CSS. flexbox

CSS. flexbox

Специфікація CSS Flexible Box Layout Module (Flexbox) покликана кардинально змінити ситуацію в кращу сторону при вирішенні величезної кількості завдань при верстці.

Flexbox дозволяє контролювати розмір, порядок і вирівнювання елементів по декількох осях, розподіл вільного місця між елементами і багато іншого.

CSS. flexbox

Основні переваги flexbox

Всі блоки дуже легко робляться "гумовим", що вже випливає з назви "flex". Елементи можуть стискатися і розтягуватися за заданими правилами, займаючи потрібний простір.

Вирівнювання по вертикалі і горизонталі, базової лінії тексту працює дуже добре.

Розташування елементів в html не має вирішального значення. Його можна поміняти в CSS. Це особливо важливо для деяких аспектів responsive верстки.

Елементи можуть автоматично вибудовуватися в кілька рядків / стовпців, займаючи весь наданий простір.

Безліч мов в світі використовують написання справа наліво rtl (right-to-left), на відміну від звичного нам ltr (left-to-right). Flexbox адаптований для цього. У ньому є поняття початку і кінця, а не право і ліво. Тобто в браузерах з локаллю rtl всі елементи будуть автоматично розташовані в реверсному порядку.

Синтаксис CSS правил дуже простий і освоюється досить швидко.

CSS. flexbox

Flexbox визначає набір CSS властивостей для контейнера (**flex-контейнер**) і його дочірніх елементів (**flex-блоків**).

Перше, що потрібно зробити - це вказати контейнеру **display: flex** або **display: inline-flex.**

HTML

```
<div class="my-flex-container">
  <div class="my-flex-block">item1</div>
  <div class="my-flex-block">item2</div>
  <div class="my-flex-block">item3</div>
</div>
```

