Flex/Grid

Flexbox

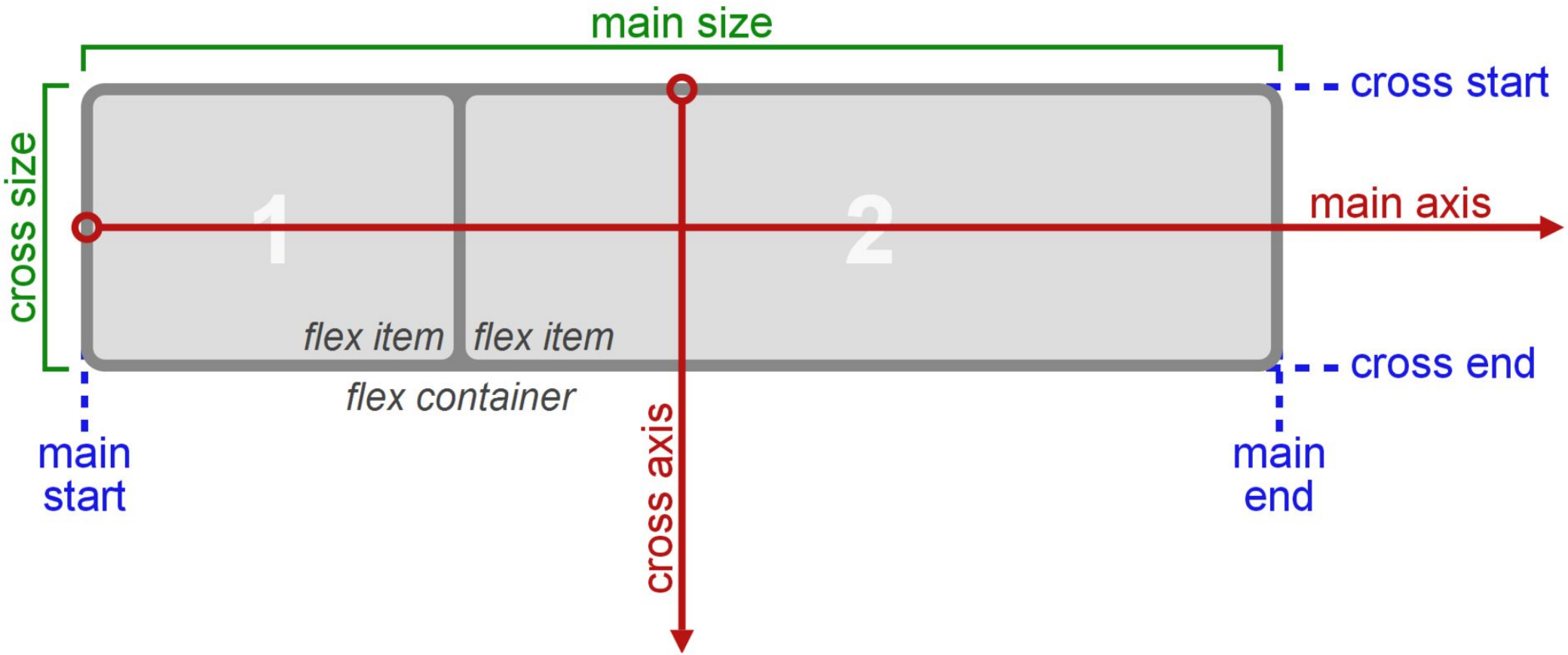
Flexbox состоит из **гибкого контейнера (flex container)** и **гибких элементов (flex items)**. Гибкие элементы могут выстраиваться в строку или столбик, а оставшееся свободное пространство распределяется между ними различными способами.

Flex-контейнер устанавливает новый гибкий контекст форматирования для его содержимого. Flex-контейнер не является блочным контейнером, поэтому для дочерних элементов не работают такие CSS-свойства, как *float*, *clear*, *vertical-align*. Также, на flex-контейнер не оказывают влияние свойства *column-*,* создающие колонки в тексте и псевдоэлементы *::first-line* и *::first-letter*.

