

Flex

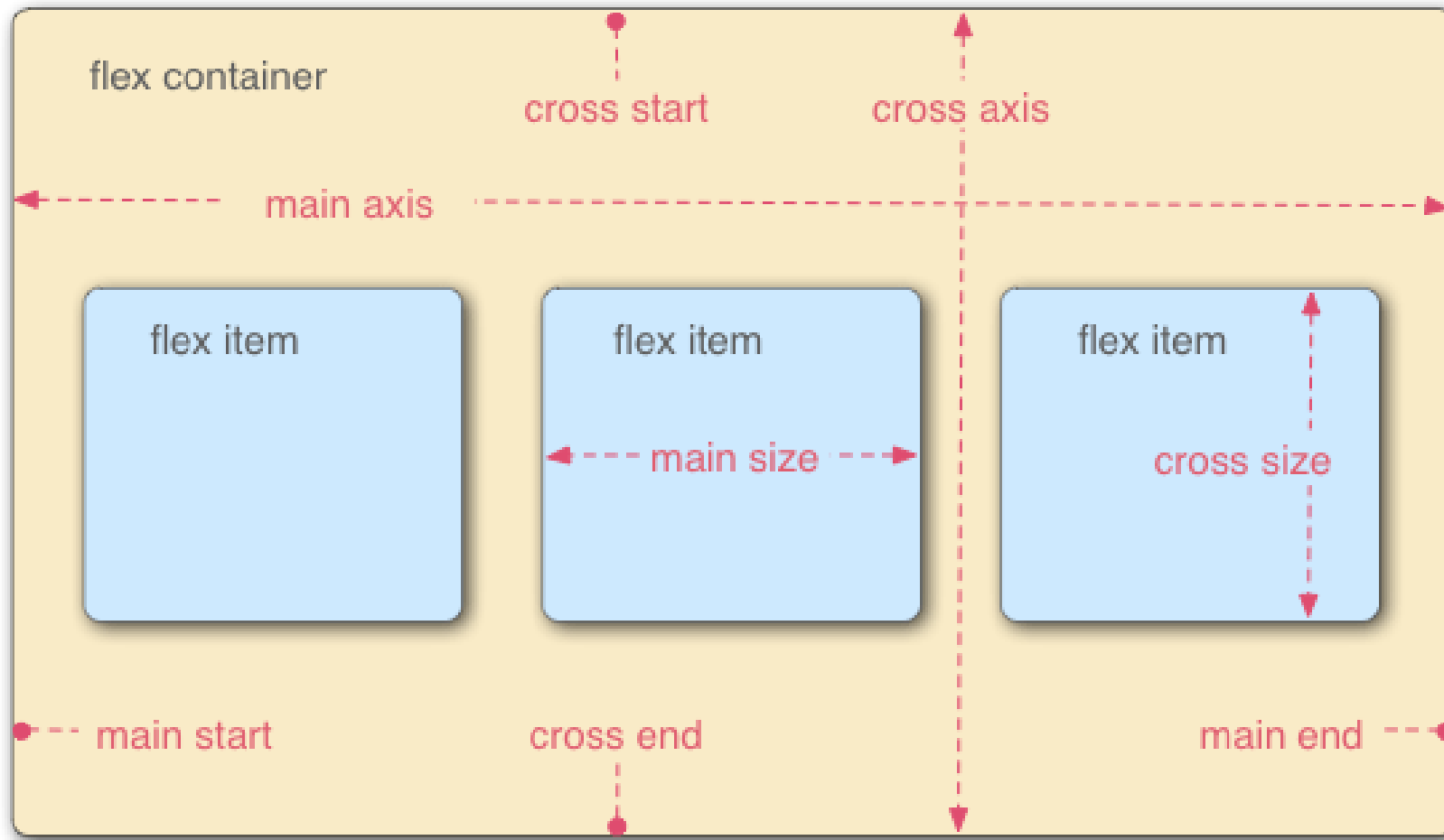
**Flexbox** to narzędzie pozwalające na szybkie tworzenie elastycznego układu strony.