CSS

```
.my-flex-container{
  display: flex;
}
```

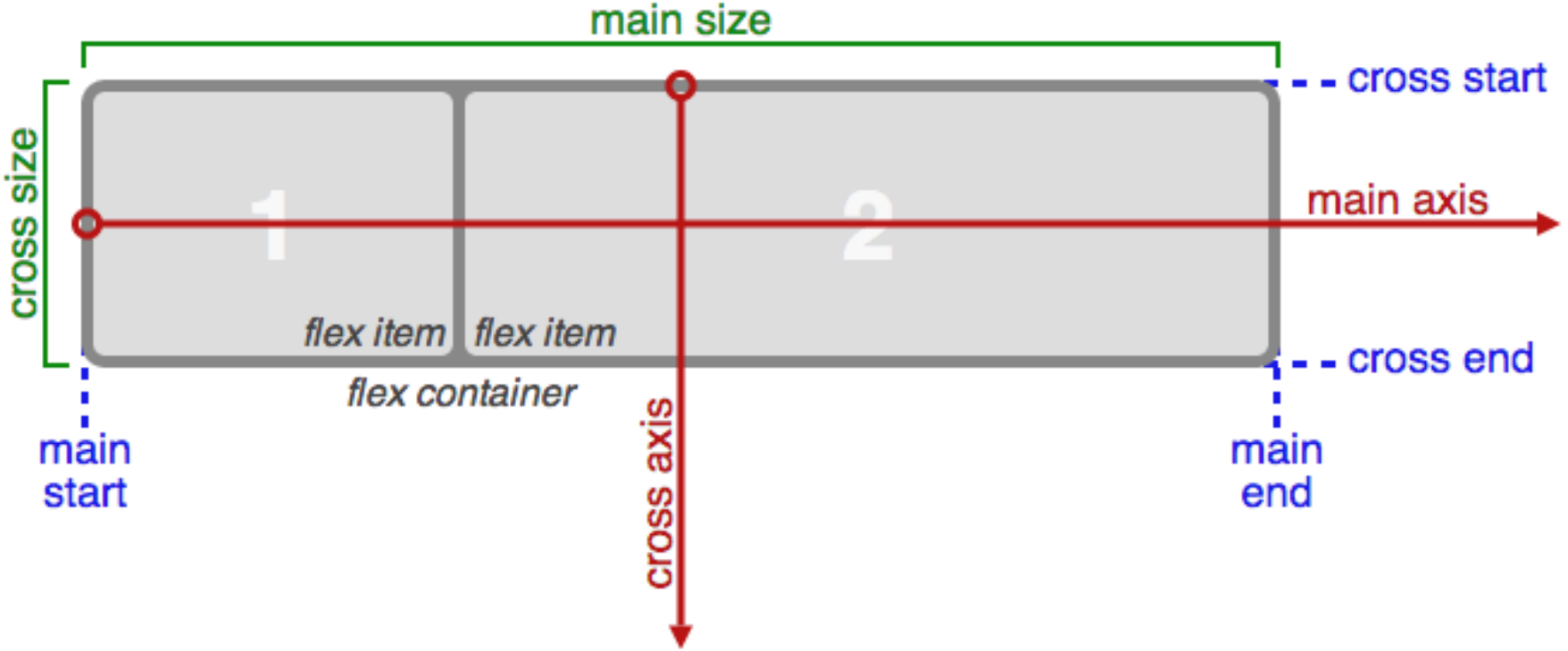
item1item2item3

CSS. flexbox

Одними з основних понять в flexbox є осі.

Головною віссю flex-контейнера є напрям, відповідно до якого розташовуються всі його дочірні елементи.
Поперечною віссю називається напрямок, перпендикулярний головної осі.

Головна вісь в ltr локалі за замовчуванням розташовується зліва направо. Поперечна - зверху вниз. Напрямок головної осі flex-контейнера можна задавати, використовуючи базову css властивість flex-direction.



CSS. flexbox

flex-direction - напрямок головної осі

Доступні значення flex-direction:

row (значення за замовчуванням): зліва направо (в rtl справа наліво)
row-reverse: справа наліво (в rtl зліва направо)
column: зверху вниз
column-reverse: від низу до верху

CSS. flexbox

justify-content - вирівнювання по головній осі.

Доступні значення justify-content:

flex-start (значення за замовчуванням): блоки притиснуті до початку головної осі

flex-end: блоки притиснуті до кінця головної осі

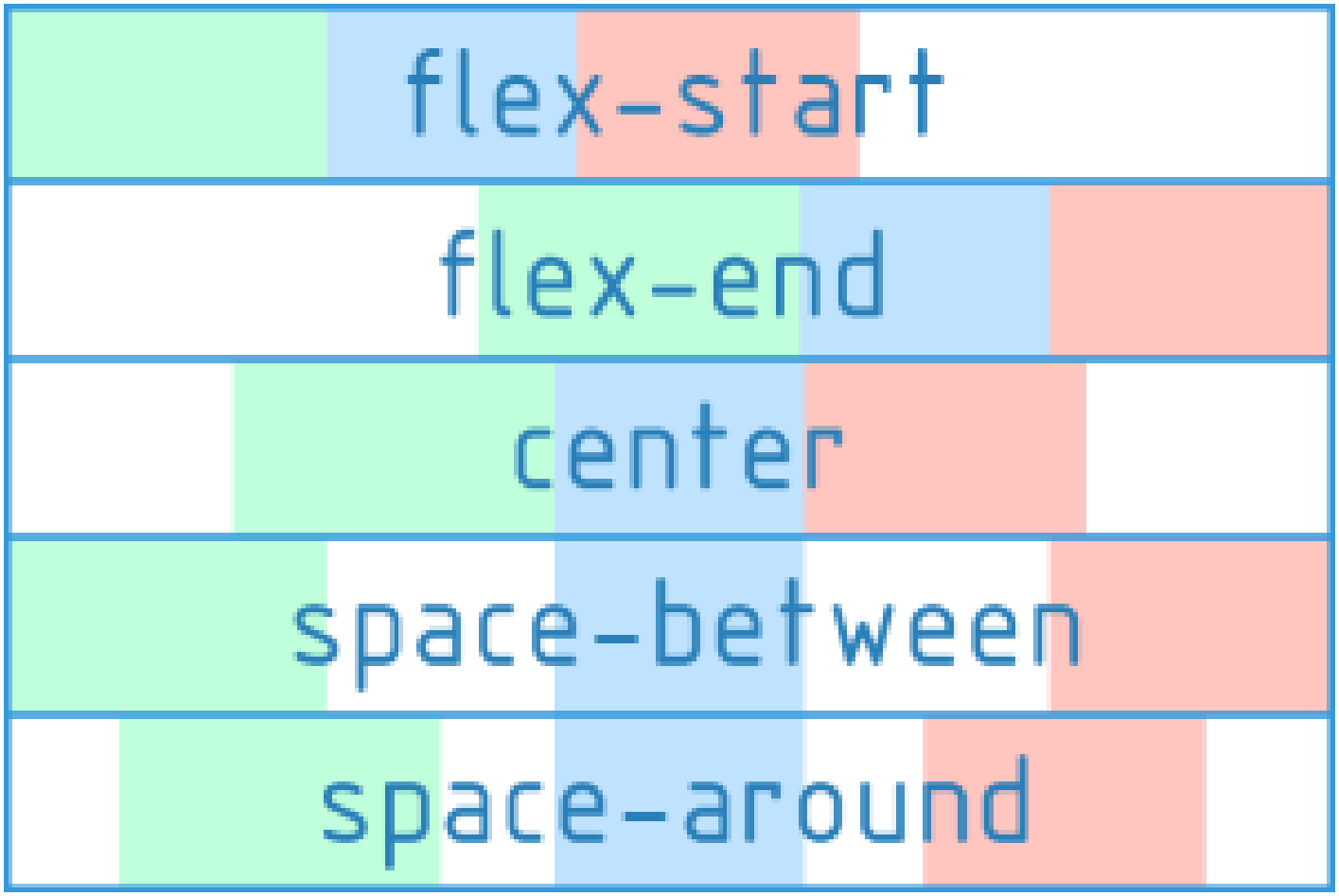
center: блоки розташовуються в центрі головної осі

