center;
  }

  [class*="col"] {
    background: lightgray;
    border: 1px solid black;
  }

  .h100 {
    height: 100px;
    background: white;
    border: 1px solid black;
  }
</style>
```

2. Na początku sekcji `<body>` dodaj:

- nagłówek pierwszego stopnia o treści `Bootstrap Grids`
- nagłówek drugiego stopnia o treści `Kontenery Bootstrap`

3. Pod nagłówkami poprzez polecenie `.container*7{container}` wygeneruj 7 kontenerów zawierających klasę `.container` i treść `container`.

4. Uzupełnij nazwy klas i treść w kolejnych kontenerach zgodnie z poniższą tabelą.

Breakpointy a klasy kontenerów	Extra small <576px	Small ≥576px sm	Medium ≥768px md	Large ≥992px lg	X-Large ≥1200px xl	XX-Large ≥1400px xxl
<code>.container</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-sm</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-md</code>	100%	100%	720px	960px	1140px	1320px
<code>.container-lg</code>	100%	100%	100%	960px	1140px	1320px
<code>.container-xl</code>	100%	100%	100%	100%	1140px	1320px
<code>.container-xxl</code>	100%	100%	100%	100%	100%	1320px
<code>.container-fluid</code>	100%	100%	100%	100%	100%	100%

5. Kod po dodaniu nazw klas i treści kontenerów:

```
<div class="container">container</div>
<div class="container-sm">container-sm</div>
<div class="container-md">container-md</div>
<div class="container-lg">container-lg</div>
<div class="container-xl">container-xl</div>
<div class="container-xxl">container-xxl</div>
<div class="container-fluid">container-fluid</div>
```

6. W programie `Visual Studio Code` korzystając z wtyczki `Live Server` (Go Live) otwórz plik `index.html` w przeglądarce `Chrome`.

7. W przeglądarce `Chrome` poprzez klawisz `F12` uruchom narzędzia developerskie.

8. Rozszerz przeglądarkę na szerokość ok. 1600px a następnie ją zwężaj i obserwuj jak zachowują się poszczególne kontenery.

## 5. Siatka Bootstrap (Grid system)

Siatki Bootstrap używamy dzieląc stronę w pionie na rzędy (ang. rows), a rzędy na kolumny (ang. columns). Szerokość każdej kolumny określana jest automatycznie lub poprzez liczbę od 1 do 12 a suma szerokości kolumn w rzędzie powinna sumować się do 12.

### 5.1. Kolumny o równej szerokości

1. Na końcu sekcji `<body>` przed linią dołączającą plik JS dodaj:

- nagłówek drugiego stopnia o treści `Siatka Bootstrap`
- nagłówek trzeciego stopnia o treści `Kolumny o równej szerokości`

2. Pod nagłówkiem `Kolumny o równej szerokości`:

- utwórz podstawowy kontener Bootstrap poprzez polecenie `.container`
- w kontenerze poprzez polecenie `.row>.col*3{kolumna $ z 3}` utwórz rząd zawierający 3 kolumny o treści `kolumna x z 3`
- w kontenerze poprzez polecenie `.row>.col*2{kolumna $ z 2}` utwórz rząd zawierający 2 kolumny o treści `kolumna x z 2`

3. Kod HTML po utworzeniu kontenera z rzędami i kolumnami:

```
<h2>Siatka Bootstrap</h2>
<h3>Kolumny o równej szerokości</h3>
<div class="container">
  <div class="row">
    <div class="col">kolumna 1 z 3</div>
    <div class="col">kolumna 2 z 3</div>
    <div class="col">kolumna 3 z 3</div>
  </div>
  <div class="row">
    <div class="col">kolumna 1 z 2</div>
    <div class="col">kolumna 2 z 2</div>
  </div>
</div>
```

## 5.2. Kolumny o ustalonej szerokości

Szerokość kolumn w siatce Bootstrap można ustawiać za pomocą cyfr (od 1 do 12) dodanej do nazwy klasy `col`, np. `col-1`, `col-2`, `col-3`, `col-4`, `col-5`, `col-6`, `col-7`, `col-8`, `col-9`, `col-10`, `col-11`, `col-12`

1. Na końcu sekcji `<body>` przed linią dołączającą plik JS dodaj nagłówek trzeciego stopnia o treści `kolumny o ustalonej szerokości`.
2. Pod nagłówkiem `kolumny o ustalonej szerokości`:
  - utwórz poprzez polecenie `.container>.row*6>.col-{col-}*2` kontener z 6 rzędami zawierającymi po 2 kolumny
  - pogrupuj kolumny w pary sumujące się do 12: 1 i 11, 2 i 10, 3 i 9, 4 i 8, 5 i 7, 6 i 6
  - uzupełnij nazwy klas i treść kolumn o cyfry
3. Kod HTML po utworzeniu kontenera z rzędami i kolumnami:

```
<h3>kolumny o ustalonej szerokości</h3>
<div class="container">
  <div class="row">
    <div class="col-1">col-1</div>
    <div class="col-11">col-11</div>
  </div>
  <div class="row">
    <div class="col-2">col-2</div>
    <div class="col-10">col-10</div>
  </div>
  <div class="row">
    <div class="col-3">col-3</div>
    <div class="col-9">col-9</div>
  </div>
  <div class="row">
    <div class="col-4">col-4</div>
    <div class="col-8">col-8</div>
  </div>
  <div class="row">
    <div class="col-5">col-5</div>
    <div class="col-7">col-7</div>
  </div>
  <div class="row">
    <div class="col-6">col-6</div>
    <div class="col-6">col-6</div>
  </div>
</div>
```

## 5.3. Kolumny o szerokości uzależnionej od treści

1. Na końcu sekcji `<body>` przed linią dołączającą plik JS dodaj nagłówek trzeciego stopnia o treści `kolumny o szerokości uzależnionej od treści`.
2. Pod nagłówkiem `kolumny o szerokości uzależnionej od treści`:
  - utwórz poprzez polecenie `.container` kontener

- w kontenerze wygeneruj poprzez polecenie `.row*2>.col{$ z 3}*3` 2 rzędy z 3 kolumnami
  - uzupełnij kolumny o klasy i treść zgodnie z poniższym wzorem
3. Kod HTML po utworzeniu kontenera z rzędami i kolumnami oraz uzupełnieniu klas i treści kolumn:

```
<div class="container">
  <div class="row">
    <div class="col col-lg-2">1 z 3</div>
    <div class="col col-md-auto">Treść o zmiennej szerokości</div>
    <div class="col col-lg-2">3 z 3</div>
  </div>
  <div class="row">
    <div class="col lg">1 z 3</div>
    <div class="col col-md-auto">Treść o zmiennej szerokości</div>
    <div class="col col-lg-2">3 z 3</div>
  </div>
</div>
```

## 5.4. Kolumny o responsywnej szerokości

Każdy z punktów przerwania ma swój własny kontener, unikalny przedrostek klasy i modyfikator. Oto jak zmienia się siatka w tych punktach przerwania:

	<b>xs</b> ≤576px	<b>sm</b> ≥576px	<b>md</b> ≥768px	<b>lg</b> ≥992px	<b>xl</b> ≥1200px	<b>xxl</b> ≥1400px
<b>max-width kontenera</b>	100%	540px	720px	960px	1140px	1320px
<b>Prefiks klasy</b>	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>	<code>.col-xxl-</code>
<b>ilość z 12 kolumn</b>	12					

1. Na końcu sekcji `<body>` przed linią dołączającą plik JS dodaj nagłówek trzeciego stopnia o treści `kolumny o responsywnej szerokości`.
2. Pod nagłówkiem `kolumny o responsywnej szerokości`:
  - utwórz poprzez polecenie `.container` kontener
  - w kontenerze wygeneruj poprzez polecenie `.row>.col-sm-{col-sm-}*2` 1 rząd z 2 kolumnami
  - uzupełnij w kolumnach nazwy klas i treść dzieląc je w proporcji 4 do 8
  - w kontenerze wygeneruj poprzez polecenie `.row>.col-sm{col-sm}*3` 1 rząd z 3 kolumnami
  - uzupełnij kolumny o klasy i treść zgodnie z poniższym wzorem

3. Kod HTML po utworzeniu kontenera z rzędami i kolumnami oraz uzupełnieniu klas i treści kolumn:

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

## 5.5. Łączenie różnych technik

1. Na końcu sekcji `<body>` przed linią dołączającą plik JS dodaj nagłówek trzeciego stopnia o treści `Łączenie różnych technik`.
2. Pod nagłówkiem `Łączenie różnych technik`:
  - utwórz poprzez polecenie `.container` kontener
  - w kontenerze wygeneruj poprzez polecenie `.row>.col{col}` 1 rząd z 1 kolumną
  - w kontenerze wygeneruj poprzez polecenie `.row>.col-sm-6.col-md-4.col-lg-3.col-xl-2.col-xxl-1{$ z 12}*12` 12 kolumn z klasami zmieniającymi szerokość kolumn w każdym punkcie przerwania
  - w kontenerze wygeneruj poprzez polecenie `.row>.col{col}` 1 rząd z 1 kolumną
3. Kod HTML po utworzeniu kontenera z rzędami i kolumnami:

```
<div class="container">
  <div class="row">
    <div class="col">col</div>
  </div>
  <div class="row">
    <div class="col-sm-6 col-md-4 col-lg-3 col-xl-2 col-xxl-1">1 z 12</div>
    <div class="col-sm-6 col-md-4 col-lg-3 col-xl-2 col-xxl-1">2 z 12</div>
    <div class="col-sm-6 col-md-4 col-lg-3 col-xl-2 col-xxl-1">3 z 12</div>
    <div class="col-sm-6 col-md-4 col-lg-3 col-xl-2 col-xxl-1">4 z 12</div>
    <div class="col-sm-6 col-md-4 col-lg-3 col-xl-2 col-xxl-1">5 z 12</div>
    <div class="col-sm-6 col-md-4 col-lg-3 col-xl-2 col-xxl-1">6 z 12</div>
    <div class="col-sm-6 col-md-4 col-lg-3 col-xl-2 col-xxl-1">7 z 12</div>
    <div class="col-sm-6 col-md-4 col-lg-3 col-xl-2 col-xxl-1">8 z 12</div>
    <div class="col-sm-6 col-md-4 col-lg-3 col-xl-2 col-xxl-1">9 z 12</div>
    <div class="col-sm-6 col-md-4 col-lg-3 col-xl-2 col-xxl-1">10 z 12</div>
    <div class="col-sm-6 col-md-4 col-lg-3 col-xl-2 col-xxl-1">11 z 12</div>
    <div class="col-sm-6 col-md-4 col-lg-3 col-xl-2 col-xxl-1">12 z 12</div>
  </div>
  <div class="row">
    <div class="col">col</div>
  </div>
</div>
```



## 5.6 Kolumny wierszy (Row columns)

Responsywne klasy `.row-cols-*` umożliwiają szybkie ustawienie liczby kolumn, które najlepiej wyświetlają zawartość na stronie. Normalne klasy `.col-*` odnoszą się do poszczególnych kolumn (np. `.col-md-4`). Klasy kolumn wierszy są ustawiane w nadrzędnym elemencie `.row` jako skrót.

1. Na końcu sekcji `<body>` przed linią dołączającą plik JS dodaj nagłówek trzeciego stopnia o treści `kolumny wierszy`.
2. Pod nagłówkiem `kolumny wierszy`:
  - utwórz poprzez polecenie `.container` kontener
  - w kontenerze wygeneruj poprzez polecenie `.row.row-cols-auto>.col{col}*4` 1 rząd z 4 kolumnami
  - zmień nazwę klasy wiersza `row-cols-auto` na inne z ilością kolumn, np. `row-cols-2` i sprawdź w przeglądarce, jak zmienia się układ strony
3. Kod HTML po utworzeniu kontenera z rzędami i kolumnami:

```
<div class="container">
  <div class="row row-cols-auto">
    <div class="col">col</div>
    <div class="col">col</div>
    <div class="col">col</div>
    <div class="col">col</div>
  </div>
</div>
```

## 5.7. Zagnieżdżanie elementów w siatce Bootstrap

Można zagnieżdżać treść z domyślną siatką. Należy dodać nowy wiersz `.row` i zestaw kolumn `.col-sm-*` w istniejącej kolumnie `.col-sm-*`. Zagnieżdżone wiersze powinny zawierać zestaw kolumn, których suma nie przekracza 12 (nie jest wymagane użycie wszystkich 12 dostępnych kolumn).

1. Na końcu sekcji `<body>` przed linią dołączającą plik JS dodaj nagłówek trzeciego stopnia o treści `Zagnieżdżanie elementów`.
2. Pod nagłówkiem `Zagnieżdżanie elementów`:
  - utwórz poprzez polecenie `.container` kontener
  - w kontenerze wygeneruj poprzez polecenie `.row>.col-sm-3{Poziom 1}*2` 1 rząd z 2 kolumnami
  - zmień klasę drugiej kolumny na `.col-sm-9` i usuń jej treść
  - wewnątrz drugiej kolumny z klasą `.col-sm-9` wygeneruj poprzez polecenie `.row>.col-sm-6{Poziom 2}*2` 1 rząd z 2 kolumnami

3. Kod HTML po utworzeniu kontenera z rzędami i kolumnami:

```
<div class="container">
  <div class="row">
    <div class="col-sm-3">Poziom 1</div>
    <div class="col-sm-9">
      <div class="row">
        <div class="col-sm-6">Poziom 2</div>
        <div class="col-sm-6">Poziom 2</div>
      </div>
    </div>
  </div>
</div>
```

## 6. Modyfikowanie kolumn

Kolumny opierają się na architekturze siatki flexbox. Flexbox oznacza, że mamy opcje zmiany poszczególnych kolumn oraz modyfikacji grup kolumn na poziomie wierszy. Podczas tworzenia układów siatki cała zawartość jest umieszczana w kolumnach. Hierarchia siatki Bootstrap przechodzi od kontenera, przez wiersz i kolumny, do treści. Bootstrap zawiera predefiniowane klasy do tworzenia szybkich, responsywnych układów. Do dyspozycji posiadamy sześć punktów przerwania i tuzin kolumn na każdym poziomie siatki, co daje nam dziesiątki gotowych klas, dzięki którym możemy tworzyć pożądane układy treści na stronie.

### 6.1. Wyrównanie treści

#### 6.1.1. Wyrównanie treści w pionie

##### 6.1.1.1. Wyrównanie wiersza w pionie

1. Na końcu sekcji `<body>` przed linią dołączającą plik JS dodaj nagłówek trzeciego stopnia o treści `wyrównanie treści w pionie`.
2. Pod nagłówkiem `wyrównanie treści w pionie`:
  - o utwórz nagłówek czwartego stopnia o treści `wyrównanie wiersza w pionie`
  - o utwórz poprzez polecenie `.container` kontener
  - o w kontenerze wygeneruj poprzez polecenie `.row.h100*3>.col{kolumna $ z 3}*3` 3 rzędy z 3 kolumnami
  - o w kolejnych rzędach dopisz klasy odpowiedzialne za wyrównanie kolumn `align-items-start`, `align-items-center`, `align-items-end`

3. Kod HTML po utworzeniu kontenera z rzędami i kolumnami:

```
<div class="container">
  <div class="row h100 align-items-start">
    <div class="col">kolumna 1 z 1</div>
    <div class="col">kolumna 2 z 3</div>
    <div class="col">kolumna 3 z 3</div>
  </div>
  <div class="row h100 align-items-center">
    <div class="col">kolumna 1 z 3</div>
    <div class="col">kolumna 2 z 3</div>
    <div class="col">kolumna 3 z 3</div>
  </div>
  <div class="row h100 align-items-end">
    <div class="col">kolumna 1 z 3</div>
    <div class="col">kolumna 2 z 3</div>
    <div class="col">kolumna 3 z 3</div>
  </div>
</div>
```

#### 6.1.1.2. Wyrównanie kolumn w pionie

1. Na końcu sekcji `<body>` przed linią dołączającą plik JS dodaj nagłówek czwartego stopnia o treści `wyrównie kolumn w pionie`.
2. Pod nagłówkiem `wyrównanie kolumn w pionie`:
  - utwórz poprzez polecenie `.container` kontener
  - w kontenerze wygeneruj poprzez polecenie `.row.h100>.col{kolumna $ z 3}*3` 1 rząd z 3 kolumnami
  - w kolejnych kolumnach dopisz klasy odpowiedzialne za wyrównanie `align-self-start`, `align-self-center`, `align-self-end`
3. Kod HTML po utworzeniu kontenera z rzędami i kolumnami:

```
<div class="containter">
  <div class="row h100">
    <div class="col align-self-start">kolumna 1 z 3</div>
    <div class="col align-self-center">kolumna 2 z 3</div>
    <div class="col align-self-end">kolumna 3 z 3</div>
  </div>
</div>
```

#### 6.1.2. Wyrównie treści w poziomie

1. Na końcu sekcji `<body>` przed linią dołączającą plik JS dodaj nagłówek czwartego stopnia o treści `wyrównie treści w poziomie`.
2. Pod nagłówkiem `wyrównie treści w poziomie`:
  - utwórz poprzez polecenie `.container` kontener
  - w kontenerze wygeneruj poprzez polecenie `.row*6>.col-4{kolumna $ z 2}*2` 6 rzędów z 2 kolumnami

- w kolejnych wierszach dopisz klasy odpowiedzialne za wyrównanie `justify-content-start`, `justify-content-center`, `justify-content-end`, `justify-content-around`, `justify-content-between`, `justify-content-evenly`

3. Kod HTML po utworzeniu kontenera z rzędami i kolumnami:

```
<div class="container">
  <div class="row justify-content-start">
    <div class="col-4">kolumna 1 z 2</div>
    <div class="col-4">kolumna 2 z 2</div>
  </div>
  <div class="row justify-content-center">
    <div class="col-4">kolumna 1 z 2</div>
    <div class="col-4">kolumna 2 z 2</div>
  </div>
  <div class="row justify-content-end">
    <div class="col-4">kolumna 1 z 2</div>
    <div class="col-4">kolumna 2 z 2</div>
  </div>
  <div class="row justify-content-around">
    <div class="col-4">kolumna 1 z 2</div>
    <div class="col-4">kolumna 2 z 2</div>
  </div>
  <div class="row justify-content-between">
    <div class="col-4">kolumna 1 z 2</div>
    <div class="col-4">kolumna 2 z 2</div>
  </div>
  <div class="row justify-content-evenly">
    <div class="col-4">kolumna 1 z 2</div>
    <div class="col-4">kolumna 2 z 2</div>
  </div>
</div>
```

## 6.2. Zawijanie kolumn

Jeżeli w jednym rzędzie zostanie umieszczonych więcej niż 12 kolumn, każda grupa dodatkowych kolumn jako jedna jednostka zostanie zawinięta w nowy wiersz.

1. Na końcu sekcji `<body>` przed linią dołączającą plik JS dodaj nagłówek trzeciego stopnia o treści `Zawijanie kolumn`.
2. Pod nagłówkiem `Zawijanie kolumn`:
  - utwórz poprzez polecenie `.container` kontener
  - w kontenerze wygeneruj poprzez polecenie `.row>.col-9{.col-9}*3` 1 rząd z 3 kolumnami
  - zmień klasę i treść drugiej kolumny na `.col-4`
  - zmień klasę i treść trzeciej kolumny na `.col-6`

3. Kod HTML po utworzeniu kontenera z rzędami i kolumnami:

```
<div class="container">
  <div class="row">
    <div class="col-9">.col-9</div>
    <div class="col-4">.col-4</div>
    <div class="col-6">.col-6</div>
  </div>
</div>
```

## 6.3. Podział kolumn

Przełamanie kolumn do nowej linii w flexboksie wymaga małego hacka: należy dodać element o szerokości: 100% wszędzie tam, gdzie chcesz zawinąć kolumny do nowej linii. Zwykle jest to realizowane za pomocą wielu wierszy, ale nie każda metoda implementacji może to uwzględnić.

1. Na końcu sekcji `<body>` przed linią dołączającą plik JS dodaj nagłówek trzeciego stopnia o treści `Podział kolumn`.
2. Pod nagłówkiem `Podział kolumn`:
  - utwórz poprzez polecenie `.container` kontener
  - w kontenerze wygeneruj poprzez polecenie `.row>.col-6.col-sm-3{.col-6 .col-sm-3}*4` 1 rząd z 4 kolumnami
  - w rzędzie pomiędzy 2 a 3 kolumną wstaw dodatkowy `<div class="w-100"></div>`

3. Kod HTML po utworzeniu kontenera z rzędami i kolumnami:

```
<div class="container">
  <div class="row">
    <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>
    <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>
    <div class="w-100"></div>
    <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>
    <div class="col-6 col-sm-3">.col-6 .col-sm-3</div>
  </div>
</div>
```

## 6.4. Zmiana kolejności elementów

Do kontrolowania wizualnej kolejności treści elementów należy użyć klas `.order-*`. Klasy te są responsywne, więc można ustawić kolejność według punktu przerwania (np. `.order-1.order-md-2`). Obejmuje obsługę od 1 do 5 na wszystkich sześciu poziomach siatki.

1. Na końcu sekcji `<body>` przed linią dołączającą plik JS dodaj nagłówek trzeciego stopnia o treści `Zmiana kolejności elementów`.
2. Pod nagłówkiem `Zmiana kolejności elementów`:
  - utwórz poprzez polecenie `.container` kontener
  - w kontenerze wygeneruj poprzez polecenie `.row>.col*3` 1 rząd z 3 kolumnami
  - w kolumnie pierwszej dodaj treść `pierwszy w DOM, bez zmiany kolejności`

- w kolumnie drugiej dodaj klasę `order-5` i treść `drugi w DOM, kolejność zwiększona do 5`
- w kolumnie trzeciej dodaj klasę `order-1` i treść `trzeci w DOM, kolejność zmniejszona do 1`

3. Kod HTML po utworzeniu kontenera z rzędami i kolumnami:

```
<div class="container">
  <div class="row">
    <div class="col">pierwszy w DOM, bez zmiany kolejności</div>
    <div class="col order-5">
      drugi w DOM, kolejność zwiększona do 5
    </div>
    <div class="col order-1">
      trzeci w DOM, kolejność zmniejszona do 1
    </div>
  </div>
</div>
```

4. Na końcu sekcji `<body>` przed linią dołączającą plik JS:

- utwórz poprzez polecenie `.container` kontener
- w kontenerze wygeneruj poprzez polecenie `.row>.col*3` 1 rząd z 3 kolumnami
- w kolumnie pierwszej dodaj klasę `.order-last` i treść `pierwszy w dom, ostatni w kolejności`
- w kolumnie drugiej dodaj treść `drugi w DOM, kolejność nie zmieniona`
- w kolumnie trzeciej dodaj klasę `.order-first` i treść `trzeci w dom, pierwszy w kolejności`

5. Kod HTML po utworzeniu kontenera z rzędami i kolumnami:

```
<div class="container">
  <div class="row">
    <div class="col order-last">
      pierwszy w dom, ostatni w kolejności
    </div>
    <div class="col">drugi w DOM, kolejność nie zmieniona</div>
    <div class="col order-first">
      trzeci w dom, pierwszy w kolejności
    </div>
  </div>
</div>
```

## 6.5. Odsuwanie kolumn

Kolumny siatki Bootstrap można przesunąć na dwa sposoby: responsywne klasy `.offset-grid` lub marginesy. Klasy siatki są dopasowywane do kolumn, podczas gdy marginesy są bardziej przydatne w przypadku szybkich układów, w których szerokość przesunięcia jest zmienna.

Na końcu sekcji `<body>` przed linią dołączającą plik JS dodaj nagłówek trzeciego stopnia o treści `odsuwanie kolumn`.

### 6.5.1. Odsuwanie kolumn przy pomocy klas offset-grid

Kolumny możemy przenieść w prawo, używając klas `.offset-md-*`. Te klasy zwiększają lewy margines kolumny o `*` kolumny. Na przykład `.offset-md-4` przesunę `.col-md-4` w czterech kolumnach.

1. Pod nagłówkiem `Odsuwanie kolumn` dodaj nagłówek czwartego stopnia o treści `Odsuwanie kolumn przy pomocy klas offset-grid`.
2. Pod nagłówkiem `Odsuwanie kolumn przy pomocy klas offset-grid`:
  - o utwórz poprzez polecenie `.container` kontener
  - o w kontenerze wygeneruj poprzez polecenie `.row>.col-md-4{.col-md-4}*2` 1 rząd z 2 kolumnami
  - o w kolumnie drugiej dodaj klasę i treść `.offset-md-4`
  - o w kontenerze wygeneruj poprzez polecenie `.row>.col-md-3.offset-md-3{.col-md-3}.offset-md-3}*2` 1 rząd z 2 kolumnami
  - o w kontenerze wygeneruj poprzez polecenie `.row>.col-md-6.offset-md-3{.col-md-6}.offset-md-3}` 1 rząd z 1 kolumną
3. Kod HTML po utworzeniu kontenera z rzędami i kolumnami:

```
<div class="container">
  <div class="row">
    <div class="col-md-4">.col-md-4</div>
    <div class="col-md-4 offset-md-4">.col-md-4 .offset-md-4</div>
  </div>
  <div class="row">
    <div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>
    <div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>
  </div>
  <div class="row">
    <div class="col-md-6 offset-md-3">.col-md-6 .offset-md-3</div>
  </div>
</div>
```

### 6.5.2. Odsuwanie kolumn przy pomocy marginesów

Wraz z przejściem na flexbox w wersji 4, można użyć klas `.ms-auto` lub `.me-auto` w celu utworzenia marginesów odsuwających sąsiadujące kolumny od siebie.

1. Na końcu sekcji `<body>` przed linią dołączającą plik JS dodaj nagłówek czwartego stopnia o treści `Odsuwanie kolumn przy pomocy marginesów`.
2. Pod nagłówkiem `Odsuwanie kolumn przy pomocy marginesów`:
  - o utwórz poprzez polecenie `.container` kontener
  - o w kontenerze wygeneruj poprzez polecenie `.row>.col-md-4{.col-md-4}*2` 1 rząd z 2 kolumnami
  - o w kolumnie drugiej dodaj klasę i treść `.ms-auto`
  - o w kontenerze wygeneruj poprzez polecenie `.row>.col-md-3.ms-md-auto{.col-md-3}.ms-md-auto}*2` 1 rząd z 2 kolumnami

- w kontenerze wygeneruj poprzez polecenie `.row>.col-auto{.col-auto}*2` 1 rząd z 2 kolumnami
- w kolumnie pierwszej ostatniego rzędu dodaj klasę i treść `.me-auto`

3. Kod HTML po utworzeniu kontenera z rzędami i kolumnami:

```
<div class="container">
  <div class="row">
    <div class="col-md-4">.col-md-4</div>
    <div class="col-md-4 ms-auto">.col-md-4 .ms-auto</div>
  </div>
  <div class="row">
    <div class="col-md-3 ms-md-auto">.col-md-3 .ms-md-auto</div>
    <div class="col-md-3 ms-md-auto">.col-md-3 .ms-md-auto</div>
  </div>
  <div class="row">
    <div class="col-auto me-auto">.col-auto .me-auto</div>
    <div class="col-auto">.col-auto</div>
  </div>
</div>
```

## 6.6. Samodzielne klasy kolumn

Klasy `.col-*` mogą być również używane poza wierszami, aby nadać elementom określoną szerokość. Jeżeli klasy kolumn są używane poza wierszami marginesy są pomijane.

Przykładowe użycie klas poza wierszami:

```
<div class="col-3 bg-light p-3 border">
  .col-3: szerokość 25%
</div>
<div class="col-sm-9 bg-light p-3 border">
  .col-sm-9: szerokość 75% powyżej punktu przerwania sm
</div>
```

Klasy kolumn mogą być używane razem z narzędziami do tworzenia responsywnych pływających obrazów. Pamiętaj, aby owinąć zawartość w opakowanie z klasą `.clearfix`, aby wyczyścić element pływający, jeśli tekst jest krótszy.



Przykładowe zastosowanie klasy `.clearfix`:

```
<div class="clearfix">
  
  <p>
    Akapit tekstu zastępczego. Używamy go tutaj, aby pokazać użycie klasy
    clearfix. Dodajemy tutaj sporo bezsensownych fraz, aby zademonstrować, w jaki
    sposób kolumny wchodzą w interakcję z pływającym obrazem.
  </p>
  <p>
    Jak widać, akapity z wdziękiem owijają się wokół pływającego obrazu. A
    teraz wyobraź sobie, jak by to wyglądało z rzeczywistą treścią, a nie tylko tym
    nudnym tekstem zastępczym, który trwa i trwa, ale w rzeczywistości nie przekazuje
    żadnych namacalnych informacji. Po prostu zajmuje miejsce i tak naprawdę nie
    powinno się go czytać.
  </p>
  <p>
    A jednak nadal wytrwale czytasz ten tekst zastępczy, mając nadzieję na
    więcej spostrzeżeń lub easter egg ukryty treści. Żart, być może. Niestety, tutaj
    tego nie ma.
  </p>
</div>
```

## 6.7. Guttery

Guttery to odstępy między elementami, używane do responsywnego rozmieszczania i wyrównywania treści w systemie siatki Bootstrap. Guttery to odstępy między zawartością kolumn, utworzone przez poziome dopełnienie. Ustawiamy dopełnienie w prawo i dopełnienie w lewo w każdej kolumnie i używamy marginesu ujemnego, aby zrównoważyć to na początku i na końcu każdego wiersza i wyrównać zawartość. Guttery zaczynają się od szerokości 1.5 rem (24 piksele). Dzięki temu możemy dopasować naszą siatkę do skali marginesów wewnętrznych i zewnętrznych. Guttery można szybko regulować. Należy użyć klas gutterów specyficznych dla punktów przerwania, aby zmodyfikować guttery poziome, pionowe lub wszystkie.

## 6.7.1. Guttery poziome

Klasy `.gx-*` mogą być używane do kontrolowania szerokości gutterów poziomych. Może zająć konieczność dostosowania elementu nadrzędnego `.container` lub `.container-fluid` przy użyciu odpowiedniego narzędzia wypełniającego, jeśli używane są również większe guttery, aby uniknąć niepożądanego przepełnienia. W poniższym przykładzie zwiększyliśmy wypełnienie za pomocą `.px-4`:

```
<div class="container px-4">
  <div class="row gx-5">
    <div class="col">
      <div class="p-3 border bg-light">Niestandardowe wypełnienie kolumn</div>
    </div>
    <div class="col">
      <div class="p-3 border bg-light">Niestandardowe wypełnienie
kolumn</div>
    </div>
  </div>
</div>
```

## 6.7.2. Guttery pionowe

Klasy `.gy-*` mogą być używane do kontrolowania szerokości pionowych odstępów między wierszami, gdy kolumny zawierają się do nowych linii. Podobnie jak poziome guttery, pionowe guttery mogą powodować przepełnienie poniżej wiersza na końcu strony. Jeśli tak się stanie, należy dodać kontener wokół `.row` z klasą `.overflow-hidden`:

```
<div class="container overflow-hidden">
  <div class="row gy-5">
    <div class="col-6">
      <div class="p-3 border bg-light">Niestandardowe wypełnienie
kolumn</div>
    </div>
    <div class="col-6">
      <div class="p-3 border bg-light">Niestandardowe wypełnienie
kolumn</div>
    </div>
    <div class="col-6">
      <div class="p-3 border bg-light">Niestandardowe wypełnienie
kolumn</div>
    </div>
    <div class="col-6">
      <div class="p-3 border bg-light">Niestandardowe wypełnienie
kolumn</div>
    </div>
  </div>
</div>
```

### 6.7.3. Połączenie gutterów poziomych i pionowych

Klasy `.g-*` mogą być używane do kontrolowania szerokości gutterów w poziomie, w poniższym przykładzie używamy mniejszej szerokości guttera, więc nie będzie potrzeby dodawania klasy kontenera `.overflow-hidden`:

```
<div class="container">
  <div class="row g-2">
    <div class="col-6">
      <div class="p-3 border bg-light">Niestandardowe wypełnienie
    </div>
    <div class="col-6">
      <div class="p-3 border bg-light">Niestandardowe wypełnienie
    </div>
    <div class="col-6">
      <div class="p-3 border bg-light">Niestandardowe wypełnienie
    </div>
    <div class="col-6">
      <div class="p-3 border bg-light">Niestandardowe wypełnienie
    </div>
  </div>
</div>
```

### 6.7.3. Usuwanie gutterów

Guttery między kolumnami w predefiniowanych klasach siatki można usunąć za pomocą klasy `.g-0`. Usuwa ona ujemne marginesy z `.row` i poziome dopełnienie ze wszystkich bezpośrednich kolumn podrzędnych. Jeżeli potrzebujemy szablonu od krawędzi do krawędzi należy pominąć nadrzędny `.container` lub `.container-fluid` i dodać `.mx-0` do `.row`, aby zapobiec przepełnieniu.

```
<div class="row g-0">
  <div class="col-sm-6 col-md-8">.col-sm-6 .col-md-8</div>
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>
</div>
```

#### 6.7.4. Guttery kolumn wierszy

Klasy gutterów między wierszami można również dodawać do kolumn wierszy. W poniższym przykładzie używamy responsywnych kolumn wierszy i responsywnych klas gutterów:

```
<div class="container">
  <div class="row row-cols-2 row-cols-lg-5 g-2 g-lg-3">
    <div class="col">
      <div class="p-3 border bg-light">kolumna wiersza</div>
    </div>
    <div class="col">
      <div class="p-3 border bg-light">kolumna wiersza</div>
    </div>
    <div class="col">
      <div class="p-3 border bg-light">kolumna wiersza</div>
    </div>
    <div class="col">
      <div class="p-3 border bg-light">kolumna wiersza</div>
    </div>
    <div class="col">
      <div class="p-3 border bg-light">kolumna wiersza</div>
    </div>
  </div>
</div>
```

## 7. Przydatne linki

1. Visual Studio Code: <https://code.visualstudio.com/download>
2. Emmet: <https://docs.emmet.io/cheat-sheet>
3. Bootstrap: <https://getbootstrap.com>
4. Popper: <https://popper.js.org>
5. Material Design for Bootstrap: <https://mdbootstrap.com>