Jest to jednowymiarowa metoda rozmieszczania elementów w wierszach lub kolumnach.

Przez długi czas jedynymi rzeczowymi narzędziami służącymi do tworzenia elastycznego układu strony w CSS były właściwości: `float` i `position`. Mają one jednak istotne ograniczenia.

Można tu wymienić:

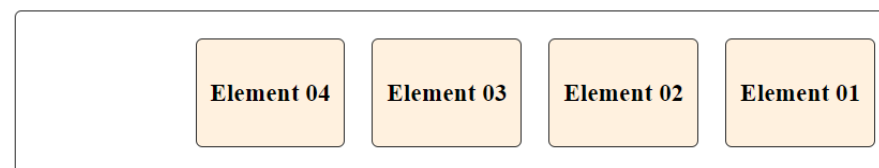
- Pionowe wycentrowanie blokowego elementu z zawartością, względem rodzica.
- Sprawienie by wszystkie "dzieci" w pojemniku zajęły taką samą ilość dostępnej szerokości/wysokości, bez względu na to jaka szerokość/wysokość jest dostępna.
- Umieszczenie kolumn o takiej samej wysokości w wielokolumnowym układzie nawet jeśli zawierają inną liczbę znaków.
- Manipulowanie kolejnością elementów



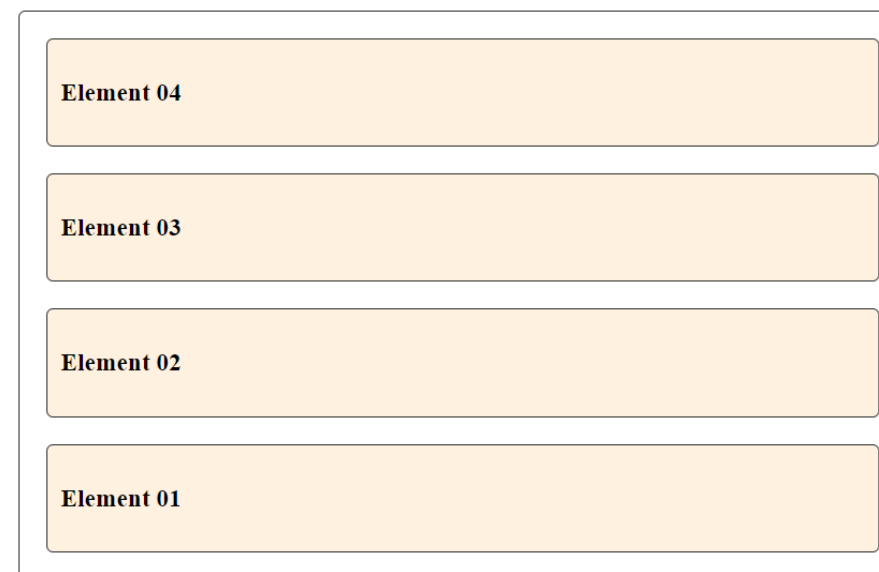
## Kierunek osi głównej (kierunek ustawienia elementów) - flex-direction

```
flex-direction: row;  
flex-direction: row-reverse;  
flex-direction: column;  
flex-direction: column-reverse;
```

