

BOOTSTRAP

Czyli strony internetowe dostosowane do „każdej” rozdzielczości ekranu

W Bootstrapie 5 jest wiele pozytywnych zmian, które przede wszystkim polepszają wydajność strony internetowej. To powoduje szereg zalet zarówno dla programistów jak i użytkowników końcowych.

- rezygnacja z jQuery.
- wprowadzenie dedykowanych ikon SVG
- rezygnacja ze wsparcia Internet Explorer 10
- zmiana środowiska do testów (JUnit na Jasmine)
- Oczywiste (patrz punkt 1.), liczne zmiany w kodzie JavaScript,
- nieznaczne zmiany w CSS,
- odświeżenie kontenerów i wprowadzenie do nich „responsywności”,
- wsparcie dla języków rtl (zmiana słów left / right na start / end)

W ramach ciekawostki można wspomnieć, że **oficjalna strona Bootstrapa zmieniła CMS z Jekyll na Hugo**. Są to systemy CMS do stron statycznych, czyli takich które nie używają standardowej bazy danych SQL w swoim funkcjonowaniu.

Siatka Bootstrapa standardowo składa się **maksymalnie z 12 kolumn**, które są rozdzielone odstępami. Sprawia to, że niezależnie od szerokości użytego ekranu zostanie on podzielony na dwanaście równych części. Takie podzielenie ekranu pozwala tworzyć kolumny, których szerokość może wahać się od 1/12 ekranu do 12/12 (rysunek 1.25).

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-------|---|-------|-------|-------|---|-------|---|-------|-------|-------|----|
| col-2 | | col-2 | | col-2 | | col-2 | | col-2 | | col-2 | |
| col-3 | | | col-3 | | | col-3 | | | col-3 | | |
| col-4 | | | | col-4 | | | | col-4 | | | |
| col-6 | | | | | | col-6 | | | | | |

Użycie klasy `col-2` spowoduje zarezerwowanie przez pojemnik 2 kolumn siatki, co oznacza, że na całej szerokości ekranu takich pojemników może być 6 — każdy z nich zajmuje 1/6 ekranu.

Zastosowanie klasy `col-3` spowoduje użycie przez pojemnik 3 kolumn, tak więc w wierszu mogą zostać umieszczone 4 identyczne pojemniki, z których każdy zajmuje 1/4 ekranu.

To samo dotyczy użycia klasy `col-4` (każdy pojemnik zajmuje 1/3 ekranu) oraz `col-6` (pojemniki zajmują po połowie ekranu).

Ponieważ siatka ma automatycznie reagować na zmianę szerokości ekranu i dostosowywać swój kształt do rozmiaru ekranu urządzenia, na którym jest wyświetlana, wprowadzono progi zmian. Bootstrap definiuje 6 szerokości ekranu, przy których może nastąpić zmiana układu siatki:

- *extra small*, do 575 px — klasa `.col-`;
- *small*, od 576 px do 767 px — klasa `.col-sm-`;
- *medium*, od 768 px do 991 px — klasa `.col-md-`;
- *large*, od 992 px do 1199 px — klasa `.col-lg-`;
- *extra large*, od 1200 px do 1399 px — klasa `.col-xl-`;
- *extra extra large* — od 1400 px — klasa `.col-xxl-`.

Przykładowo klasa `class="col-12 col-sm-6 col-md-4 col-lg-3 col-xl-2"` spowoduje, że pojemnik na urządzeniu o rozmiarze ekranu *extra small* zajmie całą jego powierzchnię, *small* — połowę, *medium* — 1/3, *large* — 1/4, a na największym — 1/6.

Aby uprościć definicję klas, stosuje się skróty. Przykładowo zapis `class="col-md-4"` sprawia, że pojemnik zajmie 1/3 ekranu nie tylko na urządzeniach z wyświetlaczem *medium*, ale także na większych — również tych z wyświetlaczami *large*, *extra large* i *extra extra large*. Zapis klasy `col-12` można pominąć, ponieważ na tego typu ekranie wszystkie bloki siłą rzeczy zostaną ustawione pionowo, jeden pod drugim.

WSKAZÓWKA

Odstępami (tzw. gutterami) pomiędzy blokami/pojemnikami w poszczególnych widokach nie musimy się przejmować — framework sam o nie zadba (ustawi je na domyślną szerokość wynoszącą 30px).

Podłączanie bootstrap do strony:

Mechanizm: Content Delivery Network (CDN)

Globalna sieć serwerów cache może znacząco przyspieszyć wczytywanie się serwisu dla użytkowników którzy są geograficznie znacznie oddaleni od miejsca gdzie znajduje się serwer. Dodatkowo CDN jest w stanie obsłużyć znacznie większy ruch niż pojedynczy serwer. Co w konsekwencji przekłada się na znaczne oszczędności zamiast budować dużą infrastrukturę na takie wydarzenia można użyć już gotowego produktu. CDN z reguły pełni może pełnić też rolę load balancera, ochrony DDoS (o ile serwis został poprawnie skonfigurowany) oraz filtra WAF chroniącego przed atakami aplikacyjnymi lub spamem.