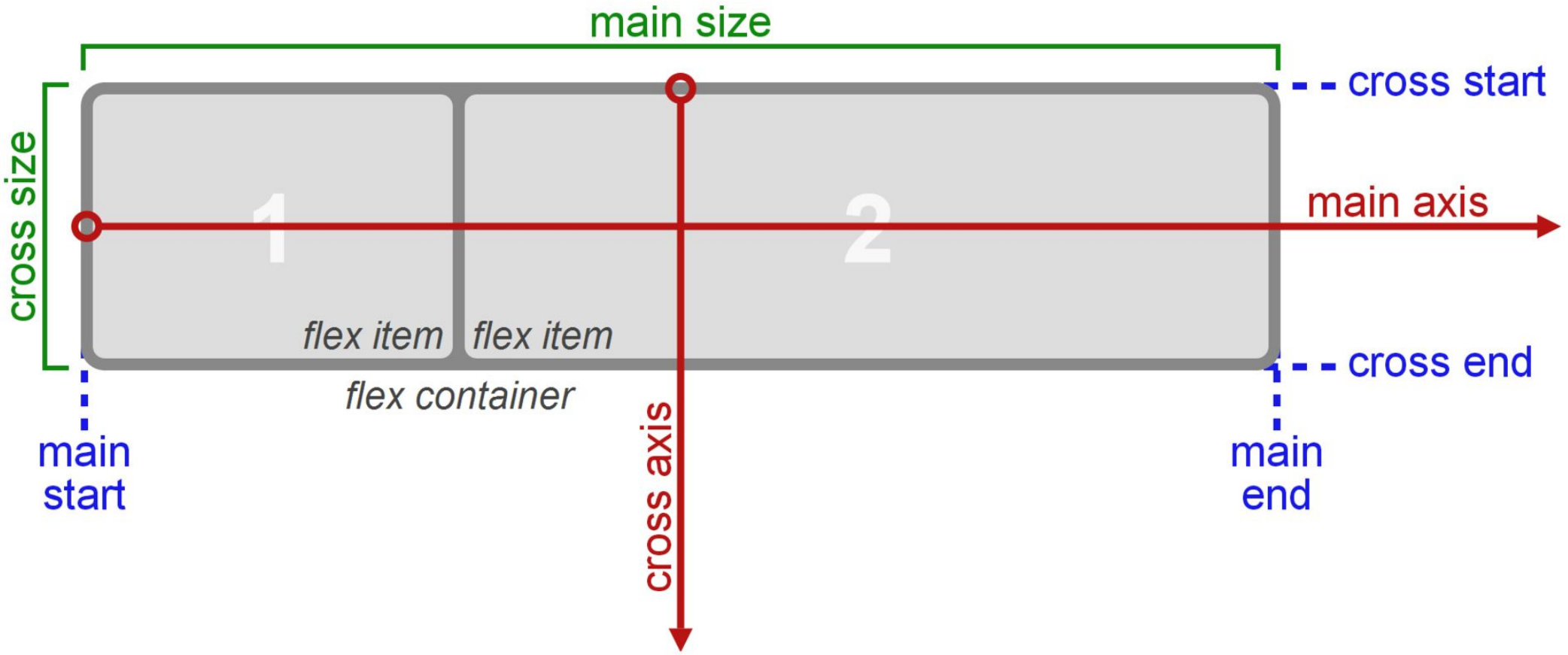
Flexbox

Модуль flexbox позволяет решать следующие задачи:

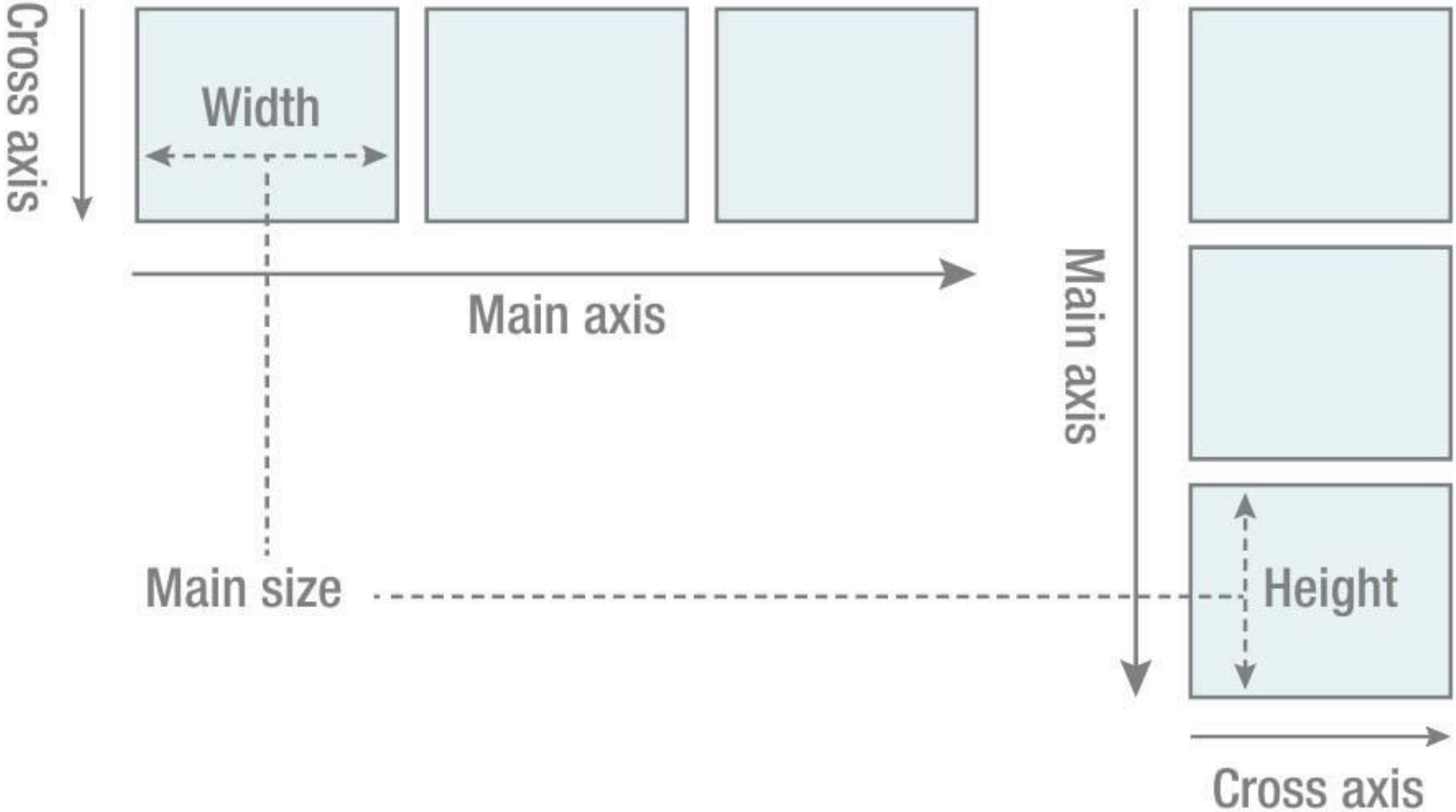
- Располагать элементы в одном из четырех направлений: **слева направо**, **справа налево**, **сверху вниз** или **снизу вверх**.
- Переопределять **порядок отображения** элементов.
- Автоматически определять **размеры элементов** таким образом, чтобы они вписывались в доступное пространство.
- Решать проблему с горизонтальным и вертикальным центрированием.
- Переносить элементы внутри контейнера, не допуская его переполнения.
- Создавать колонки одинаковой высоты.