row-reverse

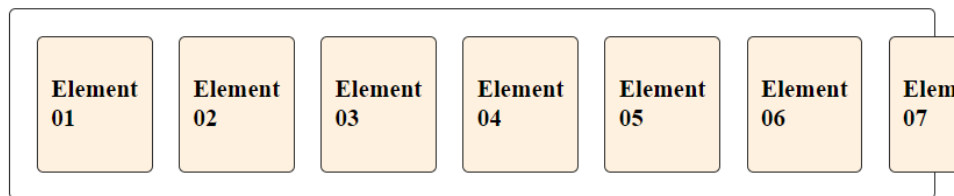


column-reverse

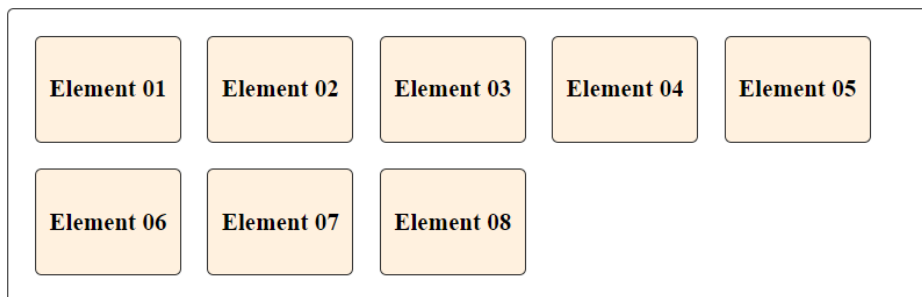


## Zawijanie wierszy - flex-wrap

`flex-wrap: nowrap;`

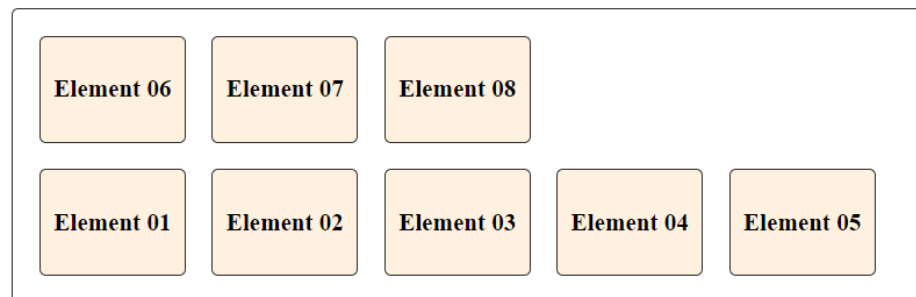


`flex-wrap: wrap;`



Jednym z problemów który pojawia się kiedy umieszczasz dużą liczbę elementów jest to, że mogą one zacząć się wylewać z elementu rodzica niszcząc układ strony.

`flex-wrap: wrap-reverse;`



## flex-flow

flex-flow to kombinacja własności flex-direction i flex-wrap

```
flex-flow: row nowrap;  
flex-flow: row wrap;  
flex-flow: row-reverse wrap;  
flex-flow: row-reverse nowrap;
```

```
flex-flow: column nowrap;  
flex-flow: column wrap;  
flex-flow: column-reverse wrap;  
flex-flow: column-reverse nowrap;
```



## flex – wymiarowanie elementów

UWAGA: Stosujemy dla elementu, nie jak dotychczas dla kontenera

**flex-grow:** niemianowana wartość proporcji, która pozwala wyliczyć jaką część kontenera zajmują poszczególne elementy.

**flex-shrink:** druga niemianowana wartość proporcji, która wchodzi w grę, gdy elementy elastyczne przepełniają pojemnik. Określa, jaka część przepełnionej ilości jest zabierana z rozmiaru każdego elementu aby zapobiec przepełnieniu ich kontenera.

**flex-basis:** minimalny rozmiar elementu. Można go określić indywidualnie za pomocą wartości podanej pikselach.

# Flexbox – wymiarowanie elementów

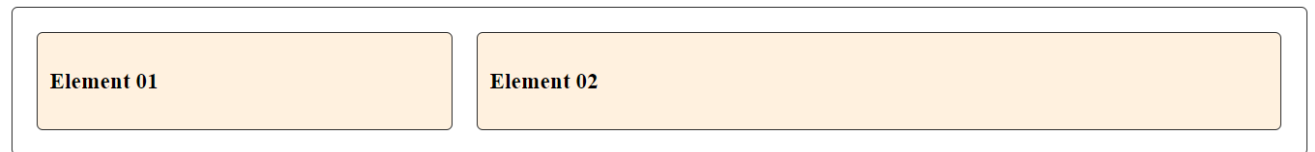


## flex – wymiarowanie elementów

UWAGA: Stosujemy dla elementu, nie jak dotychczas dla kontenera

```
flex-grow: 2;  
flex-shrink: 1;  
flex-basis: 300px;
```

