

NATALIA SOKOŁOWSKA

WYZWANIA BEDEPROGRAMISTKA

**Poznam
Bootstrapa 4
ze
Strefą Kursów**

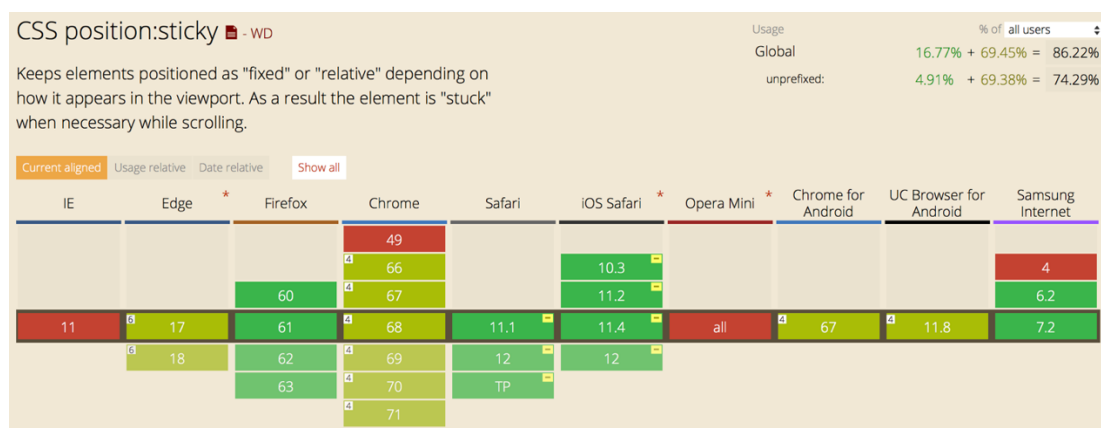
BEDEPROGRAMISTKA.PL

Display – układanie layoutu na stronie

Bootstrap 4 oferuje wiele rozwiązań do układania elementów.

Istnieje kilka klas, które warto znać:

- **Przyklejanie elementów** – służy do przyklejania elementów, niezależnie od przewijania strony (np. menu lub footer). Wyróżniamy tutaj dwie podstawowe klasy, które wystarczy dodać do elementów, które chcemy przykleić:
 - *.fixed-top/bottom* – absolutne położenie na górze/dole. Element ten zasłania górną część strony, dlatego warto pamiętać o tym, aby dodać padding w części górnej lub dolnej (w tej części, która może zostać zasłonięta przez przyklejoną część).
 - *.sticky-top* – element przyklejony, który w normalnym położeniu nie przykrywa zawartości. W momencie scrollowania, element ten automatycznie przykleja się do górnej krawędzi. Ale na dzień dzisiejszy (sierpień 2018) nie jest to jeszcze wspierane przez wszystkie przeglądarki



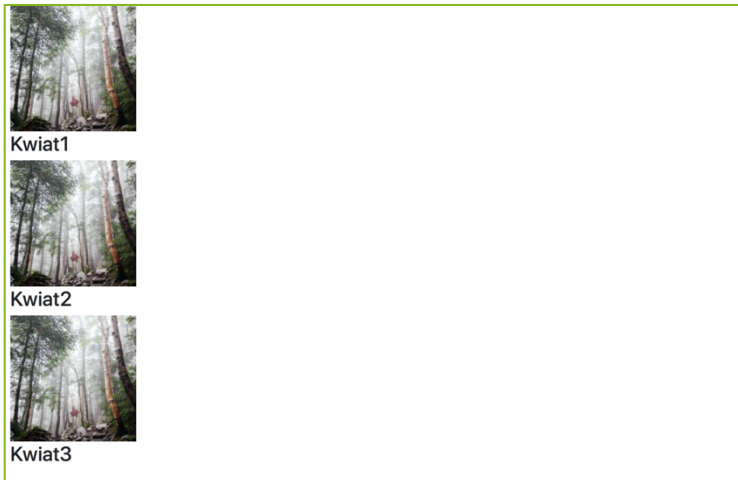
Źródło: <https://caniuse.com/#search=sticky>