space-between: перший блок розташовується на початку головної осі, останній блок - в кінці, всі інші блоки рівномірно розподілені в останньому просторі.

space-around: всі блоки рівномірно розподілені уздовж головної осі, розділяючи весь вільний простір порівну.

CSS. flexbox

justify-content - вирівнювання по головній осі.



CSS. flexbox

align-items - вирівнювання по поперечній осі.

flex-start: блоки притиснуті до початку поперечної осі

flex-end: блоки притиснуті до кінця поперечної осі

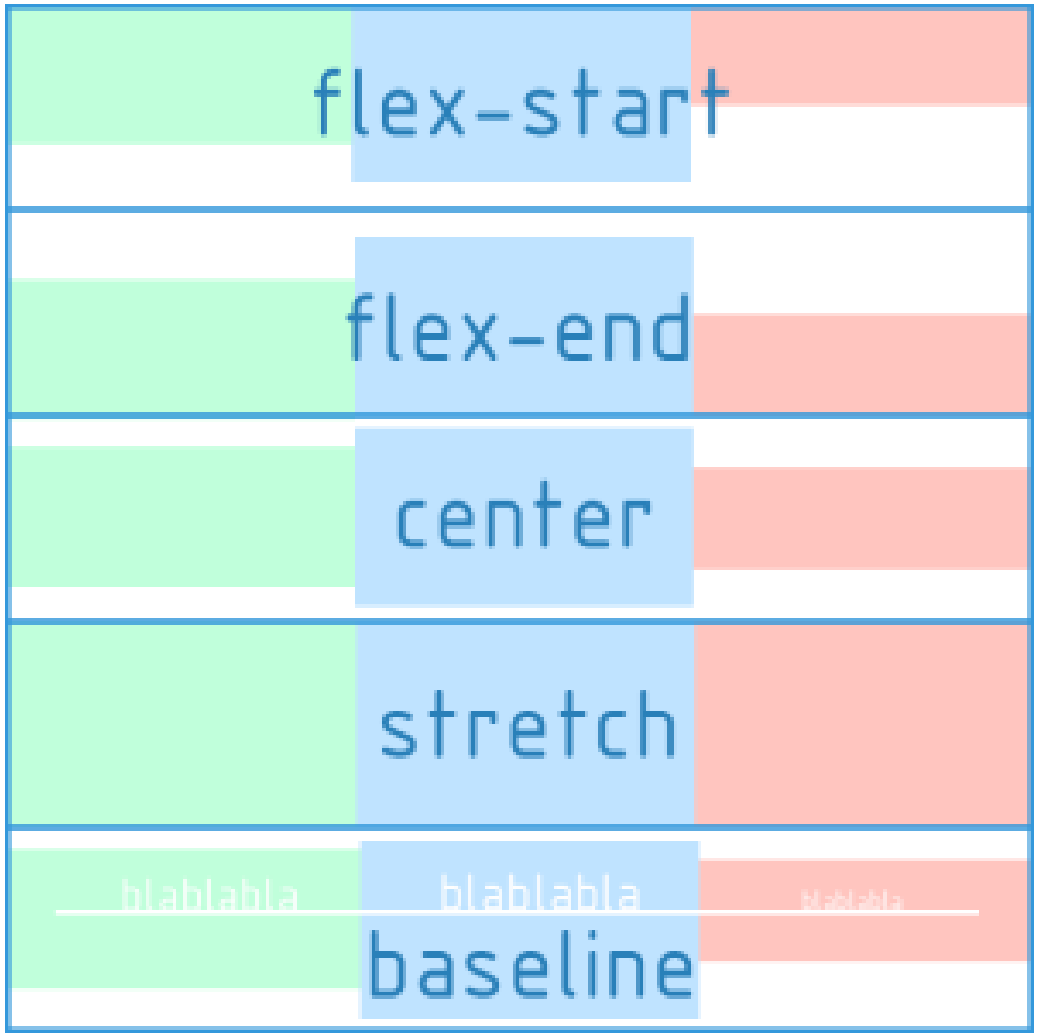
center: блоки розташовуються в центрі поперечної осі