W wersji skróconej:  
`flex: 2 1 200px;`  
grow/shrink/basis;



```
article {  
    flex-grow: 1;  
}  
article:nth-of-type(2n) {  
    flex-grow: 2;  
}
```

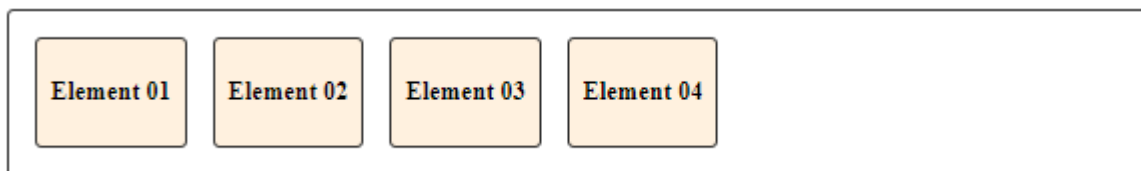


## Flexbox – Wyrównanie poziome

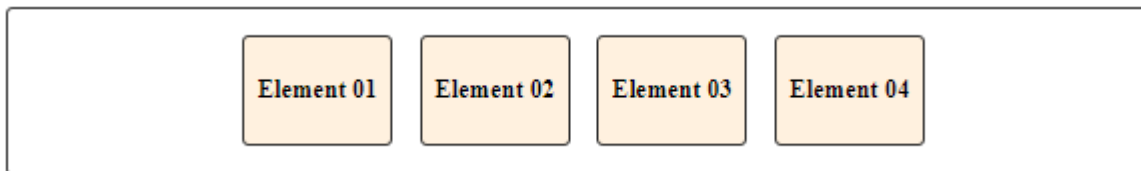


Za pomocą funkcji flexbox można wyrównać elementy flex wzdłuż osi głównej lub poprzecznej.

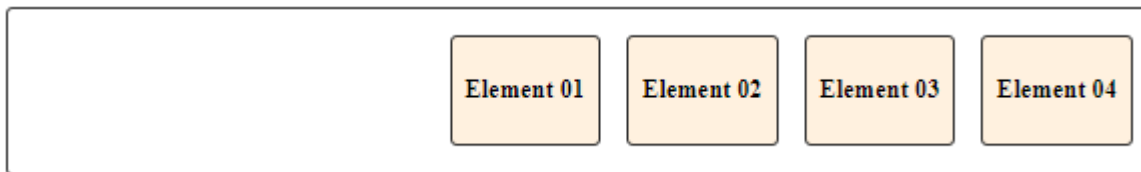
`justify-content: flex-start;`



`justify-content: center;`



`justify-content: flex-end;`

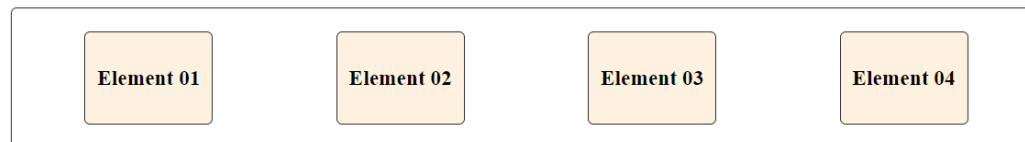


Możemy także zdefiniować jak rozłożona będzie wolna przestrzeń pomiędzy elementami.