Главная ось (main axis) — ось, вдоль которой выкладываются flex-элементы. Она простирается в основном измерении.
Main start и main end — линии, которые определяют начальную и конечную стороны flex-контейнера, относительно которых выкладываются flex-элементы (начиная с main start по направлению к main end).
Основной размер (main size) — ширина или высота flex-контейнера или flex-элементов, в зависимости от того, что из них находится в основном измерении, определяют основной размер flex-контейнера или flex-элемента.



Поперечная ось (cross axis) — ось, перпендикулярная главной оси. Она простирается в поперечном измерении.
Cross start и cross end — линии, которые определяют начальную и конечную стороны поперечной оси, относительно которых выкладываются flex-элементы.
Поперечный размер (cross size) — ширина или высота flex-контейнера или flex-элементов, в зависимости от того, что находится в поперечном измерении, являются их поперечным размером.


Cross axis
Width
Main axis
Main size
Main axis
Height
Cross axis

Flex-элементы

Flex-элементы — блоки, представляющие содержимое flex-контейнера в потоке. Flex-контейнер устанавливает новый контекст форматирования для своего содержимого, который обуславливает следующие особенности:

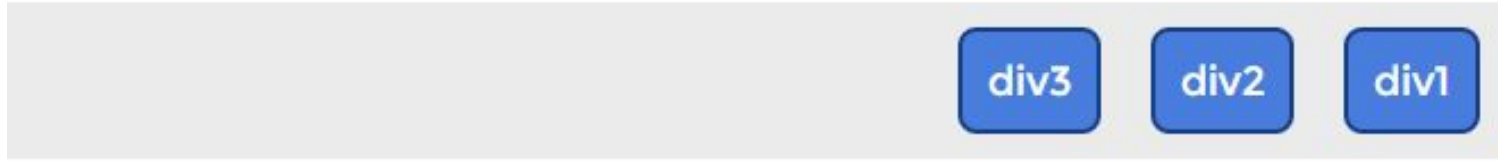
- Для flex-элементов блокируется их значение свойства *display*. Значение *display: inline-block;* и *display: table-cell;* вычисляется в *display: block;*.
- Пустое пространство между элементами исчезает: оно не становится своим собственным flex-элементом, даже если межэлементный текст обернут в анонимный flex-элемент. Для содержимого анонимного flex-элемента невозможно задать собственные стили, но оно будет наследовать их (например, параметры шрифта) от flex-контейнера.
- Абсолютно позиционированный flex-элемент не участвует в компоновке гибкого макета.
- **Поля margin соседних flex-элементов не схлопываются**.
- Процентные значения *margin* и *padding* вычисляются от внутреннего размера содержащего их блока.
- *margin: auto;* расширяются, поглощая дополнительное пространство в соответствующем измерении. Их можно использовать для выравнивания или раздвигания смежных flex-элементов.
- Автоматический минимальный размер flex-элементов по умолчанию является минимальным размером его содержимого, то есть *min-width: auto;*. Для контейнеров с прокруткой автоматический минимальный размер обычно равен нулю.

Порядок отображения flex-элементов и ориентация

*Содержимое flex-контейнера можно разложить в любом направлении и в любом порядке (переупорядочение flex-элементов внутри контейнера влияет **только на визуальный рендеринг**).

flex-direction	
Значения:	
row	Значение по умолчанию, слева направо (в rtl справа налево). Flex-элементы выкладываются в строку. Начало (*main-start*) и конец (*main-end*) направления главной оси соответствуют началу (*inline-start*) и концу (inline-end) оси строки (*inline-axis*).
row-reverse	Направление справа налево (в rtl слева направо). Flex-элементы выкладываются в строку относительно правого края контейнера (в rtl — левого).
column	Направление сверху вниз. Flex-элементы выкладываются в колонку.
column-reverse	Колонка с элементами в обратном порядке, снизу вверх.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

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




```
flex-direction: column;
```


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

Управление многострочностью flex-контейнера: flex-wrap

*По умолчанию flex-элементы укладываются в одну строку, вдоль главной оси. При переполнении они будут выходить за пределы ограничивающей рамки flex-контейнера. Свойство не наследуется.

flex-wrap	
Значения:	
nowrap	Значение по умолчанию. Flex-элементы не переносятся, а располагаются в одну линию слева направо (в rtl справа налево).
wrap	Flex-элементы переносятся, располагаясь в несколько горизонтальных рядов (если не помещаются в один ряд) в направлении слева направо (в rtl справа налево).
wrap-reverse	Flex-элементы переносятся на новые линии, располагаясь в обратном порядке слева-направо, при этом перенос происходит снизу вверх.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

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


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

Краткая запись направления и многострочности: flex-flow

*Свойство позволяет определить **направления главной и поперечной осей**, а также возможность переноса flex-элементов при необходимости на несколько строк. Представляет собой сокращённую запись свойств *flex-direction* и *flex-wrap*. Значение по умолчанию *flex-flow: row nowrap;*. свойство не наследуется.

flex-flow		
Значения:		
направление		Указывает направление главной оси. Значение по умолчанию *row.*
многострочность		Задаёт многострочность поперечной оси. Значение по умолчанию *nowrap*.
initial		Устанавливает значение свойства в значение по умолчанию.
inherit		Наследует значение свойства от родительского элемента.

Порядок отображения flex-элементов: order

*Свойство определяет порядок, в котором flex-элементы отображаются и располагаются внутри flex-контейнера. Применяется к flex-элементам. Свойство не наследуется.