baseline: блоки вирівняні по їх baseline

stretch (значення за замовчуванням): блоки розтягнуті, займаючи все доступне місце по поперечній осі, при цьому все-таки враховуються min-width / max-width, якщо такі задані.

CSS. flexbox

align-items - вирівнювання по поперечній осі.



CSS. flexbox

CSS властивості **flex-direction, justify-content, align-items** повинні застосовуватися безпосередньо до flex-контейнеру, а не до його дочірніх елементів.

Демо основних властивостей flex-контейнера

CSS. flexbox

Багаторядкова організація блоків всередині flex-контейнера.
flex-wrap

Всі попередні приклади були побудовані з врахуванням однорядкового розташування блоків.
За замовчуванням flex-контейнер завжди буде мати блоки всередині себе в одну лінію. Однак, специфікацією також підтримується багаторядковий режим.
За багаторядковість всередині flex-контейнера відповідає CSS властивість **flex-wrap**.

Доступні значення flex-wrap:

nowrap (значення за замовчуванням): блоки розташовані в одну лінію зліва направо (в rtl справа наліво)
wrap: блоки розташовані в декілька горизонтальних рядів (якщо не поміщаються в один ряд). Вони слідують один за одним зліва направо (в rtl справа наліво)
wrap-reverse: теж що і wrap, але блоки розташовуються в зворотному порядку.

CSS. flexbox

flex-flow - зручне скорочення для flex-direction + flex-wrap

По суті, flex-flow надає можливість в одну властивість описати напрямок головної і багаторядковість поперечної осі.
За замовчуванням flex-flow: row nowrap.

flex-flow: <'flex-direction'> || <'Flex-wrap'>

```
/* тобто ... */
.my-flex-block{
    flex-direcrion:column;
    flex-wrap: wrap;
}
/* те саме, що ... */
.my-flex-block{
    flex-flow: column wrap;
}
```