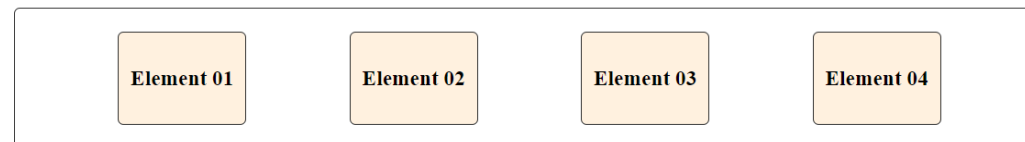
`justify-content: space-between;`



`justify-content: space-around;`



`justify-content: space-evenly;`

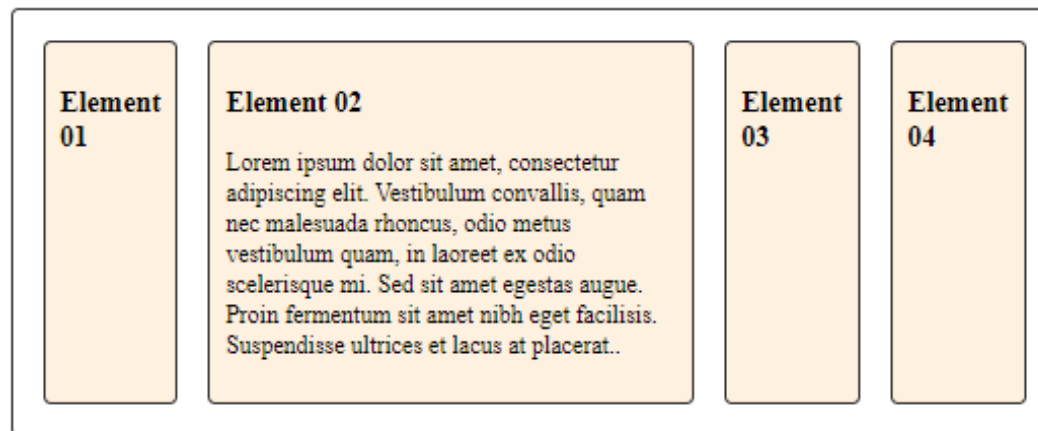


## Flexbox – Wyrównanie w pionie



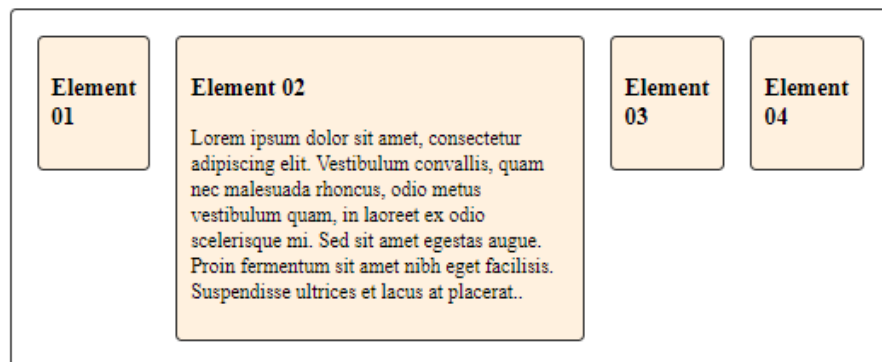
Standardowo wysokość elementów dopasowana zostanie do elementu najwyższego.

Wyrównanie elementów w pionie regulować możemy dodając do stylu kontenera własność **align-items**:

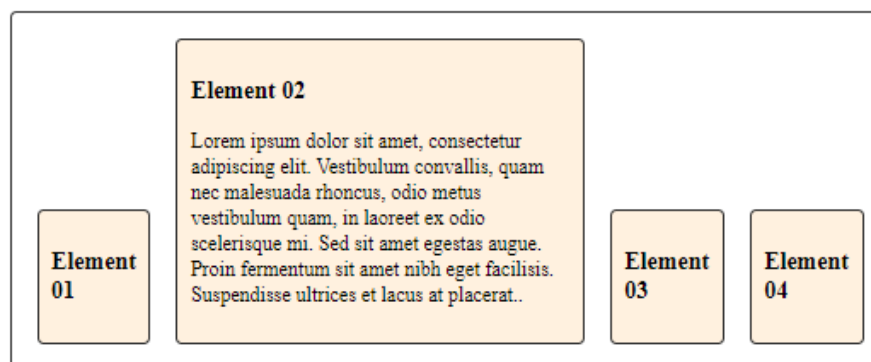


Wartość domyślna: **align-items**: stretch;