*Первоначально все flex-элементы имеют *order: 0;*. При указании значения от -1 для элемента он перемещается в начало сроки, значение 1 — в конец. Если несколько flex-элементов имеют одинаковое значение *order*, они будут отображаться в соответствии с исходным порядком.

order		
Значения:		
число		Свойство задается целым числом, отвечающим за порядок отображения flex-элементов. Значение по умолчанию *0.*
initial		Устанавливает значение свойства в значение по умолчанию.
inherit		Наследует значение свойства от родительского элемента.

Гибкость flex-элементов

Определяющим аспектом гибкого макета является возможность «сгибать» flex-элементы, изменяя их ширину/высоту (в зависимости от того, какой размер находится на главной оси), чтобы заполнить доступное пространство в основном измерении. Это делается с помощью свойства flex. Flex-контейнер распределяет свободное пространство между своими дочерними элементами (пропорционально их коэффициенту flex-grow) для заполнения контейнера или сжимает их (пропорционально их коэффициенту flex-shrink), чтобы предотвратить переполнение.

*Flex-элемент будет полностью «негибок», если его значения flex-grow и flex-shrink равны нулю, и «гибкий» в противном случае.

Свойство flex

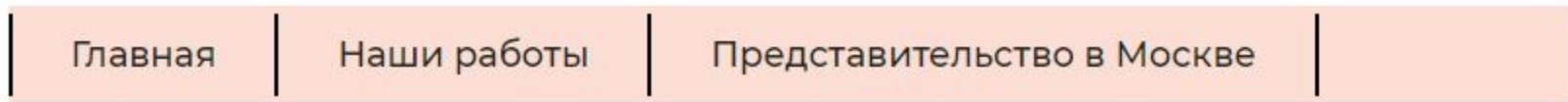
*Свойство является сокращённой записью свойств *flex-grow*, *flex-shrink* и *flex-basis*. Значение по умолчанию: *flex: 0 1 auto;*. Можно указывать как одно, так и все три значения свойств. Свойство не наследуется.

flex	
Значения:	
коэффициент растяжения	Коэффициент увеличения ширины flex-элемента относительно других flex-элементов.
коэффициент сужения	Коэффициент уменьшения ширины flex-элемента относительно других flex-элементов.
базовая ширина	Базовая ширина flex-элемента.
auto	Эквивалентно *flex: 1 1 auto;*.
none	Эквивалентно *flex: 0 0 auto;*.
initial	Устанавливает значение свойства в значение по умолчанию. Эквивалентно *flex: 0 1 auto;*.
inherit	Наследует значение свойства от родительского элемента.

Коэффициент роста: flex-grow

*Свойство определяет коэффициент роста одного flex-элемента относительно других flex-элементов в flex-контейнере при распределении положительного свободного пространства. Если сумма значений *flex-grow* flex-элементов в строке меньше 1, они занимают менее 100% свободного пространства. Свойство не наследуется.

flex-grow		
Значения:		
число		Положительное целое или дробное число, устанавливающее коэффициент роста flex-элемента. Значение по умолчанию *0*.
initial		Устанавливает значение свойства в значение по умолчанию.
inherit		Наследует значение свойства от родительского элемента.




```
li {flex-grow: 0;}
```


```
li {flex-grow: 1;}
```


```
li:last-child {flex-grow: 1;}
```

Коэффициент сжатия: flex-shrink

*Свойство указывает коэффициент сжатия flex-элемента относительно других flex-элементов при распределении отрицательного свободного пространства. Умножается на базовый размер *flex-basis*. Отрицательное пространство распределяется пропорционально тому, насколько элемент может сжаться, поэтому, например, маленький flex-элемент не уменьшится до нуля, пока не будет заметно уменьшен flex-элемент большего размера. Свойство не наследуется.

flex-shrink		
Значения:		
число		Положительное целое или дробное число, устанавливающее коэффициент роста flex-элемента. Значение по умолчанию *1*.
initial		Устанавливает значение свойства в значение по умолчанию.
inherit		Наследует значение свойства от родительского элемента.


div1
div2
div3
style="flex-
shrink: 2"
div4


span1
span2
span3
style="flex-
shrink: 3"
span4
span5

Базовый размер flex-basis

*Свойство устанавливает начальный основной размер flex-элемента до распределения свободного пространства в соответствии с коэффициентами гибкости. Для всех значений, кроме *auto* и *content*, базовый гибкий размер определяется так же, как *width* в горизонтальных режимах записи. Процентные значения определяются относительно размера flex-контейнера, а если размер не задан, используемым значением для *flex-basis* являются размеры его содержимого. Не наследуется.

flex-basis	
Значения:	
auto	Значение по умолчанию. Элемент получает базовый размер, соответствующий размеру его содержимого (если он не задан явно).
content	Определяет базовый размер в зависимости от содержимого flex-элемента.
длина	Базовый размер определяется так же, как для ширины и высоты. Задается в единицах длины.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

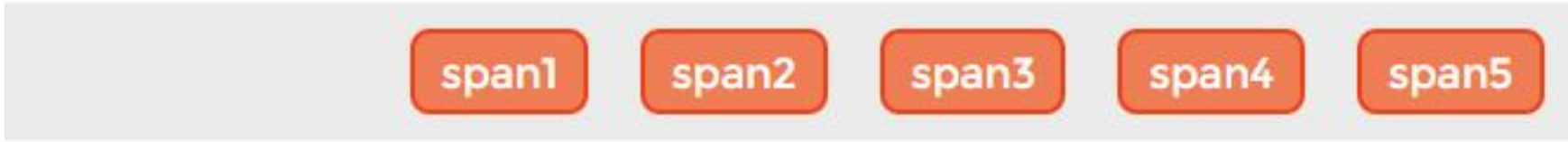
Выравнивание по главной оси: justify-content

