

# Flexbox Layout - Flexible Box Layout Module

направленный на улучшение работы с элементами:

- выравнивание
- изменения направление и порядка их в контейнере( даже когда они динамические или неизвестного размера
- изменение ширины или высоты ее потомков, чтобы заполнять свободное пространство при различных размерах экрана.
- растягивание элементов для заполнения свободного места или сжатие их, чтобы предотвратить выход за границы.
- расположение элементов в html не имеет решающего значения. Его можно поменять в CSS.

# display: flex | inline-flex; Свойства flex-контейнера

Применяется к: родительскому элементу flex-контейнера.

Свойство display: flex делает элемент flex-контейнером, прямые потомки (первого уровня вложенности) превращаются во flex-элементы.

# HTML

01. <div class="container">

02.     <div class="item">1</div>

03.     <div class="item">2</div>

04.     <div class="item">3</div>

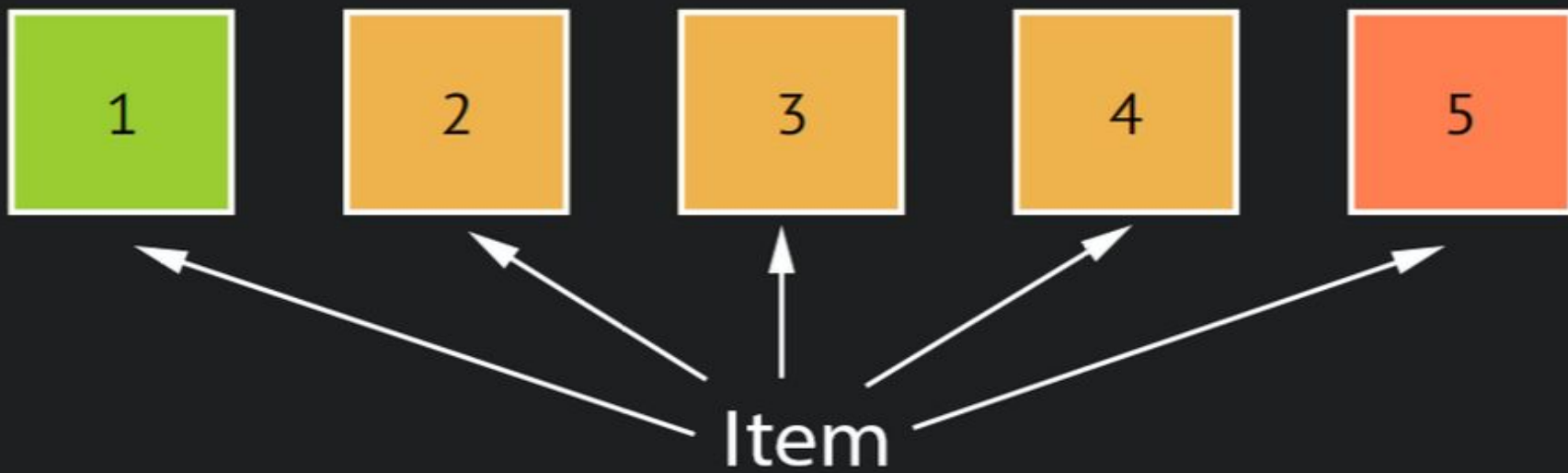
05.     <div class="item">4</div>

06.     <div class="item">5</div>

07. </div>

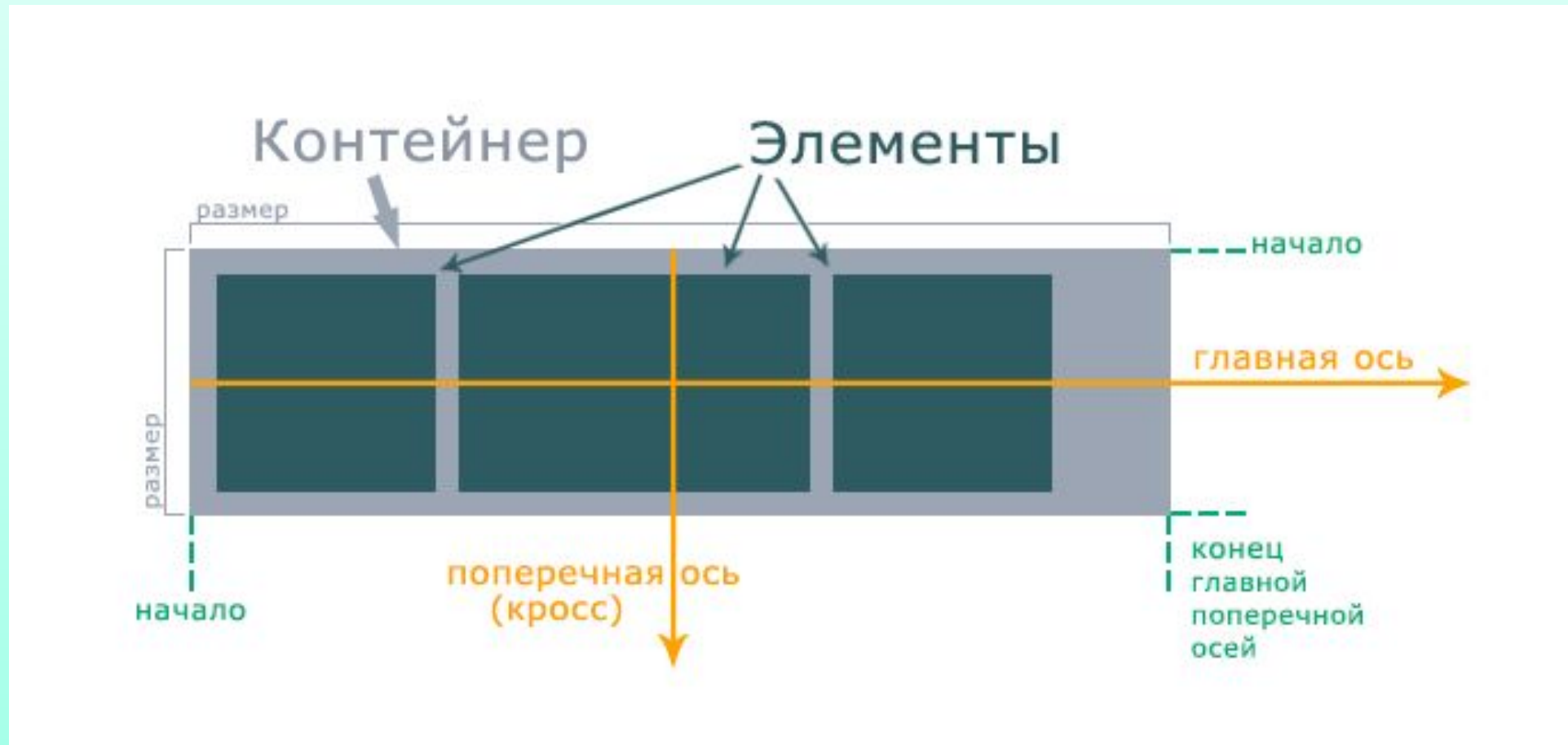
# Субъекты

Container



# ОСНОВЫ

Если обычный лейаут основывается на направлениях потоков блочных и инлайн-элементов, то flex-лейаут основывается на «направлениях flex-потока».



