

Flex

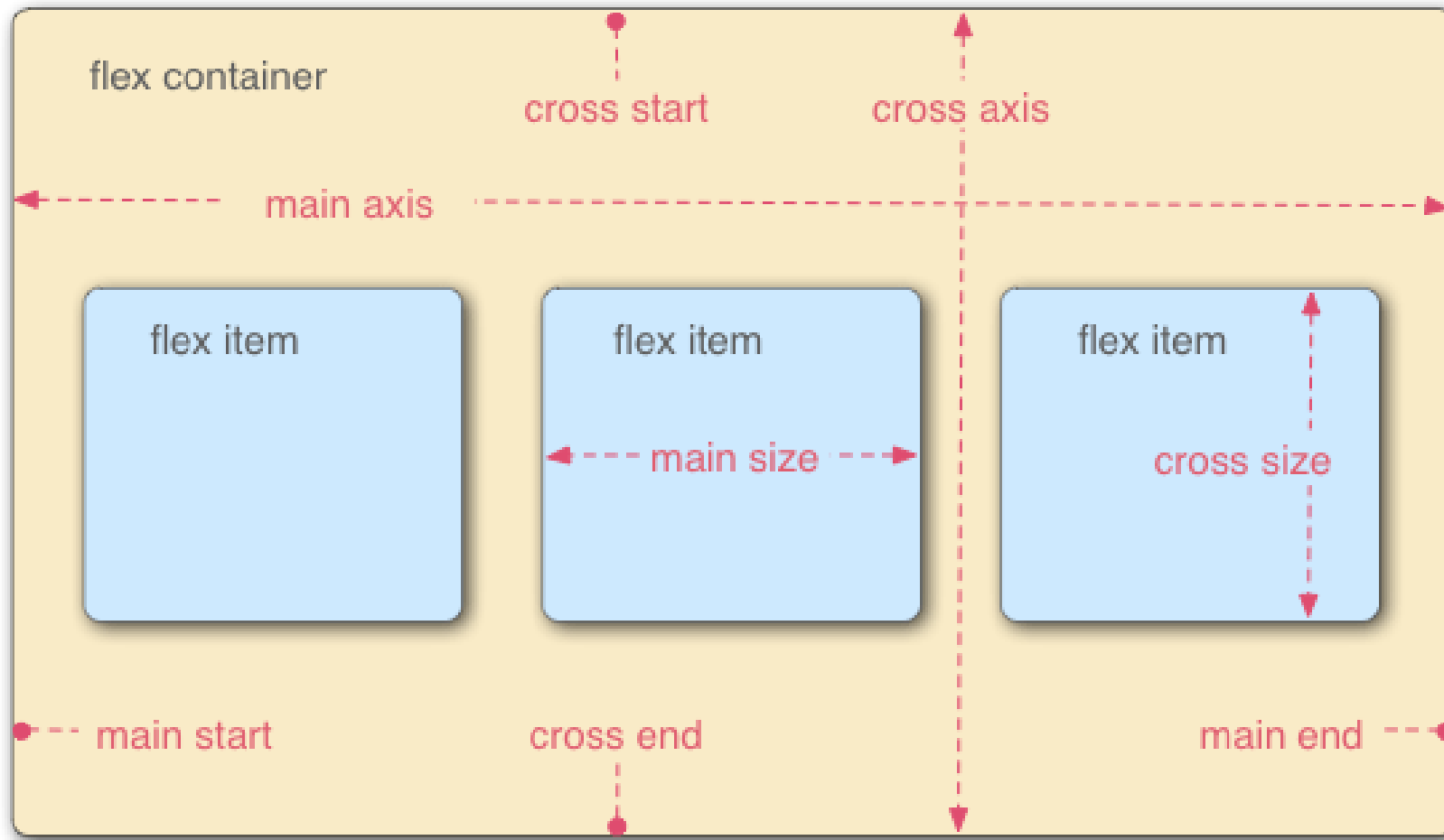
Flexbox to narzędzie pozwalające na szybkie tworzenie elastycznego układu strony.