*Свойство выравнивает flex-элементы по главной оси flex-контейнера, распределяя свободное пространство, незанятое flex-элементами. Когда элемент преобразуется в flex-контейнер, flex-элементы по умолчанию сгруппированы вместе (если для них не заданы поля *margin*). Промежутки добавляются после расчета значений *margin* и *flex-grow*. Если какие-либо элементы имеют ненулевое значение *flex-grow* или *margin: auto;*, свойство не будет оказывать влияния. Свойство не наследуется.

justify-content	
Значения:	
flex-start	Значение по умолчанию. Flex-элементы выкладываются в направлении, идущем от начальной линии flex-контейнера.
flex-end	Flex-элементы выкладываются в направлении, идущем от конечной линии flex-контейнера.
content	Flex-элементы выравниваются по центру flex-контейнера.
space-between	Flex-элементы равномерно распределяются по линии. Первый flex-элемент помещается вровень с краем начальной линии, последний flex-элемент — вровень с краем конечной линии, а остальные flex-элементы на линии распределяются так, чтобы расстояние между любыми двумя соседними элементами было одинаковым. Если оставшееся свободное пространство отрицательно или в строке присутствует только один flex-элемент, это значение идентично параметру *flex-start*.
space-around	Flex-элементы на линии распределяются так, чтобы расстояние между любыми двумя смежными flex-элементами было одинаковым, а расстояние между первым/последним flex-элементами и краями flex-контейнера составляло половину от расстояния между flex-элементами.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

```
justify-content: flex-start;
```


```
justify-content: flex-end;
```




```
justify-content: center;
```


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


```
justify-content: space-around;
```

Выравнивание по поперечной оси: align-items и align-self

*Flex-элементы можно выравнивать **по поперечной оси текущей строки** flex-контейнера. *align-items* устанавливает выравнивание **для всех элементов** flex-контейнера, включая анонимные flex-элементы. *align-self* позволяет переопределить это выравнивание для отдельных flex-элементов. Если любое из поперечных *margin* flex-элемента имеет значение *auto*, *align-self* не имеет никакого влияния.

align-items	
Значения:	
flex-start	Верхний край flex-элемента помещается вплотную с flex-линией (или на расстоянии, с учетом заданных полей *margin* и рамок *border* элемента), проходящей через начало поперечной оси.
flex-end	Нижний край flex-элемента помещается вплотную с flex-линией (или на расстоянии, с учетом заданных полей *margin* и рамок *border* элемента), проходящей через конец поперечной оси.
content	Поля flex-элемента центрируется по поперечной оси в пределах flex-линии.
baseline	Базовые линии всех flex-элементов, участвующих в выравнивании, совпадают.
stretch	Если поперечный размер flex-элемента вычисляется как *auto* и ни одно из поперечных значений *margin* не равно *auto*, элемент растягивается. Значение по умолчанию.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

```
align-items: stretch;
```


```
align-items: flex-start;
```

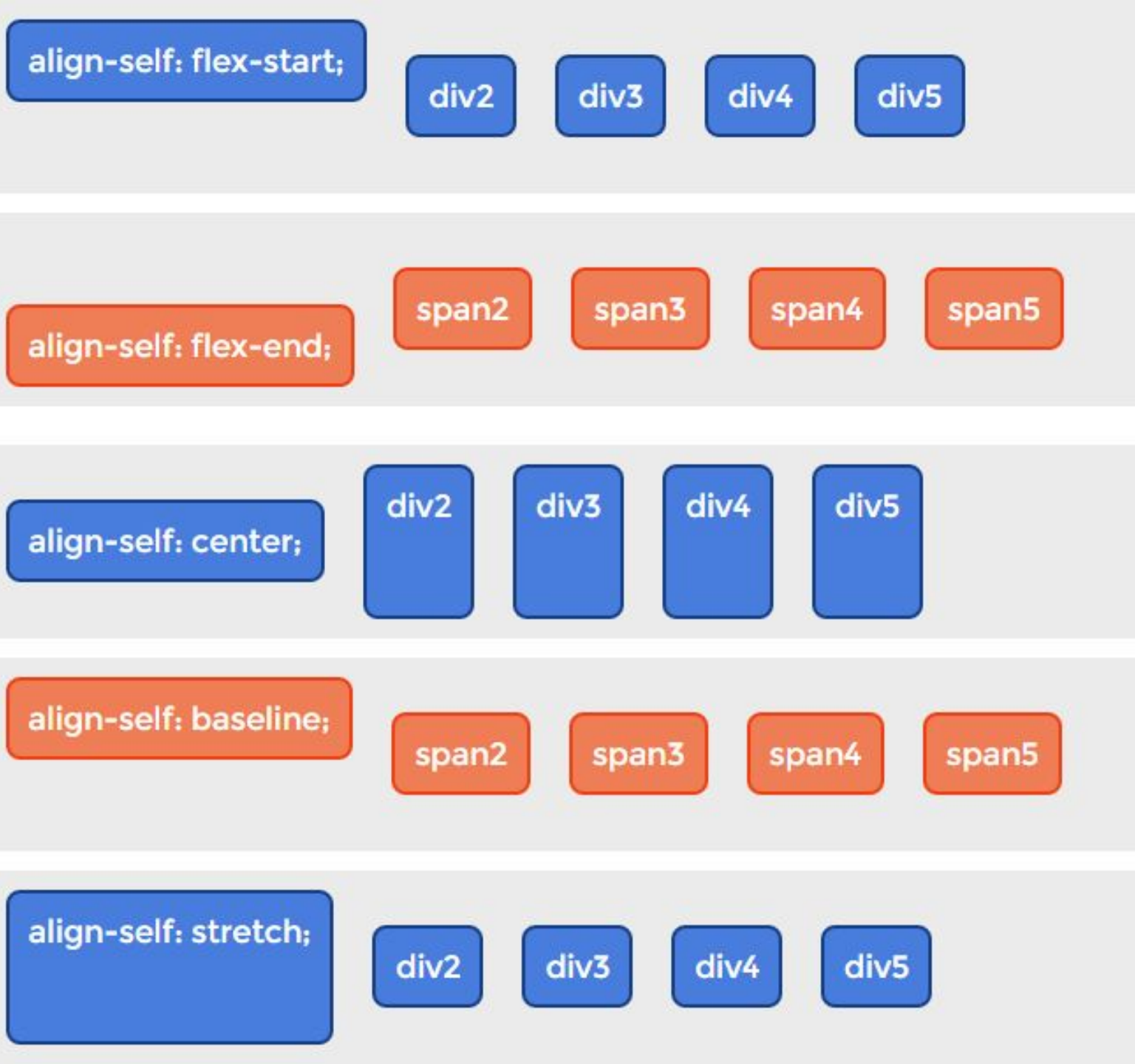

```
align-items: flex-end;
```


```
align-items: center;
```


```
align-items: baseline;
```