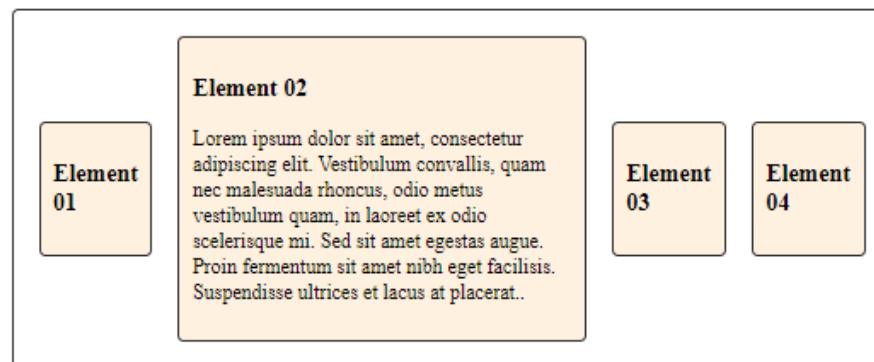
# Flexbox – Wyrównanie w pionie



`align-items: flex-start;`

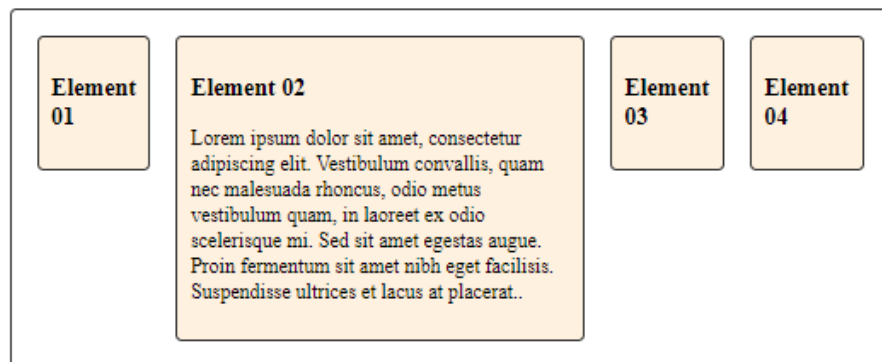


`align-items: flex-end;`

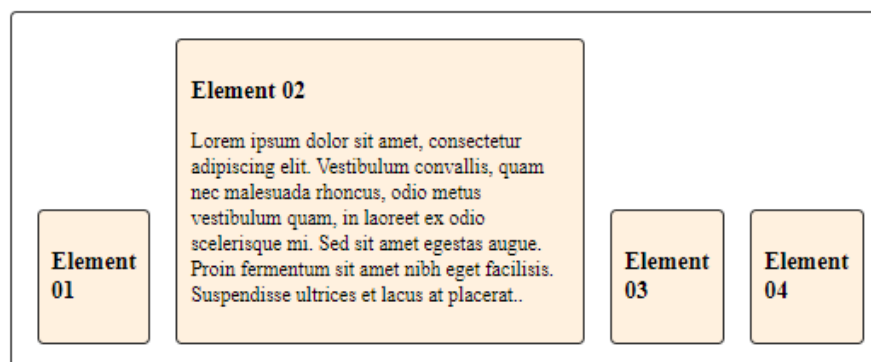


`align-items: center;`

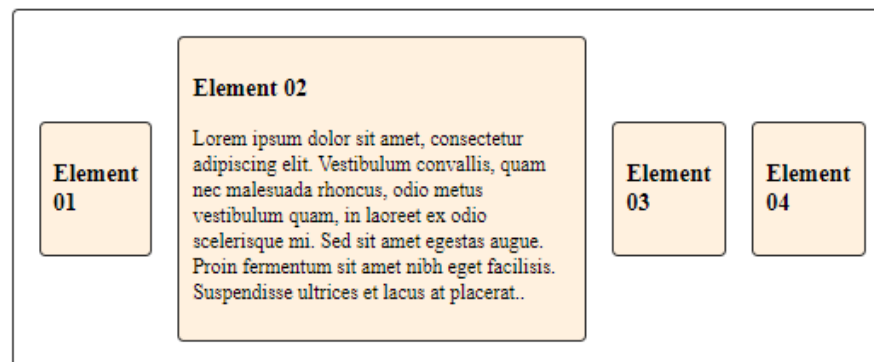
# Flexbox – Wyrównanie w pionie



`align-items: flex-start;`



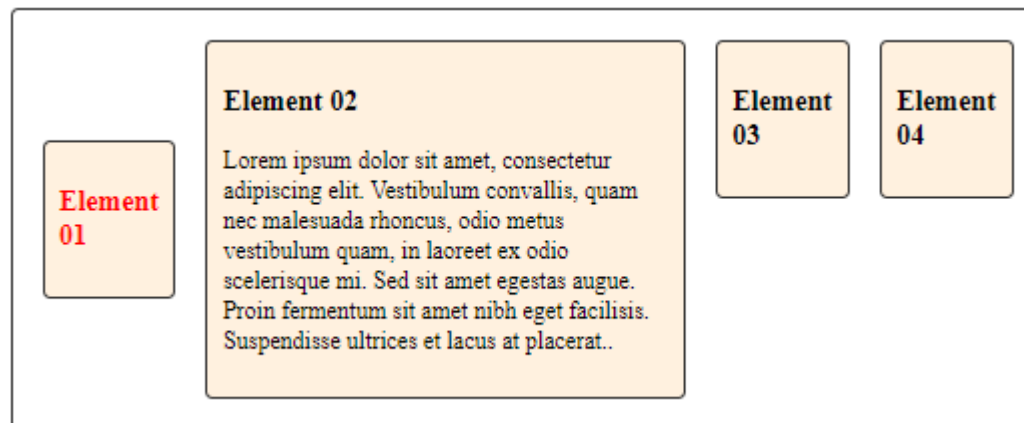
`align-items: flex-end;`



`align-items: center;`

Możliwe jest indywidualne zdefiniowanie sposobu wyrównania w pionie. Służy do tego własność **align-self**:

```
section {  
  display: flex;  
  align-items: flex-start;  
}  
  
article:first-child {  
  align-self: center;  
  color: red;  
}
```



Za pomocą własności `order` manipulować możemy kolejnością elementów.

Zauważyć należy że wartość tego parametru to nie docelowy numer elementu, lecz jego waga.