**FlexBox** состоит из контейнера и его дочерних элементов (items).

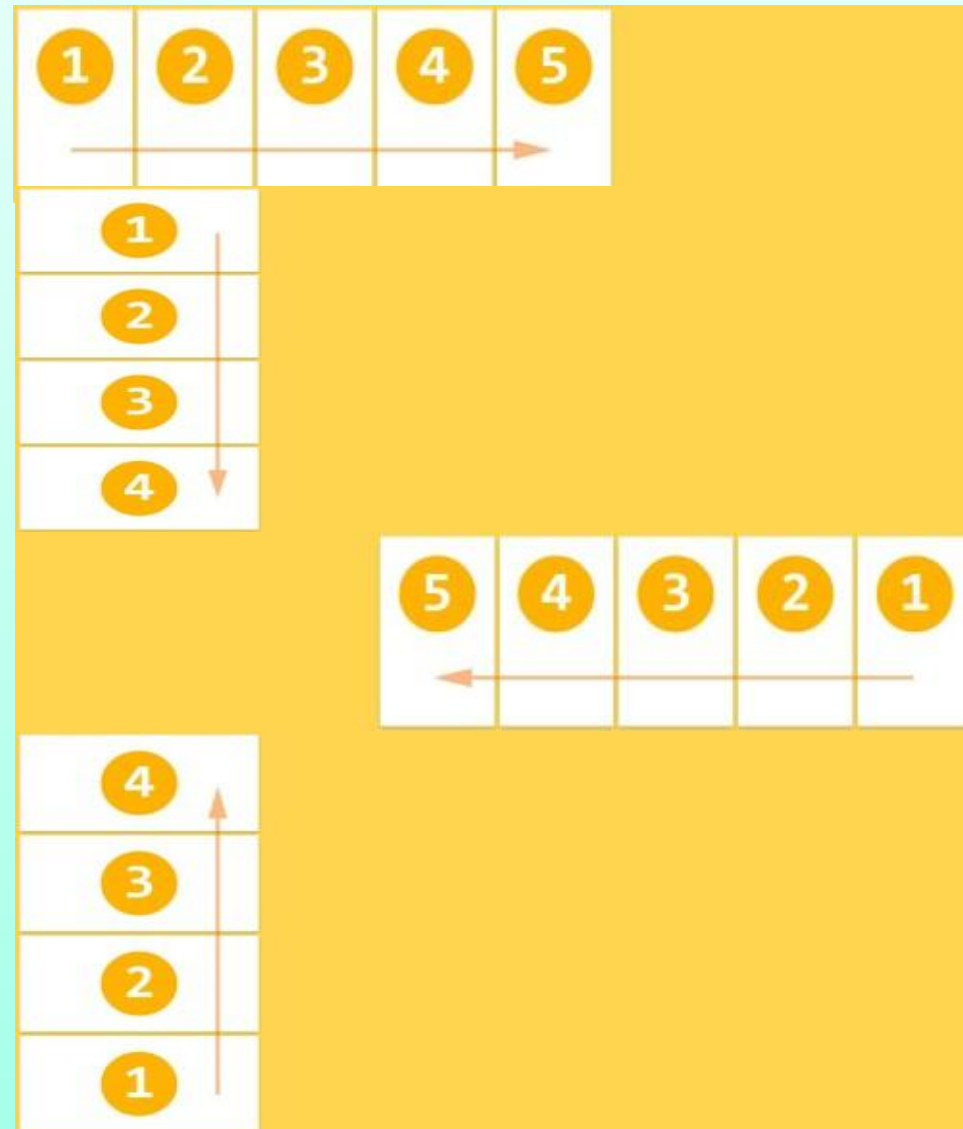
- **Главная ось** - главное направление движения элементов внутри контейнера. Направление главной оси можно изменить с помощью свойства [flex-direction](#). При смене осей, меняются только направления движения блоков внутри, а начало, конец и размер контейнера остаются прежними.
- **Начало и конец главной оси** - элементы располагаются от начала и до конца контейнера.
- **Поперечная ось** - направление движения элементов, когда они не умещаются в контейнер по направлению главной оси. Поперечная ось всегда перпендикулярна главной.
- **Начало и конец поперечной оси** - по поперечной оси заполняются ряды от начала и до конца контейнера.
- **Размер (главный и поперечный)** - базовая величина по которой высчитывается ширина или высота внутренних элементов, если размер указан не точно (указан в процентах или не указан вообще, а элемент должен растянуться или сжаться).

# Flex-direction

Свойства flex-контейнера

Изменяет направление главной оси контейнера. Поперечная ось меняется соответственно.

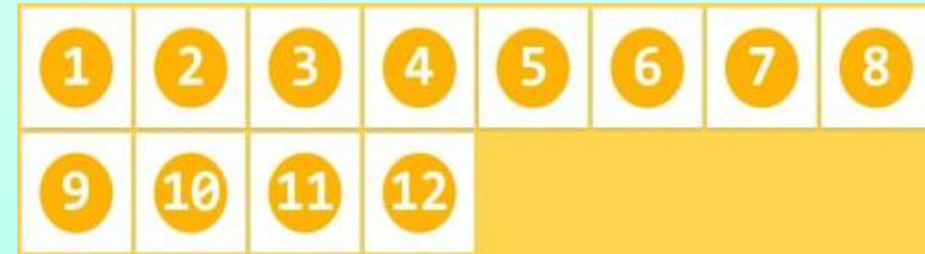
- **row (default)** - направление элементов слева направо (→)
- **column** - направление элементов сверху вниз (↓)
- **row-reverse** - направление элементов справа налево (←)
- **column-reverse** - направление элементов снизу вверх (↑)



# Flex-wrap: Свойства flex-контейнера

Управляет переносом непомещающихся в контейнер элементов.

- **nowrap (default)** - вложенные элементы располагаются в один ряд (при `direction=row`) или в одну колонку (при `direction=column`) независимо от того помещаются они в контейнер или нет.
- **wrap** - включает перенос элементов на следующий ряд, если они не помещаются в контейнер. Так включается движение элементов по поперечной оси.
- **wrap-reverse** - тоже что `wrap` только перенос будет не вниз, а вверх (в обратном направлении).





# Flex-flow: flex-direction flex-wrap; Свойства flex-контейнера

Это свойство – сокращённая запись для свойств `flex-direction` и `flex-wrap`.  
Также можно указать только одно из значений:

```
.flex-container {  
    flex-flow: <flex-direction> || <flex-wrap>;  
  
    flex-flow: <flex-direction> && <flex-wrap>;  
}
```

**Значение по умолчанию:** `row nowrap`;

# Justify-content

Свойства flex-контейнера

Выравнивает элементы по основной оси.

- **flex-start (default)** - элементы будут идти с начала (в конце может остаться место).
- **flex-end** - элементы выравниваются по концу (место останется в начале)
- **center** - по центру (место останется слева и права)
- **space-between** - крайние элементы прижимаются к краям (место между элементами распределяется равномерно)
- **space-around** - свободное пространство равномерно распределяется между элементами (крайние элементы не прижимаются к краям).