Idź na stronę: <https://getbootstrap.com/>

Poszukaj na stronie jak użyć CDN.

Build fast, responsive sites with Bootstrap

Powerful, extensible, and feature-packed frontend toolkit. Build and customize with Sass, utilize prebuilt grid system and components, and bring projects to life with powerful JavaScript plugins.

```
$ npm i bootstrap@5.2.1
```

[Read the docs](#)

Currently **v5.2.1** · [Download](#) · [v4.6.x docs](#) · [All releases](#)

Klikamy Read the docs. Następnie tworzymy plik **index.html** jak pokazano na stronie internetowej.

Możemy też dołączyć także Popper.js (przed zamknięciem sekcji `</body>`).

Popper.js nie jest biblioteką, a podstawą do budowania jej. **Można powiedzieć, że jest to maszyna do ustawiania i pozycjonowania elementów, takich jak tooltipy.**

PRZYKŁAD 1

Index.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <!-- Bootstrap is developed mobile first -->
    <!-- BOOTSTRAP - important for proper rendering and touch zooming for all devices -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>jQUERY</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="css/master.css">
  </head>
  <body>
    <div id="container">

      <div id="page">
        <div class="wymiar"></div>
        <div class="wymiar"></div>
      </div>
    </div>
    <!--BOTSTRAP for responsive app only with HTML5-->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-..."></script>
    <!-- POPPER for dropdowns, popovers, or tooltips INCLUDED in bootstrap.bundle.min.js ^ -->
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-..."></script>
    <script src="lib/jquery-3.6.1.js"></script>
    <script type="text/javascript" src="myscript.js"></script>
  </body>
</html>
```

CSS:

```
div{
  margin: 12px;
  padding: 3px;
}

div#page {
  padding: 3px;
  background-color: rgb(180, 251, 152);
}

.wymiar {
  width: 200px;
  height: 40px;
  margin: 6px;
  background-color: rgb(137, 103, 228);
}
```

Strona:



Zmiana w index:

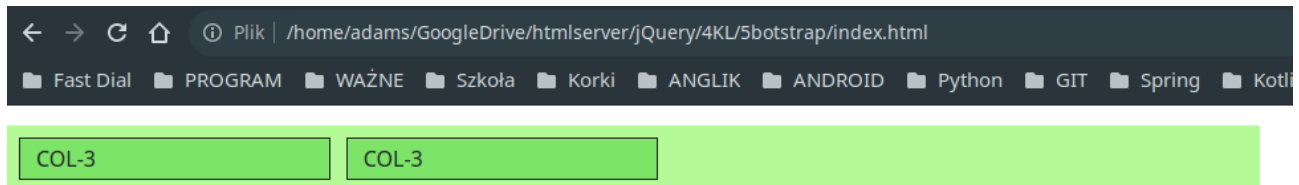
```
<div id="container">

  <div id="page" class="col-6 row">
    <div class="col-3 mydivs">COL-3</div>
    <div class="col-3 mydivs">COL-3</div>
  </div>
</div>
```

oraz w CSS

```
.mydivs {
  background-color: rgb(124, 228, 103);
  border: 1px solid;
  margin: 6px;
}
```

Zmienia się wygląd strony:



dodanie do **id=page** klasy z bootstrapa: **col-6** ustawia szerokość div na 6 kolumn

dodanie klasy z bootstrapa: **col-3** ustawia szerokość div na 3 kolumny

Zauważ że do jednego elementu/znacznika HTML można dodawać wiele klas!!!

Szybkie formatowanie tabel:

```
<table class="table">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">First</th>
      <th scope="col">Last</th>
      <th scope="col">Handle</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Mark</td>
      <td>Otto</td>
      <td>@mdo</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>Jacob</td>
      <td>Thornton</td>
      <td>@fat</td>
    </tr>
    <tr>
      <th scope="row">3</th>
      <td colspan="2">Larry the Bird</td>
      <td>@twitter</td>
    </tr>
  </tbody>
</table>
```

wygląd na stronie:

| # | First | Last | Handle |
|---|----------------|----------|----------|
| 1 | Mark | Otto | @mdo |
| 2 | Jacob | Thornton | @fat |
| 3 | Larry the Bird | | @twitter |

Dostępne schematy kolorów w bootstrapie: (można użyć także do TR, TD)

```
<table class="table-primary">...</table>
<table class="table-secondary">...</table>
<table class="table-success">...</table>
<table class="table-danger">...</table>
<table class="table-warning">...</table>
<table class="table-info">...</table>
<table class="table-light">...</table>
<table class="table-dark">...</table>
```

```
<tr class="table-primary">...</tr>
<tr class="table-secondary">...</tr>
<tr class="table-success">...</tr>
<tr class="table-danger">...</tr>
<tr class="table-warning">...</tr>
<tr class="table-info">...</tr>
<tr class="table-light">...</tr>
<tr class="table-dark">...</tr>
```

Więc jak w naszym kodzie zmienimy zapis:

```
<table class="table table-light">
  <thead>
  <tr>
```

otrzymamy:

| # | First | Last | Handle |
|---|----------------|----------|----------|
| 1 | Mark | Otto | @mdo |
| 2 | Jacob | Thornton | @fat |
| 3 | Larry the Bird | | @twitter |

lub

```
<table class="table table-dark">
  <thead>
  <tr>
```

| # | First | Last | Handle |
|---|----------------|----------|----------|
| 1 | Mark | Otto | @mdo |
| 2 | Jacob | Thornton | @fat |
| 3 | Larry the Bird | | @twitter |

dorzucając:

```
<table class="table table-dark table-striped">
  <thead>
  <tr>
```

Otrzymamy:

| # | First | Last | Handle |
|---|----------------|----------|----------|
| 1 | Mark | Otto | @mdo |
| 2 | Jacob | Thornton | @fat |
| 3 | Larry the Bird | | @twitter |

Można też tak zrobić z kolumnami:

```
<table class="table table-dark table-striped-columns">
  <thead>
  <tr>
```

Przypominam kolumna 2 ma ustawione:

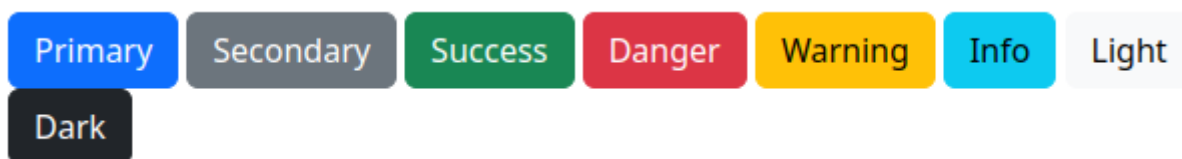
```
<td colspan="2">Larry the Bird</td>
```

| # | First | Last | Handle |
|---|----------------|----------|----------|
| 1 | Mark | Otto | @mdo |
| 2 | Jacob | Thornton | @fat |
| 3 | Larry the Bird | | @twitter |

Wrzućmy jeszcze przyciski:

```
<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-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>
```

Efekt na stronie:



Ponieważ mamy ustawiony DIV w którym są przyciski na [col-4](#) dlatego DARK nie zmieścił się w jednym wierszu.

Dodatkowo każdy z tych przycisków zmienia kolor po najechaniu na niego kursorem myszki.

Po więcej informacji patrz:

<https://getbootstrap.com/docs/5.2/components/buttons/>

3 zadania

CEL:

Używając bootstrapa oraz jQuery stwórz galerię zdjęć z wykorzystaniem:

<https://getbootstrap.com/docs/5.2/components/carousel/>

Użyj przynajmniej 6 zdjęć (zapisz je w folderze lub pobieraj bezpośrednio z internetu) – ścieżki do zdjęć zapisane w tablicy!!!

ZAŁOŻENIA:

Wygeneruj stronę fragment strony HTML (carousel) za pomocą jQuery.

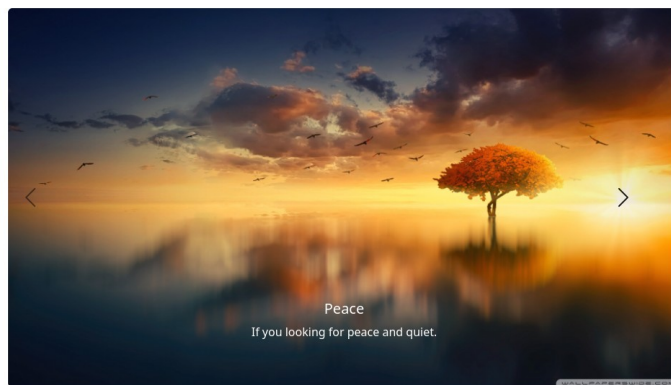
Dodatkowo dodaj możliwość zmiany interwału czasowego przełączania się pomiędzy obrazkami (data-bs-interval="2000"). Czyli użytkownik wpisuje nową wartość w sekundach i po naciśnięciu przycisku ZATWIERDŹ nastąpi zmiana czasu przełączania między obrazkami.

Ustaw wielkość miejsca gdzie wyświetla się obrazek na 6 kolumn, obok daj możliwość zmiany interwału czasowego – w następnych 6 kolumnach.

Pod spodem galerii należy umieścić tabelkę w której znajdzie się:

Lp. Nazwa obrazka Wielkość w kB obrazka tak aby co drugi wiersz był ciemniejszy/jaśniejszy ALE NIE nagłówek z opisem (tabela ma być zrobiona w HTML z wykorzystaniem bootstrapa).

PRZYKŁADOWY WYGLĄD



Wpisz co ile sekund ma zmieniać się obrazek

ZATWIERDŹ

Zwróć uwagę na galerię. Napis, przełączniki na obrazku oraz pod obrazkiem!

Oczywiście jeśli zrobisz ładniejszy wygląd to super!!!