Выравнивание по поперечной оси: align-items и align-self

align-self	
Значения:	
auto	Значение по умолчанию. Flex-элемент использует выравнивание, указанное в свойстве *align-items* flex-контейнера.
flex-start	Верхний край flex-элемента помещается вплотную с flex-линией (или на расстоянии, с учетом заданных полей *margin* и рамок *border* элемента), проходящей через начало поперечной оси.
flex-end	Нижний край flex-элемента помещается вплотную с flex-линией (или на расстоянии, с учетом заданных полей *margin* и рамок *border* элемента), проходящей через конец поперечной оси.
center	Поля flex-элемента центрируется по поперечной оси в пределах flex-линии.
baseline	Flex-элемент выравнивается по базовой линии.
stretch	Если поперечный размер flex-элемента вычисляется как *auto* и ни одно из поперечных значений *margin* не равно *auto*, элемент растягивается. Значение по умолчанию.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.


align-self: flex-start;
div2
div3
div4
div5
align-self: flex-end;
span2
span3
span4
span5
align-self: center;
div2
div3
div4
div5
align-self: baseline;
span2
span3
span4
span5
align-self: stretch;
div2
div3
div4
div5

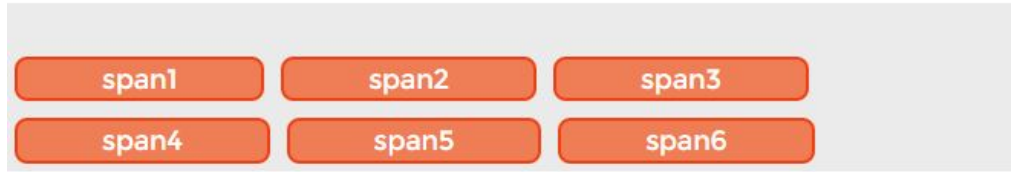
Выравнивание строк flex-контейнера: align-content

*Свойство выравнивает **строки** в flex-контейнере при наличии дополнительного пространства на поперечной оси, аналогично выравниванию отдельных элементов на главной оси с помощью свойства *justify-content*. Свойство **не влияет на однострочный flex-контейнер**. Не наследуется.

align-content	
Значения:	
flex-start	Строки укладываются по направлению к началу flex-контейнера. Край первой строки помещается вплотную к краю flex-контейнера, каждая последующая — вплотную к предыдущей строке.
flex-end	Строки укладываются по направлению к концу flex-контейнера. Край последней строки помещается вплотную к краю flex-контейнера, каждая предыдущая — вплотную к последующей строке.
center	Строки укладываются по направлению к центру flex-контейнера. Строки расположены вплотную друг к другу и выровнены по центру flex-контейнера с равным расстоянием между начальным краем содержимого flex-контейнера и первой строкой и между конечным краем содержимого flex-контейнера и последней строкой.
space-between	Строки равномерно распределены в flex-контейнере. Если оставшееся свободное пространство отрицательно или в flex-контейнере имеется только одна flex-линия, это значение идентично *flex-start*. В противном случае край первой строки помещается вплотную к начальному краю содержимого flex-контейнера, край последней строки — вплотную к конечному краю содержимого flex-контейнера. Остальные строки распределены так, чтобы расстояние между любыми двумя соседними строками было одинаковым.
space-around	Строки равномерно распределены в flex-контейнере с половинным пробелом на обоих концах. Если оставшееся свободное пространство отрицательно, это значение идентично *center*. В противном случае строки распределяются таким образом, чтобы расстояние между любыми двумя соседними строками было одинаковым, а расстояние между первой/последней строками и краями содержимого flex-контейнера составляло половину от расстояния между строками.
stretch	Значение по умолчанию. Строки flex-элементов равномерно растягиваются, заполняя все доступное пространство. Если оставшееся свободное пространство отрицательно, это значение идентично flex-start. В противном случае свободное пространство будет разделено поровну между всеми строками, увеличивая их поперечный размер.
initial	Устанавливает значение свойства в значение по умолчанию.