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:

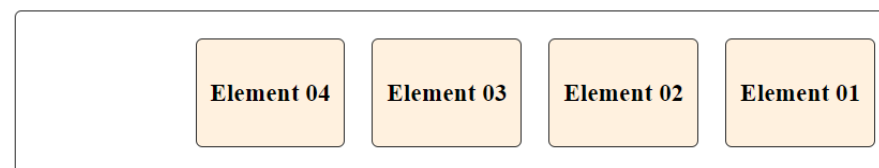
- Pionowe wycentrowanie blokowego elementu z zawartością, względem rodzica.
- Sprawienie by wszystkie "dzieci" w pojemniku zajęły taką samą ilość dostępnej szerokości/wysokości, bez względu na to jaka szerokość/wysokość jest dostępna.
- Umieszczenie kolumn o takiej samej wysokości w wielokolumnowym układzie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



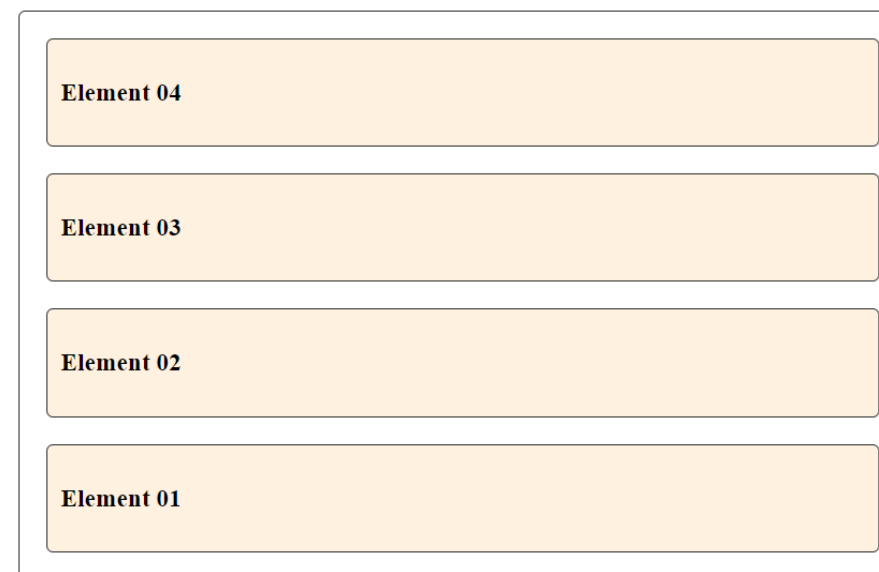
Kierunek osi głównej (kierunek ustawienia elementów) - flex-direction

