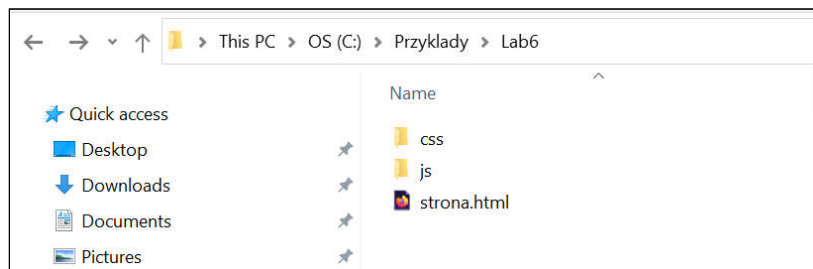


Tworzenie Aplikacji Internetowych

Laboratorium 6

Wykorzystanie bibliotek tworzenia interfejsów – na przykładzie biblioteki Bootstrap 5.1

Biblioteka Bootstrap jest darmowa i można ją pobrać pod adresem: <https://getbootstrap.com/> (przycisk **download** umieszczony w prawym górnym rogu). Możliwa jest także instalacja biblioteki za pomocą Node.js. Najprostszy sposób wykorzystania biblioteki to rozpakowanie pobranych plików i umieszczenie katalogów **js** oraz **css** bezpośrednio w folderze tworzonej strony internetowej:



Katalogi te zawierają kilka wersji biblioteki i komponentów różniących się zawartością (szczegóły można znaleźć tutaj: <https://getbootstrap.com/docs/5.1/getting-started/contents/>). Podstawowe pliki biblioteki to:

- js/bootstrap.min.js
- css/bootstrap.min.css

Aby wykorzystać podstawową wersję biblioteki należy zatem załączyć oba pliki w nagłówku head. Przedstawia to następujący przykład:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <script src="js/bootstrap.min.js"></script>
  </head>
  <body>

  </body>
</html>
```

Bibliotekę można także łączyć w wersji online – szczegóły: <https://getbootstrap.com/docs/5.1/getting-started/introduction/>

Wykorzystanie biblioteki.

Elementy biblioteki można tworzyć umieszczając odpowiednie **znaczniki na stronie**. Takim znacznikom należy nadać odpowiednie wartości atrybutów **class** (często nadaje się kilka klas oddzielonych znakiem spacji), a także (zależnie od funkcjonalności) atrybuty poprzedzone przedrostkiem **aria** (funkcjonalność) oraz **data** (dane). Przykład utworzenie przycisku o przypisanych trzech klasach związanych z Bootstrap:

```
<button class="btn btn-success btn-lg">Przycisk</button>
```

Przycisk

W powyższym przykładzie klasa **btn** oznacza przycisk z biblioteki Bootstrap, klasa **btn-success** oznacza styl przycisku (zielony), natomiast klasa **btn-lg** oznacza wielkość przycisku (lg – Large – duży). Inne wartości klas które można nadać (więcej w dokumentacji: <https://getbootstrap.com/docs/5.1/components/buttons/>):

- klasa btn-primary – przycisk standardowy (niebieski)
- klasa btn-warning – przycisk ostrzeżenia (żółty)
- klasa btn-dark – przycisk ciemny (czarny)
- klasa btn-sm – przycisk mały
- klasa btn-lg – przycisk duży
- atrybut disabled – przycisk nieaktywny

Elementy biblioteki Bootstrap posiadają także dodatkowe **metody**. Aby je wykorzystać lub obsłużyć, nie wystarczy odnaleźć obiekt DOM przedstawiający dany element, ponieważ obiekt DOM nie posiada informacji o rozszerzanej funkcjonalności. Zamiast tego należy utworzyć nowy obiekt **Bootstrap** odpowiedniego typu i podać jako parametr odnaleziony obiekt DOM. Przykładem może być przycisk, który posiada metodę toggle (oznaczenie przycisku jako aktywny). Aby wywołać taką metodę należy to zrobić następująco:

```
<body>
  <button id="pierwszy" class="btn btn-danger btn-lg">Pierwszy</button>
  <button id="drugi" class="btn btn-danger btn-lg">Drugi</button>
  <script>
    let pierwszy = document.getElementById("pierwszy"); // odnajdujemy obiekt DOM
    let przycisk = new bootstrap.Button(pierwszy); // przekształcamy na Bootstrap button
    przycisk.toggle(); // wywołujemy metodę zmieniającą stan przycisku Bootstrap button
  </script>
</body>
```

Wynik – pierwszy przycisk ma ciemniejszy kolor (stan został zmieniony):



Nawigacja

Innym przykładowym elementem może być menu **nawigacji**. Taki element tworzy się ze standardowych znaczników **ul** i **li**. Znacznikowi **ul** należy nadać klasę **nav** oraz opcjonalnie klasy:

- justify-content-center – wyrównanie nawigacji do środka
- justify-content-end – wyrównanie nawigacji do prawej
- flex-column – menu horyzontalne
- nav-tabs – menu w formie zakładek
- nav-pills – menu w formie przycisków
- nav-fill – menu rozszerzone na całą szerokość dostępnej przestrzeni

Elementy menu umieszcza się wewnątrz znacznika **li**, któremu należy nadać klasę **nav-item**. Opcjonalnie:

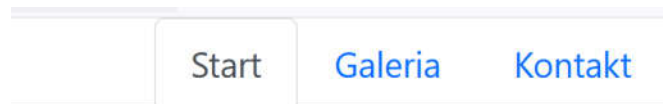
- dropdown – element nawigacji będzie posiadał znaczek rozwijania podmenu (do rozwijanego menu należy wykorzystać popper i załączyć zamiast standardowej biblioteki js/bootstrap.bundle.min.js)

Wewnątrz znaczników li można umieścić np. **hiperłącza** (class nav-link oraz active dla aktywnego linku) lub **przyciski** (class nav-link oraz active dla aktywnego przycisku). Do zapewnienia poprawnej funkcjonalności skryptu należy także zaznaczonej pozycji z menu przypisać atrybut aria-selected="true".

Aby menu było **interaktywne** (np. automatycznie zmieniało zaznaczenie przy wywoływaniu jedynie funkcji JavaScript), należy dodać odpowiednie atrybuty roli (do ul – tablist, do li – presentation, do przycisku – tab, oprócz tego, należy do przycisku dodać jeszcze atrybut związany z danymi - data-bs-toggle="tab"). Bez takich atrybutów nawigacja jest statyczna (może np. otwierać zewnętrzne strony, wewnątrz których należy innemu elementowi menu przypisać wartość active).

Przykład nawigacji interaktywnej (zmieniające się zakładki po kliknięciu):

```
<ul class="nav nav-tabs justify-content-end" role="tablist">
  <li class="nav-item" role="presentation">
    <button class="nav-link active" data-bs-toggle="tab" role="tab"
      aria-selected="true">Start</button>
  </li>
  <li class="nav-item" role="presentation">
    <button class="nav-link" data-bs-toggle="tab" role="tab">Galeria</button>
  </li>
  <li class="nav-item" role="presentation">
    <button class="nav-link" data-bs-toggle="tab" role="tab">Kontakt</button>
  </li>
</ul>
```



Kolumnowy układ wyświetlania informacji na stronie

Bootstrap posiada 12 kolumnowy system budowania układu strony oparty o flexbox. Layout tworzy się za pomocą znaczników **div**. Głównym elementem jest **div** o klasie **container** (responsywne dopasowanie strony) lub klasie **container-fluid** (responsywne dopasowanie strony z rozciągnięciem do 100% szerokości) lub klasie **container-md** (połączenie powyższych).

Wewnątrz głównego diva należy utworzyć div o klasie **row** (wiersz), wewnątrz którego można układać kolumny o klasie **col** (do 12 kolumn). Każdej kolumnie można również ustawić zajmowany obszar (np. **col-2** zajmie 2 kolumny, **col-4** zajmie 4 kolumny). Kolumny zawsze dopasowują się do łącznej liczby kolumn, tzn. dodając kolumnę bez doawania zajmowanego obszaru **col** oraz **col-4**, obszar pierwszej kolumny rozłoży się na 8 (dzięki temu suma kolumn będzie równa 12).

Przykład:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <script src="js/bootstrap.min.js"></script>
    <style>
      div { border: 1px solid gray; }
      .row { height: 200px; }
    </style>
  </head>
  <body>
    <div class="container-fluid">
      <div class="row">
        <div class="col">8 kolumn - nie podano liczby (12 - 4 = 8)</div>
        <div class="col-4">4 kolumny</div>
      </div>
    </div>
  </body>
</html>
```

Powyższy przykład stworzy następującą stronę:

8 kolumn – nie podano liczby (12 – 4 = 8)	4 kolumny
---	-----------

Zmieniając klasę kolumn z col-X na col-sm-X (gdzie X oznacza oczywiście zajmowaną liczbę kolumn) zmienia się sposób wyświetlania strony w przypadku większych rozdzielczości (takie kolumny umieszczają się jedna pod drugą co pozwala uzyskać lepszy efekt na urządzeniach mobilnych).