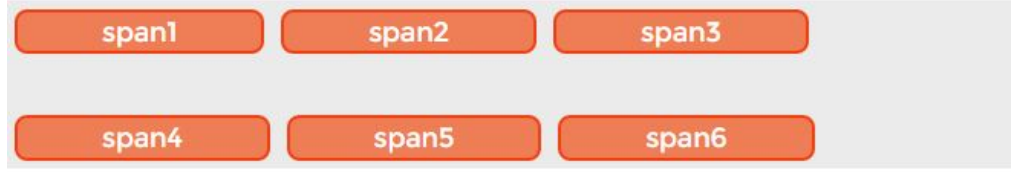
```
align-content: flex-start;
```


```
align-content: flex-end;
```



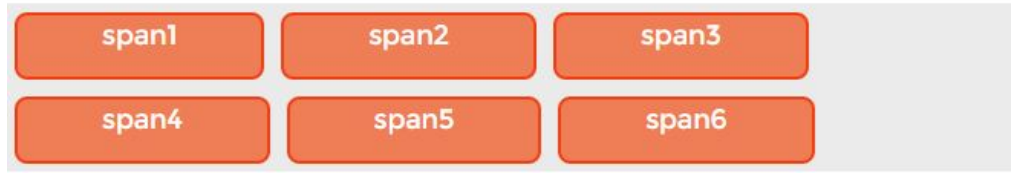

```
align-content: center;
```


```
align-content: space-between;
```




```
align-content: space-around;
```


```
align-content: stretch;
```

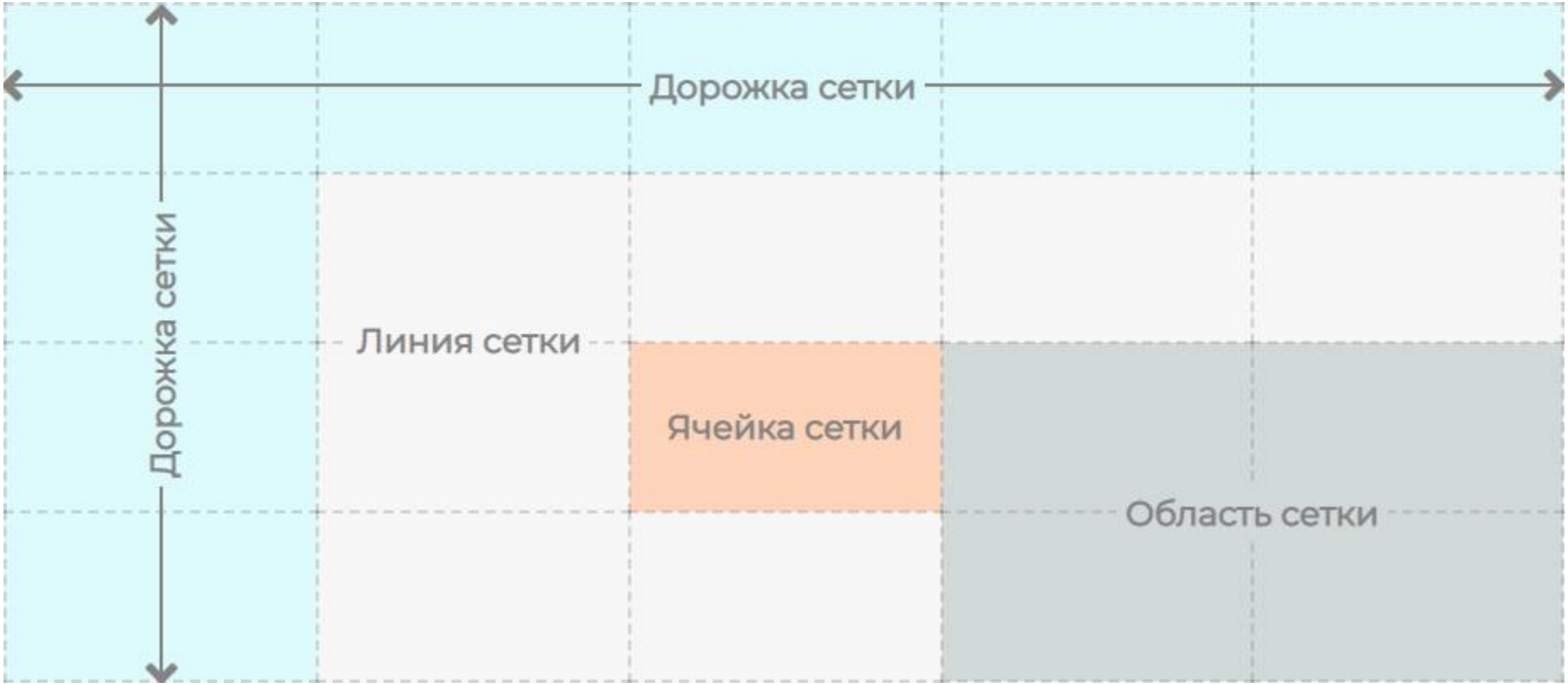


Grid layout

Это система **двумерного** макета, оптимизированная для дизайна пользовательского интерфейса. Главная идея, лежащая в основе макета сетки, заключается в **разделении веб-страницы на столбцы и строки**. В образовавшиеся области сетки можно помещать элементы сетки, а управлять их размерами и расположением можно с помощью специальных свойств модуля.

Кроме того, благодаря своей способности явно размещать элементы в сетке, Grid Layout позволяет кардинально преобразовывать структуру визуального макета (отображаемого на экране), не требуя соответствующих изменений разметки.

Grid обеспечивает двухмерное выравнивание, использует нисходящий подход к макету, допускает явное перекрытие элементов и обладает более мощными связующими возможностями. Flexbox фокусируется на распределении пространства по оси, использует более простой восходящий подход к макету, может использовать систему переноса строк на основе размера контента для управления своей вторичной осью и опирается на базовую иерархию разметки для построения более сложных макетов.


Дорожка сетки
Дорожка сетки
Линия сетки
Ячейка сетки
Область сетки

Сетка (grid) представляет собой набор пересекающихся горизонтальных и вертикальных линий, делящих пространство grid-контейнера на области сетки, в которые могут быть помещены элементы сетки.

Линии сетки (grid lines) — это невидимые горизонтальные и вертикальные разделительные линии, они существуют по обе стороны от строки и столбца. На них можно ссылаться по числовому индексу (используя свойства *grid-column-star*t, *grid-column-end*, *grid-row-start* и *grid-row-end*) или имени, заданному в CSS-коде. Числовые индексы сетки зависят от стиля языка, поэтому первым столбцом может быть как самый левый, так и самый правый столбец.