# Align-content

Свойства flex-контейнера

Выравнивает ряды, в которых находятся элементы по поперечной оси.  
То же что justify-content только для поперечной оси.

- **stretch (default)** - линии растягиваются заполняя строку полностью
- **flex-start** - линии будут идти с начала (в конце может остаться место).
- **flex-end** - линии выравниваются по концу (место останется в начале)



# Align-content

Свойства flex-контейнера (продолжение)

- **center** - по центру (место останется слева и права)
- **space-between** - крайние линии прижимаются к краям (место между линиями распределяется равномерно)
- **space-around** - свободное пространство равномерно распределяется между элементами (крайние элементы не прижимаются к краям).



# align-items

Свойства flex-контейнера

Выравнивает элементы по поперечной оси внутри ряда (невидимой строки).

- **stretch (default)** - элементы растягиваются заполняя строку полностью
- **flex-start** - элементы прижимаются к началу строки
- **flex-end** - элементы прижимаются к концу строки
- **center** - элементы выравниваются по центру строки
- **baseline** - элементы выравниваются по базовой линии текста



# Flex-grow Свойства flex-элемента

Задаёт коэффициент увеличения элемента при наличии свободного места в контейнере. По умолчанию `flex-grow: 0` т.е. никакой из элементов не должен увеличиваться и заполнять свободное место в контейнере

**Значение по умолчанию: `flex-grow: 0`;**



# Flex-shrink

Свойства flex-элемента

Задаёт коэффициент уменьшения элемента. Свойство противоположное flex-grow и определяет как элемент должен сжиматься, если в контейнере не остается свободного места. Т.е. свойство начинает работать, когда сумма размеров всех элементов больше чем размер контейнера.



**Значение по умолчанию: flex-shrink: 1;**

# Flex-basis

Свойства flex-элемента

Устанавливает базовую ширину элемента - ширину до того как будут высчитаны остальные условия влияющие на ширину элемента. Значение можно указать в px, %, em, rem и т.д. Итоговая ширина будет зависеть от базовой ширины и значений flex-grow, flex-shrink и контента внутри блока. В режиме auto элемент получает базовую ширину относительно контента внутри него.



**Значение по умолчанию: flex-basis: auto;**



# Flex: flex-grow, flex-shrink, flex-basis; Свойства flex-элемента

Это свойство – сокращённая запись для свойств `flex-direction` и `flex-wrap`.  
Также можно указать только одно из значений:

```
.flex-item { flex: <flex-grow> || <flex-shrink> || <flex-basis>;  
}
```

**Значение по умолчанию:** 0 1 auto;

# Align-self

Свойства flex-элемента

Позволяет изменить свойство align-items, только для отдельного элемента.

**stretch** - элемент растягиваются заполняя строку полностью

**flex-start** - элемент прижимаются к началу строки

**flex-end** - элемент прижимаются к концу строки

**center** - элемент выравниваются по центру строки

**baseline** - элемент выравниваются по базовой линии текста



Значение по умолчанию: align-self: auto;

# Order <integer>

Позволяет менять порядок (позицию, положение) элемента в общем ряду.



**Значение по умолчанию: order: 0;**

# Вендорные префиксы

Вендорный префикс	Производитель	Браузер	Браузерный движок
<a href="#">-o-</a> , <a href="#">-op-</a> , <a href="#">-xv-</a>	Opera Software	Opera	Presto
<a href="#">-moz-</a>	проект Mozilla	Firefox, SeaMonkey, Camino и др.	Gecko
<a href="#">-ms-</a>	Microsoft	Internet Explorer 8	Trident
<a href="#">-khtml-</a>	проект KDE	Safari до версии 3, Konqueror и др.	KHTML послужил основой для WebKit
<a href="#">-webkit-</a>	Apple	Safari 3+, Google Chrome и др.	WebKit

# Кроссбраузерный flexbox

[https://wp-kama.ru/id\\_8045/flex-v-css.html#flex-direction](https://wp-kama.ru/id_8045/flex-v-css.html#flex-direction)

<https://habrahabr.ru/post/329820/>

<https://webbeaver.ru/future/flexbox/>

<http://uirockstar.net/tool/cssflexboxgenerator.html>

```
display: -webkit-box;
```

```
display: -webkit-flex;
```

```
display: -ms-flexbox;
```

```
display: box;
```

```
display: flexbox;
```

```
display: flex;
```

```
-webkit-flex-wrap: wrap;
```

```
flex-wrap: wrap;
```

```
-webkit-flex-direction: row;
```

```
flex-direction: row;
```

```
-webkit-justify-content: space-between;
```

```
justify-content: space-between;
```