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

row-reverse

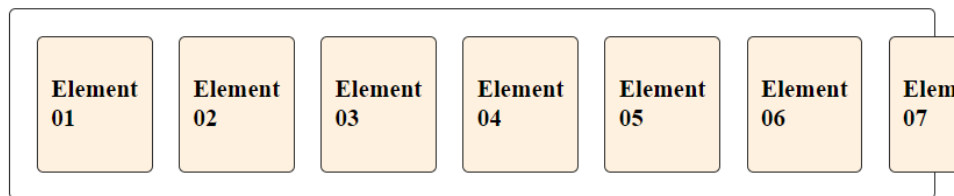


column-reverse

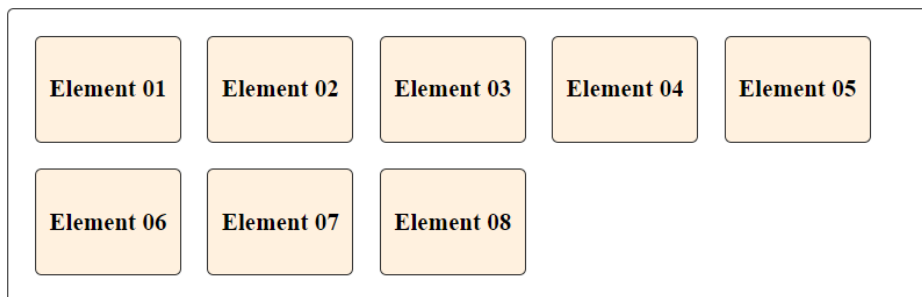


Zawijanie wierszy - flex-wrap

`flex-wrap: nowrap;`

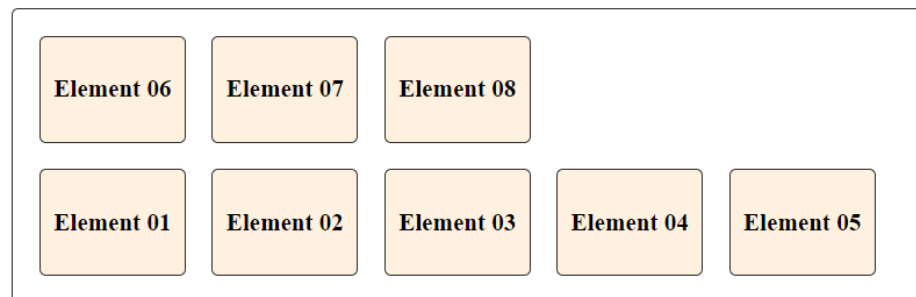


`flex-wrap: wrap;`



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

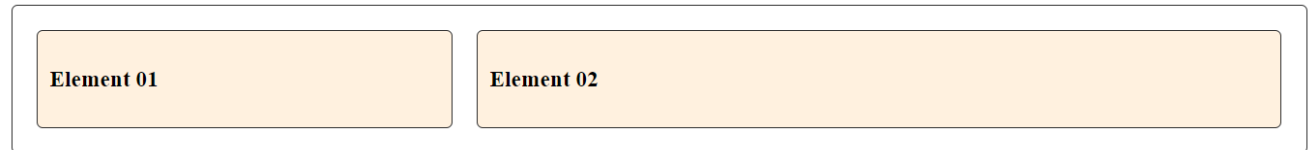


flex – wymiarowanie elementów

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;



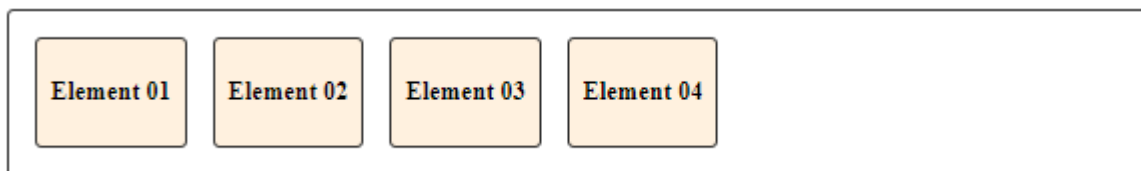
```
article {  
    flex-grow: 1;  
}  
article:nth-of-type(2n) {  
    flex-grow: 2;  
}
```


Flexbox – Wyrównanie poziome

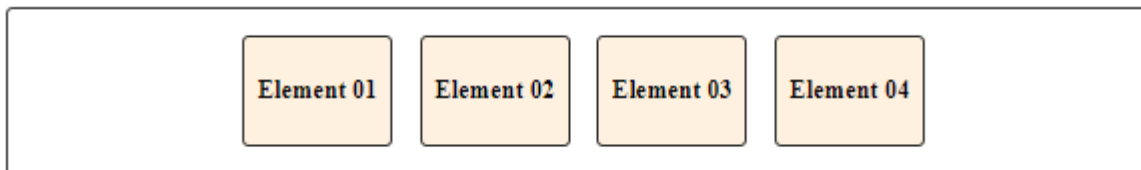


Za pomocą funkcji flexbox można wyrównać elementy flex wzdłuż osi głównej lub poprzecznej.