- **Display** – możliwości wyświetlania elementów. Klasy te przyjmują schemat *.d-wartość*. Te wartości to taki trochę skrót, który zwyczajowo definiuje się w *CSS*. Można do nich dodawać breakpoints czyli np. tak będzie wyglądała klasa, która sprawi, że element na urządzeniach powyżej 992px będzie wyświetlany jako element blokowy *.d-lg-block*;
 - block

- inline
- inline-block
- flex
 - flex-column – układ kolumnowy
 - flex-row – układ wierszowy
 - flex-column/row-reverse – układ kolumnowy/wierszowy odwrócony
 - flex-wrap/nowrap – zawijanie elementów/niezawijanie
- None
- Table
- Table-cell
- Table-row
- Inline-flex

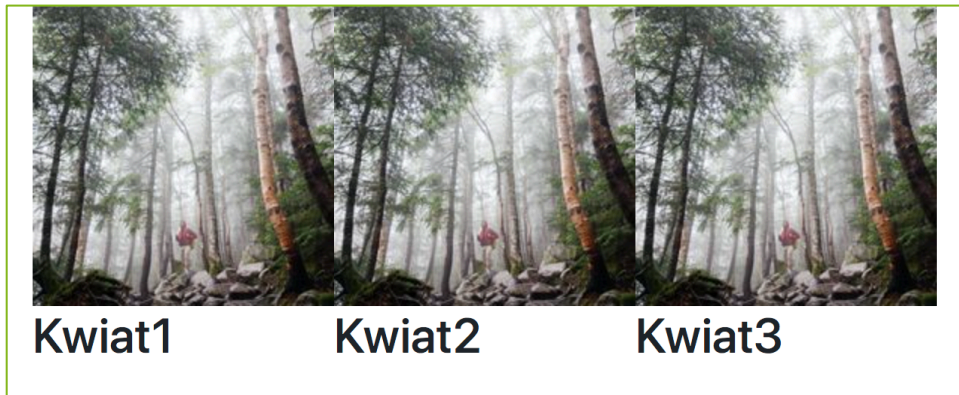
Króciutko o Flexboxie na przykładzie Bootstrapa

Przykład użycie klasy `.flex`. Mamy stworzone 3 boxy, z których każdy zawiera zdjęcie i napis. Boxy domyślnie układają się w sposób blokowy.



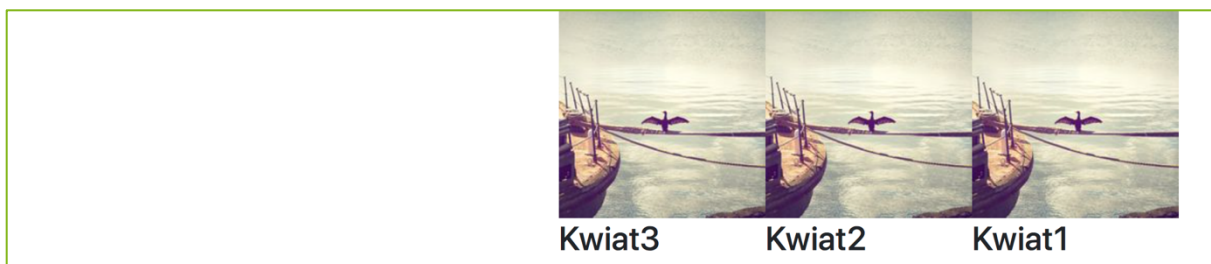
```
<div class="container">
  <div class="main">
    <div class="box">
      
      <h2>Kwiat1</h2>
    </div>
    <div class="box">
      
      <h2>Kwiat2</h2>
    </div>
    <div class="box">
      
      <h2>Kwiat3</h2>
    </div>
  </div>
</div>
```

Po dodaniu do klasy `.main` (to nasza klasa potrzebna na poczet przykładu) klasy `.d-flex` od razu zmienił nam się układ na kolumnowy (we Flexboxie domyślną wartością jest `.flex-row`).



```
<div class="container">
  <div class="main d-flex">
    <div class="box">
      
      <h2>Kwiat1</h2>
    </div>
    <div class="box">
      
      <h2>Kwiat2</h2>
    </div>
    <div class="box">
      
      <h2>Kwiat3</h2>
    </div>
  </div>
</div>
```

Klasa `.flex-row/column-reverse` sprawia, że kolejność elementów zostaje odwrócona.