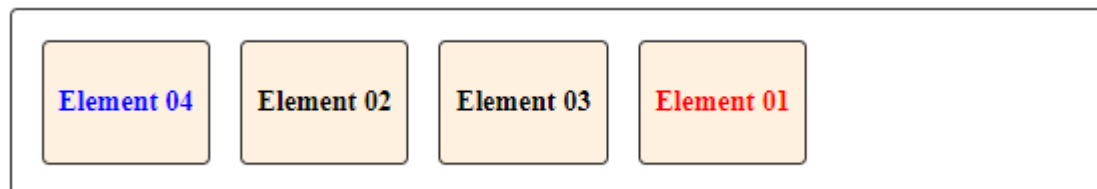
Domyślnie wszystkie elementy mają wagę 0,

- Czym niższa waga tym wcześniej wyrenderowany zostanie element.
- Elementy o takiej samej wadze wyświetlone zostaną w kolejności zgodnej z ich kolejnością w dokumencie html

```
article:first-child {  
  order: 10;  
  color: red;  
}
```

```
article:last-child {  
  order: -1;  
  color: blue;  
}
```

Dopuszczalne są wagi ujemne.



## Tytuł strony

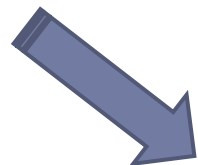
### Tekst 01

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum convallis, quam nec malesuada rhoncus, odio metus vestibulum quam, in laoreet ex odio scelerisque mi. Sed sit amet egestas augue. Proin fermentum sit amet nibh eget facilisis. Suspendisse ultrices et lacus at placerat..