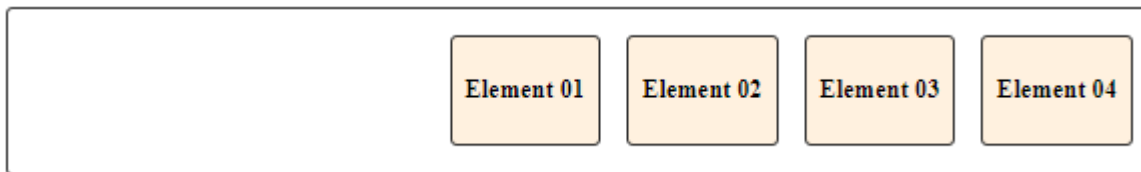
`justify-content: flex-start;`



`justify-content: center;`



`justify-content: flex-end;`

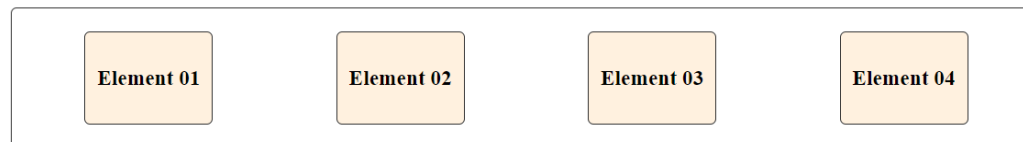


Możemy także zdefiniować jak rozłożona będzie wolna przestrzeń pomiędzy elementami.