Bardziej zaawansowany przykład tworzenia układu strony:

```
<div class="container-fluid">
  <div class="row"> <!-- wiersz 1 - logo i menu -->
    <div class="col-sm-2">logo</div>
    <div class="col-sm-10">menu</div>
  </div>
  <div class="row"> <!-- wiersz 2 - lewa główna i prawa sekcja -->
    <div class="col-sm-2">lewa<br>sekcja</div>
    <div class="col-sm-8">główna</div>
    <div class="col-sm-2">prawa<br>sekcja</div>
  </div>
</div>
```

Wygląd przy dużej rozdzielczości:

logo	menu	
lewa sekcja	główna	prawa sekcja

Wygląd przy mniejszej rozdzielczości (urządzenie mobilne):

logo
menu
lewa sekcja
główna
prawa sekcja

Więcej szczegółów o układach strony możliwych do uzyskania za pomocą Bootstrap:

<https://getbootstrap.com/docs/5.1/layout/grid/>

<https://getbootstrap.com/docs/5.1/layout/columns/>

Zadanie 1

- Pobrać bibliotekę **Bootstrap** i umieścić obok tworzonej strony internetowej (jak na stronie pierwszej)
- Utworzyć stronę internetową testując przykład podany na dole trzeciej strony tej instrukcji
- Podmienić zawartość body tak, aby zawierała układ z czwartej strony niniejszej instrukcji (logo, menu, lewa sekcja, główna, prawa sekcja)
- Zmodyfikować układ strony tak, aby "prawa sekcja" została usunięta, a część główna wypełniała pozostałą przestrzeń (usuwając trzecią kolumnę z drugiego wiersza).
- W znaczniku **div** zawierającym tekst "menu" umieścić zamiast tego tekstu interaktywną nawigację ze strony trzeciej tej instrukcji (przekopiować całą zawartość).
- Zmodyfikować menu tak, aby było wyświetlane po lewej stronie układu (tuż obok logo). Aby to zrobić należy usunąć odpowiednią klasę ze znacznika **ul**.
- Za pomocą **CSS** zmienić kolor tła pierwszego wiersza układu strony na #eeeeee (div wewnątrz którego znajduje się logo oraz menu). Zmianę koloru dokonać w dowolny poznany wcześniej sposób.
- Z sekcji style usunąć ustawienie obramowania elementu div: `div { border: 1px solid gray; }`, po czym przetestować utworzoną stronę oraz wysokość wiersza `.row { height: 200px; }`.
- W sekcji "lewa sekcja" umieścić kilka przycisków zgodnie z przykładami ze strony pierwszej i drugiej.
- W sekcji "główna" utworzyć listę **ul** o przypisanej klasie **pagination**, wewnątrz tej listy utworzyć kilka znaczników **li** o przypisanej klasie **page-item**, wewnątrz znaczników **li** utworzyć hiperłącza o przypisanej klasie **page-link** oraz atrybucie **href="#"**. Tekstem wewnątrz kolejnych hiperłączy powinny być kolejne liczby (1, 2, 3, 4, 5...).
- Przetestować również działanie strony przy mniejszej rozdzielczości (zmniejszając okno przeglądarki lub powiększając zawartość strony – Ctrl+scroll myszy w górę).

Zadanie 2

- W ramach zadania należy przetestować poniższy kod (obok strony należy umieścić katalogi js oraz css z biblioteki Bootstrap).
- Wyszukać trzy dowolne obrazki w internecie i uzupełnić do nich odnośniki w poniższym kodzie (parametr src znaczników img)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <script src="js/bootstrap.min.js"></script>
    <style>
      .carousel-item { height:100vh; }
      .carousel-item:nth-child(1) { background: silver; }
      .carousel-item:nth-child(2) { background: gray; }
      .carousel-item:nth-child(3) { background: black; }
    </style>
  </head>
  <body>
    <div id="carouselExampleCaptions" class="carousel slide" data-bs-ride="carousel">
      <div class="carousel-inner">
        <div class="carousel-item active">
          
          <div class="carousel-caption d-none d-md-block">
            <h5>Pierwszy element</h5>
            <p>Tekst</p>
          </div>
        </div>
        <div class="carousel-item">
          
        </div>
      </div>
    </div>
  </body>
</html>
```

```

        <div class="carousel-caption d-none d-md-block">
            <h5>Drugi element</h5>
            <p>Tekst</p>
        </div>
    </div>
    <div class="carousel-item">
        
        <div class="carousel-caption d-none d-md-block">
            <h5>Trzeci element</h5>
            <p>Tekst</p>
        </div>
    </div>
</div>
<button class="carousel-control-prev" type="button"
    data-bs-target="#carouselExampleCaptions" data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
</button>
<button class="carousel-control-next" type="button"
    data-bs-target="#carouselExampleCaptions" data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
</button>
</div>
</body>
</html>

```

Uwaga: dla Bootstrap 5 można zainstalować wtyczkę do Visual Studio Code, która ułatwia tworzenie elementów Bootstrap. Można taką funkcjonalność przetestować samodzielnie w domu.