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

```

<div class="main d-flex flex-row-reverse">
  <div class="box">
    
    <h2>Kwiat1</h2>
  </div>
  <div class="box">
    
    <h2>Kwiat2</h2>
  </div>
  <div class="box">
    
    <h2>Kwiat3</h2>
  </div>
</div>
</div>

```

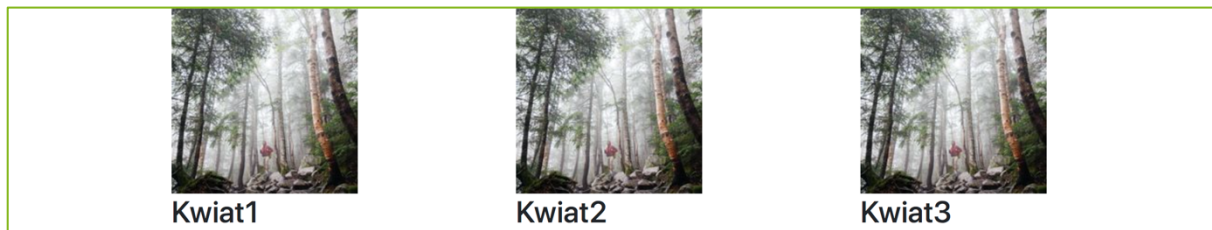
Możemy również dostosowywać wyświetlanie w zależności od rozdzielczości ekranu. Dla smartfonów chcemy, aby układ wyświetlał się kolumnowo. Czyli zastosujemy klasę `.d-flex` oraz klasę `.flex-column`. Ale jednocześnie, w miarę wzrostu dostępnej przestrzeni (przy zwiększaniu rozdzielczości), chcemy, aby układ zmienił się na wierszowy. Dlatego wystarczy, że dodamy klasę `.flex-md-row`.

```

<div class="container">
  <div class="main d-flex flex-column flex-md-row">
    <div class="box">
      
      <h2>Kwiat1</h2>
    </div>
    <div class="box">
      
      <h2>Kwiat2</h2>
    </div>
    <div class="box">
      
      <h2>Kwiat3</h2>
    </div>
  </div>
</div>

```

Nasze obrazki nie układają się ładnie, dlatego proponuję, abyśmy dodali do nich poznaną już wcześniej klasę `.justify-content-around`, aby je wyśrodkować i dodać między nimi równe marginesy.



```
<div class="container">
  <div class="main d-flex flex-column flex-md-row justify-content-around">
    <div class="box">
      
      <h2>Kwiat1</h2>
    </div>
    <div class="box">
      
      <h2>Kwiat2</h2>
    </div>
    <div class="box">
      
      <h2>Kwiat3</h2>
    </div>
  </div>
</div>
```

Marginesy i paddingi

Bootstrap oferuje dodatkowe klasy, które mogą przydać się do ustawiania marginesów i paddingów:

- m – margines
- p – padding
- y – dla osi y (góra/dół)
- x – dla osi x (lewa/prawa)
- t/r/b/l – top (góra)/ right(prawa) / bottom (dół), left (lewa)

Jak tego używać? Bardzo prosto.

Jeśli chcesz ustawić padding o wartości 2 dla każdej ze stron (góra/prawo/dół/lewo) wystarczy, że dodasz klasę `.p-2`. Dodanie marginesu dolnego o wartości 4, będzie miało zapis `.mb-4`. Gdy będziesz chciał dodać margines o wartości 2, po bokach (prawo/lewo) wystarczy, że dodasz klasę `.mx-2`.

Wartości, które są tutaj używane to wartości rem. 1 = 1 rem. Dopuszczalne są wartości w przedziale 0-5 oraz auto. 1 rem = 16px, 2 rem = 32px, itd..

Na pewno kojarzysz zapis `CSS margin: 0 auto`, służący do wyśrodkowania danych elementów. Jeśli chciałbyś osiągnąć to samo w Bootstrapie, musisz ustawić elementowi szerokość (`width`) oraz dodać klasę `.mx-auto`.