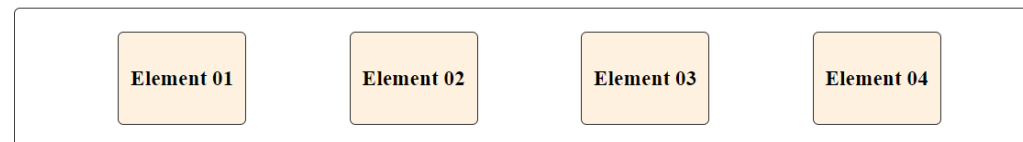
`justify-content: space-between;`



`justify-content: space-around;`



`justify-content: space-evenly;`

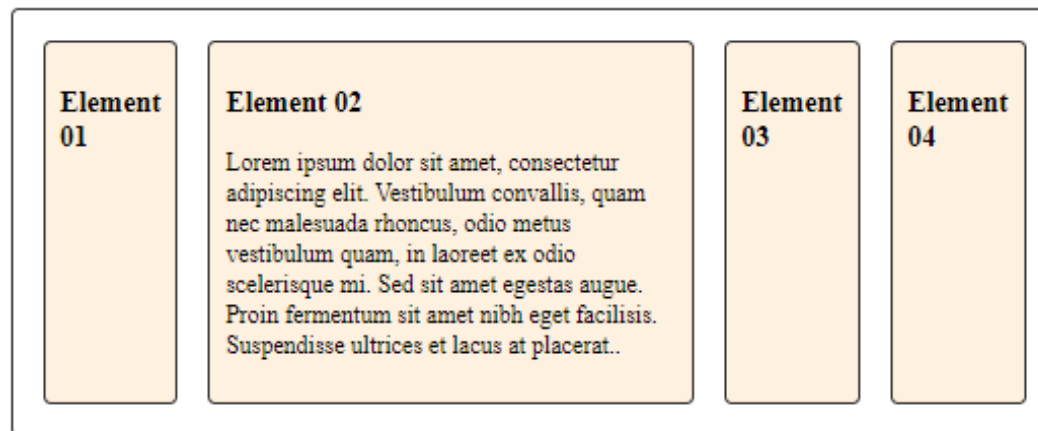


Flexbox – Wyrównanie w pionie



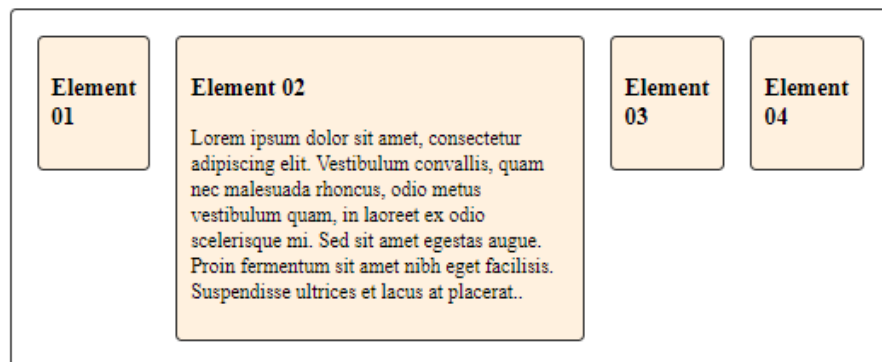
Standardowo wysokość elementów dopasowana zostanie do elementu najwyższego.

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

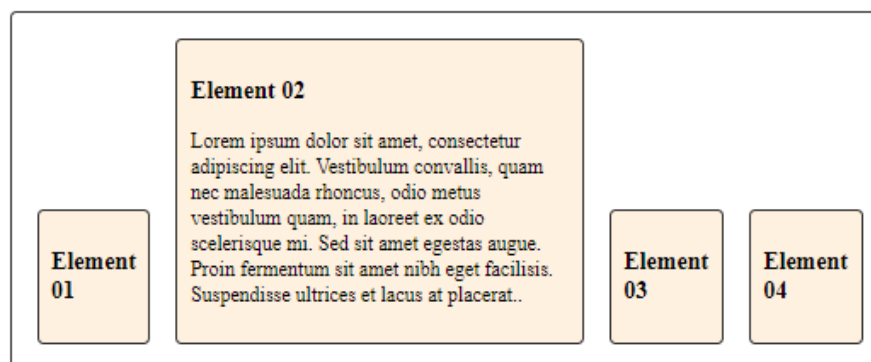


Wartość domyślna: **align-items**: stretch;

