

Ściągą CSS Flexbox

www.MMCSchool.pl

01. Flex container (rodzic)

Lista właściwości, które nadajemy na rodzica:

- **display:** flex / inline-flex
- **flex-direction:** row / column / row-reverse / column-reverse
- **flex-wrap:** wrap / nowrap
- **align-items:** stretch / flex-start / flex-end / center / baseline
- **justify-content:** flex-start / flex-end / center / space-around / space-between / space-evenly
- **align-content:** stretch / flex-start / flex-end / space-around / space-between / space-evenly
- **gap / row-gap / column-gap:** dowolna wartość, np. 300px, 20rem itd.

Wskazówki:

- Pogrubione niebieskie wartości to wartości domyślne.
- Do **justify-content**, **align-items** itd. możemy dodać inne wartości, np. **start** / **end** (możecie poczytać o tym na **MDN**).

Różnica między **start** a **flex-start** polega na tym, że **start** odnosi się do kierunku tekstu. W Polsce czytamy wszystko od lewej do prawej, więc **start** wskazuje na stronę **lewą**.

W krajach arabskich czyta się od prawej do lewej, więc tam **start** wskazuje na stronę **prawą**.

Flex-start natomiast zawsze wskazuje na początek **flex-direction**.

02. Flex item (dziecko)

Lista właściwości, które nadajemy na dzieci:

- **flex-basis:** **auto** / x (x to dowolna wartość, np. 300px, 20rem itd.)
- **flex-grow:** **0** / x (x to dowolna wartość liczbowa)
- **flex-shrink:** x / **1** (x to dowolna wartość liczbowa)
- **order:** **0** / x (x to dowolna liczba całkowita)
- **align-self:** **auto** / flex-start / flex-end / center / stretch

03. Zapisy skrócone

- **flex:** [flex-grow] [flex-shrink] [flex-basis]
- **flex-flow:** [flex-direction] [flex-wrap]

Przykłady:

- **flex: 1 2 30px;** to zapis skrócony dla:
flex-grow: 1;
flex-shrink: 2;
flex-basis: 30px;
- **flex-flow: column wrap;** to zapis skrócony dla:
flex-direction: column;
flex-wrap: wrap;

Warto pamiętać!

Order zmienia kolejność jedynie w warstwie wizualnej. W DOM kolejność zostaje oryginalna, co jest ważne ze względu na **czytniki ekranów**.

Z **order** należy korzystać tylko wtedy, kiedy nie zaburzamy logiki treści na stronie.

04. Flex-direction

Flex-direction wskazuje, w którą stronę mają układać się elementy (**flex items**) wewnątrz rodzica (**flex container**).

Row



Row-reverse



Column

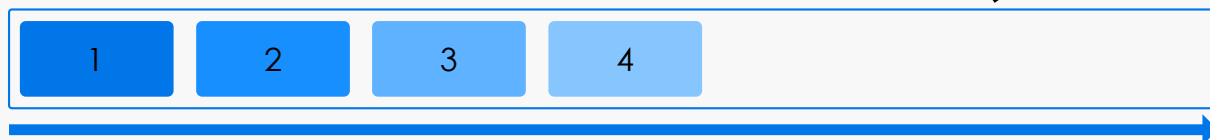


Column-reverse

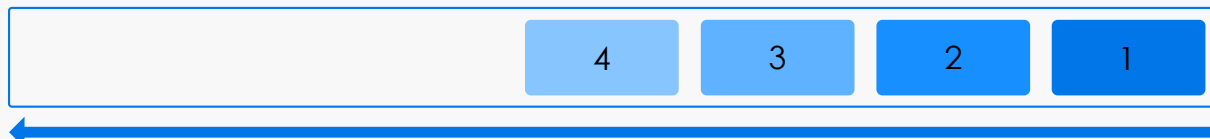


Przykłady:

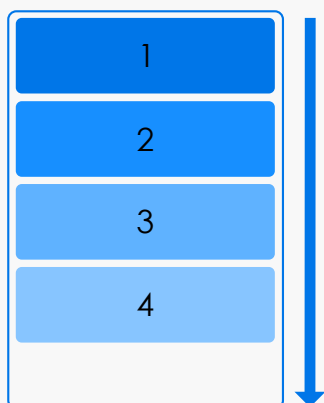
Row



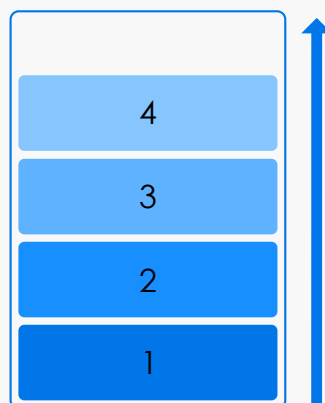
Row-reverse



Column



Column-reverse



05. Justify-content

Justify-content jest osią równoległą do **flex-direction**.

flex-direction: row;



justify-content

flex-direction: column;

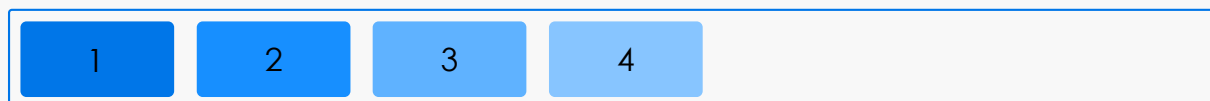


justify-content

Przykłady:

Poniższe przykłady reprezentują zachowanie **justify-content** z **flex-direction: row;**

flex-start (elementy układają się na początku rodzica)



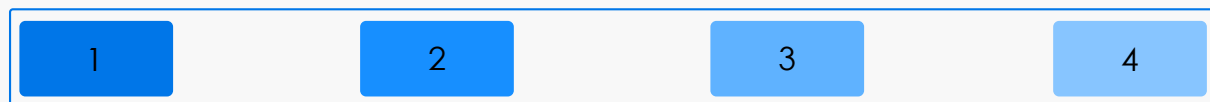
flex-end (elementy układają się na końcu rodzica)



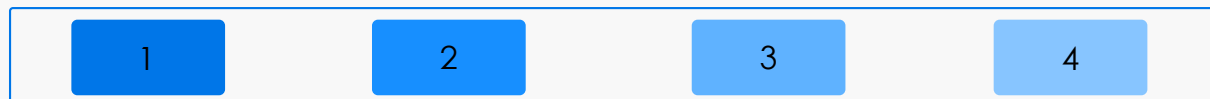
center (elementy są wyśrodkowane)



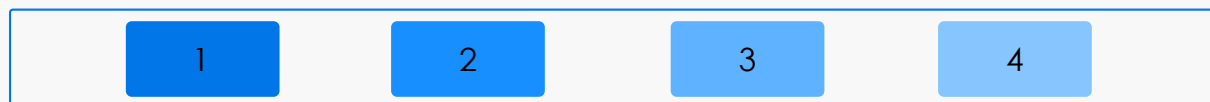
space-between (elementy mają przestrzeń między sobą i są przyklejone do krawędzi)



space-around (elementy mają przestrzeń wokół siebie i krawędzi)



space-evenly (elementy mają równą przestrzeń wokół siebie i krawędzi)



06. Space-around vs space-evenly

Space-around działa podobnie jak marginesy, tzn. sumuje się po bokach.

Jeśli mielibyśmy **120px** wolnej przestrzeni, to każdy element otrzyma **20px** odstępu z lewej i prawej strony (zachowanie podobne do **margin: 0 20px;**).

Dlatego odstęp od krawędzi będzie mniejszy, niż odstęp między elementami.



Space-evenly rozdziela wolną przestrzeń tak, aby odstęp od krawędzi jak i między elementami był dokładnie taki sam.