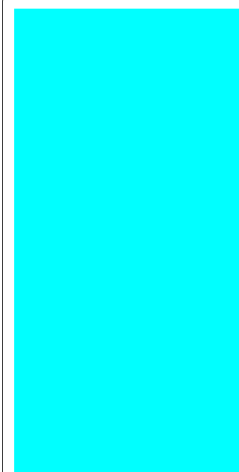
A tu jakaś reklama

autor: wykład

wykonano na wykładzie



## Tytuł strony



### Tekst 01

A tu  
jakaś  
reklama

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum convallis, quam nec malesuada rhoncus, odio metus vestibulum quam, in laoreet ex odio scelerisque mi. Sed sit amet egestas augue. Proin fermentum sit amet nibh eget facilisis. Suspendisse ultrices et lacus at placerat..

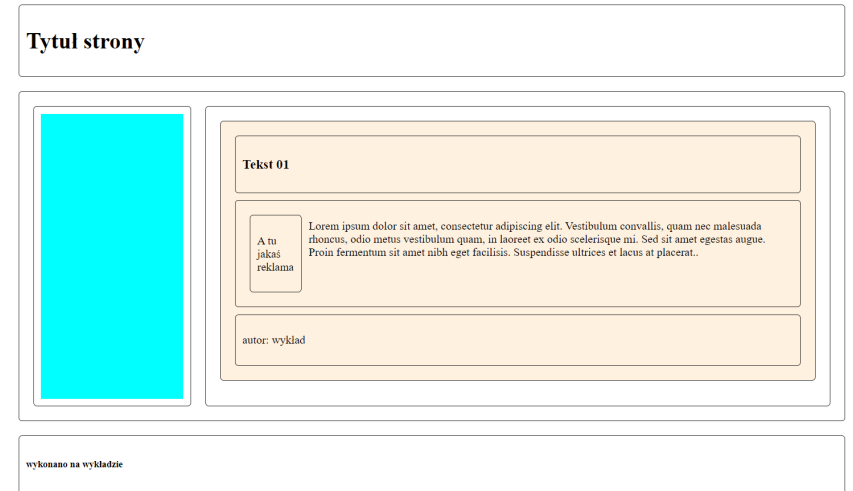
autor: wykład

wykonano na wykładzie





```
<body>
  <header>
    <h1>Tytuł strony</h1>
  </header>
  <main>
    <section>
      <nav>
      </nav>
    </section>
    <section>
      <article>
        <header>
          <h1>Tekst 01</h1>
        </header>
        <main>
          <p>
            Lorem ipsum dolor sit amet, consectetur adipiscing elit...
          </p>
          <aside>
            <p>A tu jakaś reklama</p>
          </aside>
        </main>
      </article>
    </section>
  </main>
  <footer>
    <h5>wykonano na wykładzie</h5>
  </footer>
</body>
```





```
<body>
  <header>
    <h1>Tytuł strony</h1>
  </header>
  <main>
    <section>
      <nav>
      </nav>
    </section>
    <section>
      <article>
        <header>
          <h1>Tekst 01</h1>
        </header>
        <main>
          <p>
            Lorem ipsum dolor sit amet, consectetur adipiscing elit...
          </p>
          <aside>
            <p>A tu jakaś reklama</p>
          </aside>
        </main>
        <footer>
          <p>autor: wykład</p>
        </footer>
      </article>
    </section>
  </main>
  <footer>
    <h5>wykonano na wykładzie</h5>
  </footer>
</body>
```

```
section,
header,
main,
footer,
aside,
article {
  margin: 10px;
  padding: 10px;
  border: 1px solid rgb(47, 47, 47);
  border-radius: 5px;
}
article {
  background-color: rgb(255, 241, 223);
}
body {
  display: flex;
  flex-direction: column;
}
nav {
  width: 200px;
  height: 400px;
  background-color: aqua;
}
body>main {
  display: flex;
}
article>main
{
  display: flex;
}
article>main aside {
  order: -1;
}
```