CSS. flexbox

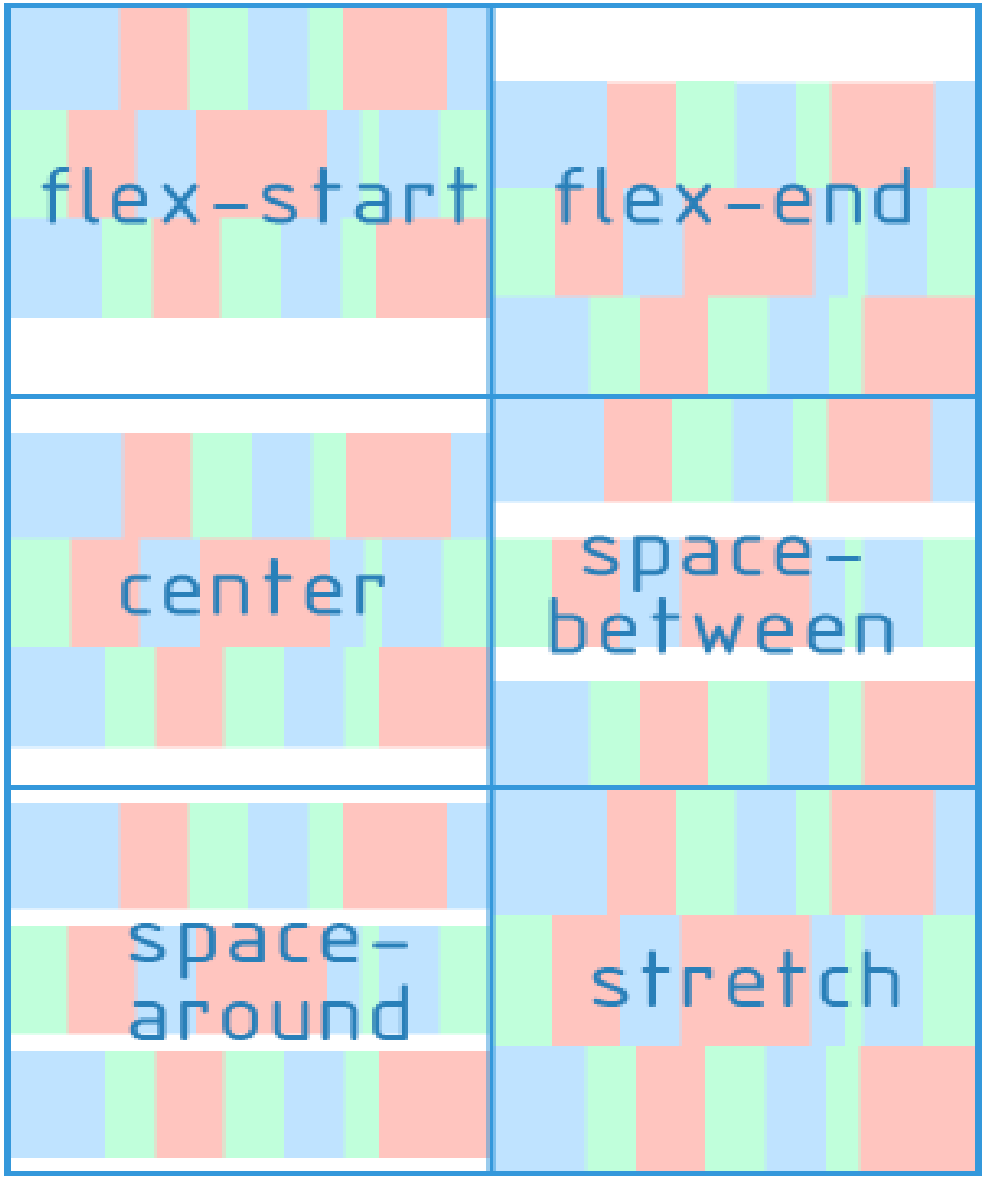
align-content визначає те, яким чином кілька рядів блоків будуть вирівняні по вертикалі і як вони поділять між собою весь простір flex-контейнера.

Важливо: align-content працює тільки в багаторядковому режимі (тобто в разі flex-wrap: wrap; або flex-wrap: wrap-reverse;)

flex-start: ряди блоків притиснуті до початку flex-контейнера.
flex-end: ряди блоків притиснуті до кінця flex-контейнера
center: ряди блоків знаходяться в центрі flex-контейнера
space-between: перший ряд блоків розташовується на початку flex-контейнера, останній ряд блоків блок - в кінці, всі інші ряди рівномірно розподілені в решті простору.
space-around: ряди блоків рівномірно розподілені від початку до кінця flex-контейнера, розділяючи весь вільний простір порівну.
stretch (значення за замовчуванням): Ряди блоків розтягнуті, щоб зайняти весь наявний простір.

CSS. flexbox

align-content визначає те, яким чином кілька рядів блоків будуть вирівняні по вертикалі і як вони поділять між собою весь простір flex-контейнера.



CSS. flexbox

CSS властивості flex-wrap і align-content повинні застосовуватися безпосередньо до flex-контейнеру, а не до його дочірніх елементів.

Демо властивостей багаторядковості в flex

CSS. flexbox

CSS правила для дочірніх елементів flex-контейнера (flex-блоків)

flex-basis - базовий розмір окремо взятого flex-блоку

Задає початковий розмір по головній осі для flex-блоку до того, як до нього будуть застосовані перетворення, засновані на інших flex-факторах.