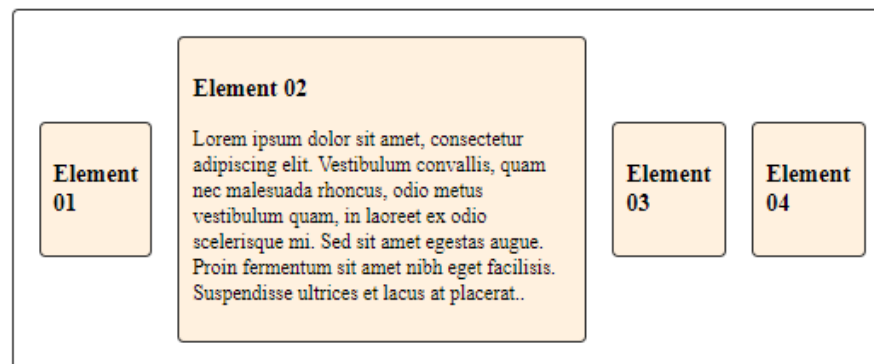
Flexbox – Wyrównanie w pionie



`align-items: flex-start;`

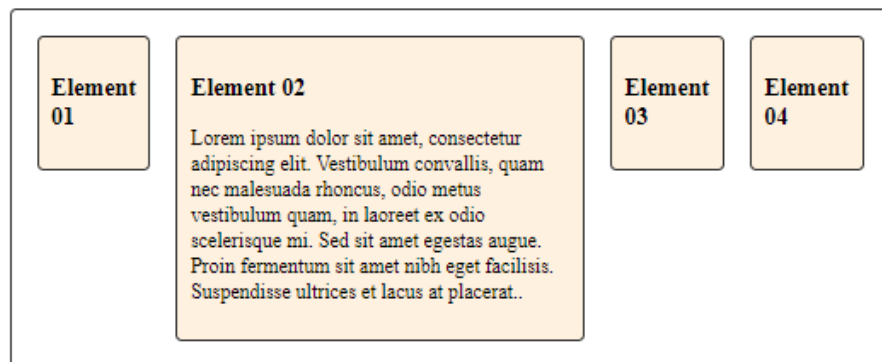


`align-items: flex-end;`

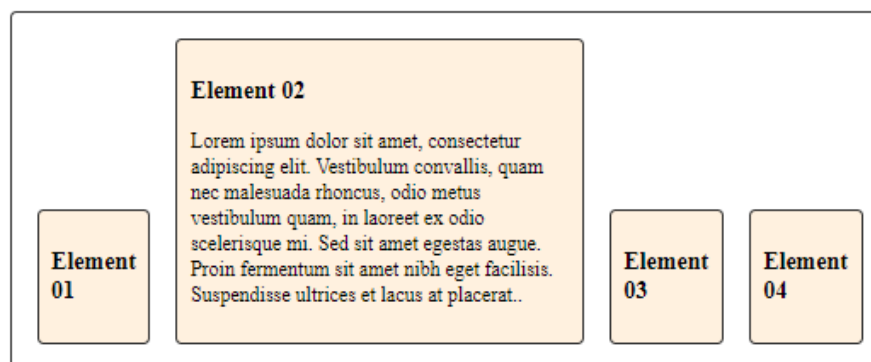


`align-items: center;`

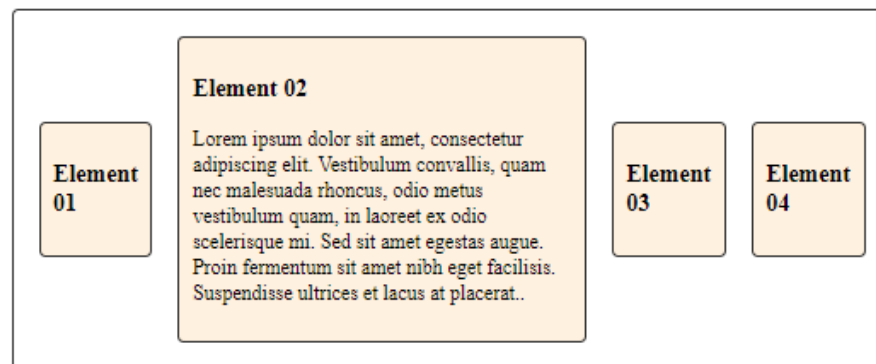
Flexbox – Wyrównanie w pionie



`align-items: flex-start;`



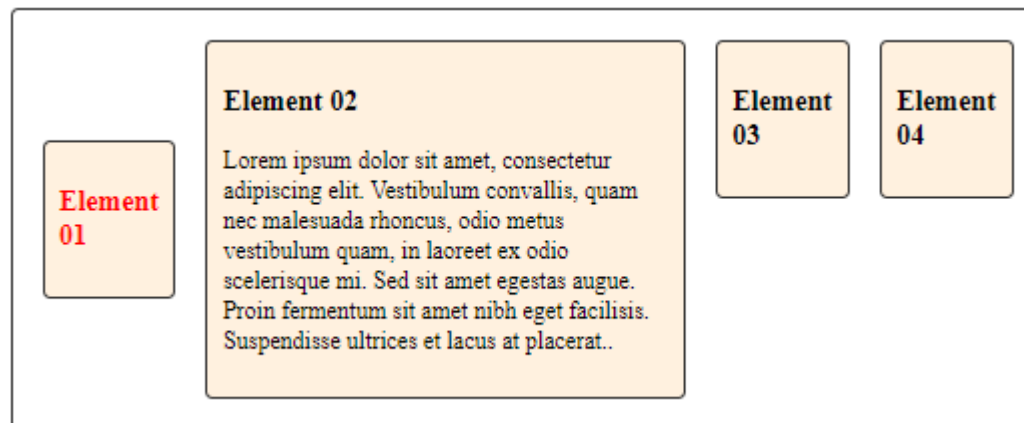
`align-items: flex-end;`



`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;  
}
```



Flexbox – Kolejność elementów



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.