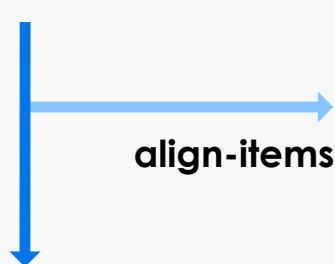
07. Align-items

Align-items jest osią prostopadłą do **flex-direction**.

flex-direction: row;



flex-direction: column;

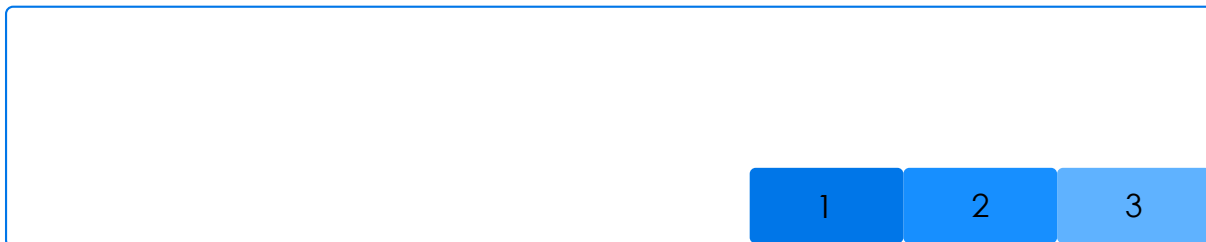
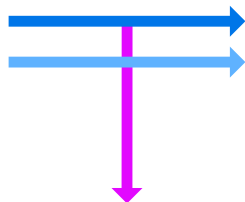


Warto pamiętać!

Przy **flex-direction: row**, **justify-content** pracuje na osi X, a **align-items** na osi Y.
Przy **column**, **justify-content** pracuje na osi Y, a **align-items** na osi X.

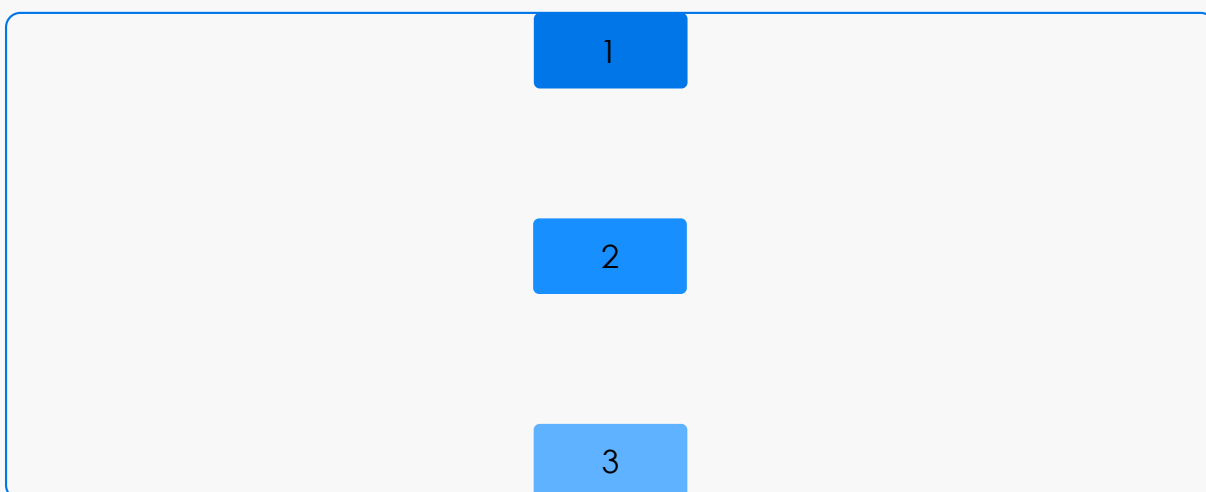
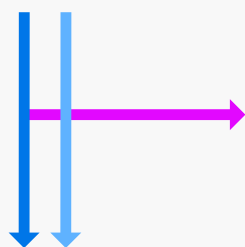
08. Justify-content & align-items

flex-direction: row; **justify-content:** flex-end; **align-items:** flex-end;



Elementy są w prawym dolnym rogu, ponieważ przesunęliśmy je na koniec na osi X (**justify-content**) oraz na koniec na osi Y (**align-items**).

flex-direction: column; **justify-content:** space-between; **align-items:** center;



Elementy są wyśrodkowane na osi X (**align-items**) oraz mają przestrzeń między sobą na osi Y (**justify-content**).