Може бути заданий в будь-яких одиницях вимірювання довжини (px, em,%, ...) або auto (за замовчуванням). Якщо заданий як auto - за основу беруться розміри блоку (width, height), які, в свою чергу, можуть залежати від розміру контенту, якщо не вказані явно.

CSS. flexbox

flex-grow - "жадібність" окремо взятого flex-блоку

Визначає те, на скільки окремий flex-блок може бути більшим сусідніх елементів, якщо це необхідно.
flex-grow приймає безрозмірне значення (за замовчуванням 0)

Приклад 1:

Якщо всі flex-блоки всередині flex-контейнера мають flex-grow: 1, то вони будуть однакового розміру
Якщо один з них має flex-grow: 2, то він буде в 2 рази більшим, ніж всі інші

Приклад 2:

Якщо всі flex-блоки всередині flex-контейнера мають flex-grow: 3, то вони будуть однакового розміру
Якщо один з них має flex-grow: 12, то він буде в 4 рази більше, ніж всі інші

Тобто абсолютне значення flex-grow не визначає точну ширину. Воно визначає його ступінь "жадібності" по відношенню до інших flex-блоків того ж рівня.

CSS. flexbox

flex-shrink - фактор «зжимання" окремо взятого flex-блоку.
Визначає, наскільки flex-блок буде зменшуватися щодо сусідніх елементів всередині flex-контейнера в разі нестачі вільного місця.
За замовчуванням дорівнює 1.

flex - короткий запис для властивостей flex-grow, flex-shrink і flex-basis
flex: none | [<'Flex-grow'> <'flex-shrink'>? || <'Flex-basis'>]

```
/* тобто ... */
.my-flex-block{
  flex-grow:12;
  flex-shrink:3;
  flex-basis: 30em;
}
/* те саме, що ... */
.my-flex-block{
  flex: 12 3 30em;
}
```

Демо для flex-grow, flex-shrink і flex-basis

CSS. flexbox

align-self - вирівнювання окремо взятого flex-блоку по поперечній осі.

Доступні значення align-self (ті ж 5 варіантів, що і для align-items)

flex-start: flex-блок притиснутий до початку поперечної осі
flex-end: flex-блок притиснутий до кінця поперечної осі
center: flex-блок розташовується в центрі поперечної осі
baseline: flex-блок вирівнюється по baseline
stretch (значення за замовчуванням): flex-блок розтягнутий, щоб зайняти весь доступний простір по поперечній осі, при цьому враховуються min-width / max-width, якщо такі задані.

CSS. flexbox

order - порядок виводу окремо взятого flex-блоку всередині flex-контейнера.

За замовчуванням всі блоки будуть слідувати один за одним в порядку, заданому в html. Однак цей порядок можна змінити за допомогою властивості **order**. Вона задається цілим числом і за замовчуванням дорівнює 0.

```
<div class="my-flex-container">
<div class="my-flex-block" style="order: 5" >item1</div>
<div class="my-flex-block" style="order: 10">item2</div>
<div class="my-flex-block" style="order: 5" >item3</div>
<div class="my-flex-block" style="order: 5" >item4</div>
<div class="my-flex-block" style="order: 0" >item5</div>
</div>
```

В даному випадку, блоки будуть слідувати один за іншим уздовж головної осі в наступному порядку: item5, item1, item3, item4, item2

Демо для align-self і order

CSS. flexbox

margin: auto по вертикалі.

Звичне всім вирівнювання по горизонталі через margin: auto тут працює і для вертикалі!

```
.my-flex-container {
  display: flex;
  height: 300px;
}
.my-flex-block {
  width: 100px;
  height: 100px;
  margin: auto;
}
```

CSS. flexbox

Не забувайте про **margin**. Вони враховуються при установці вирівнювання по осях. Також важливо пам'ятати, що margin-и в flexbox не «схлопуються», як це відбувається в звичайному потоці.

Значення **float** у flex-блоків не враховується і не має значення.

CSS. flexbox

https://tproger.ru/translations/how-css-flexbox-works/

https://developer.mozilla.org/uk/docs/Web/CSS/CSS_Flexible_Box_Layout/Using_CSS_flexible_boxes

http://flexbox.help/