Выделяют две группы линий сетки: одна группа определяет столбцы, которые проходят вдоль оси блока (ось столбцов), и перпендикулярная группа, определяющая строки, простирающиеся вдоль линейной оси (ось строк).

Дорожка сетки (grid track) — пространство между двумя соседними линиями сетки, используется для определения либо столбца, либо строки сетки. Дорожка идет от одного края контейнера к другому, размер зависит от расположения линий сетки, которые ее определяют. Дорожки сетки аналогичны столбцам и строкам таблицы. По умолчанию смежные дорожки плотно прилегают друг к другу, задать расстояние между ними можно с помощью свойств *row-gap*, *column-gap* и *gap*.

Ячейка сетки (grid cell) — пространство, ограниченное четырьмя линиями сетки, аналогично ячейке таблицы. Ячейка сетки — это область, в которой можно разместить контент. Это наименьшая единица сетки, на которую можно ссылаться при позиционировании элементов сетки. К ячейкам сетки нельзя обращаться напрямую с помощью CSS-свойств.

Область сетки (grid area) — прямоугольная область, ограниченная четырьмя линиями сетки и состоящая из одной или нескольких соседних ячеек. Область может быть такой же маленькой, как одна ячейка, или такой же большой, как все ячейки сетки. Область сетки может быть задана явно с помощью свойства grid-template-areas, по умолчанию на нее ссылаются ограничивающие линии сетки.

Элементы сетки (grid items) — отдельные элементы, которые назначаются области сетки (или ячейке сетки). Каждый контейнер-сетка включает ноль и более элементов сетки; каждый дочерний элемент контейнера-сетки автоматически становится элементом сетки.

Дорожки, ячейки и области сетки построены из линий сетки. Тем не менее не требуется, чтобы все области сетки были заполнены элементами, вполне возможно, что некоторые или даже большинство ячеек сетки будут пустыми от любого содержимого. Также возможно, что элементы сетки будут перекрывать друг друга, либо определять перекрывающиеся области сетки.

Контейнер-сетка (grid container)

Это блок, который устанавливает контекст форматирования по типу сетки, то есть **создает область с сеткой**, а дочерние элементы располагаются в соответствии с правилами **компоновки сетки**, а не блочной компоновки. Когда вы определяете контейнер сетки с помощью *display: grid* или *display: inline-grid*, вы создаете новый контекст форматирования для содержимого этого контейнера, который влияет только на дочерние элементы сетки.

```
div style=display: grid;
```

Donec in tellus vel neque sollicitudin lobortis sed a arcu. Quisque eu arcu vel eros pretium posuere. Quisque semper nunc ligula, eget laoreet nibh faucibus vel.

```
div style=display: inline-grid;
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin viverra lorem libero, eget ultricies risus vulputate eu. Donec pretium in risus a suscipit.

Контейнер-сетка (grid container)

Контейнер-сетка бывает двух видов: обычный ***display: grid*** и встроенный ***display: inline-grid***.

Первый генерирует grid-контейнер уровня **блока**, **второй** — grid-контейнер уровня **строки**. Контейнеры-сетки не являются блочными контейнерами, поэтому некоторые CSS-свойства не работают в контексте макета сетки:

- *float* и *clear* игнорируются элементами сетки (но не самим контейнером-сеткой).
- *vertical-align* не влияет на элементы сетки.
- Псевдоэлементы *::first-line* и *::first-letter* не применяются к контейнеру-сетке и его потомкам.
- Если контейнер-сетка является контейнером уровня строки *display: inline-grid* и для него заданы обтекание или абсолютное позиционирование, то вычисляемое значение свойства *display* будет *grid*.

Определение сетки

Когда вы создаете контейнер-сетку, сетка по умолчанию имеет **один столбец и одну строку**, которые занимают полный размер контейнера. Для разделения контейнера-сетки на столбцы и строки используются свойства ***grid-template-columns***, ***grid-template-rows*** и ***grid-template-areas***. С помощью этих свойств можно определить сетку явно.

Окончательная сетка может оказаться больше из-за элементов сетки, размещенных вне явной сетки; в этом случае будут созданы неявные дорожки, размер этих неявных дорожек будет определяться свойствами *grid-auto-rows* и *grid-auto-columns*.

Свойства *grid* и *grid-template* — это сокращенные обозначения, которые можно использовать для одновременной установки всех трех явных свойств сетки *grid-template-columns*, *grid-template-rows* и *grid-template-areas*. *grid* сбрасывает свойства, управляющие неявной сеткой, тогда как свойство *grid-template* оставляет их без изменений.

grid-template-rows, grid-template-columns

*Размеры дорожек сетки можно задавать с помощью положительных значений, используя относительные единицы длины — например, *em*, *vh*, *vw*; абсолютные единицы длины — *px*; и проценты *%*. Размеры в *%* вычисляются от **ширины или высоты контейнера-сетки**.

grid-template-rows, grid-template-columns	
Значения:	
none	Указывает, что свойство не создает явных дорожек сетки (хотя явные дорожки сетки все еще могут создаваться свойством *grid-template-areas*). При отсутствии явной сетки любые строки/столбцы будут генерироваться неявно, а их размер будет определяться свойствами *grid-auto-rows* и *grid-auto-columns*. Значение по умолчанию.
(автоматический) список дорожек	Устанавливает список дорожек в виде последовательности функций определения размера дорожек и названий линий сетки. Каждая функция определения размера дорожки может быть задана в единицах длины, как процент от размера контейнера-сетки или доля свободного пространства в сетке. Размер также может быть указан как диапазон с помощью нотации *minmax()*.

grid-template-rows, grid-template-columns

fr — единица длины, которая позволяет создавать гибкие дорожки. **Не является единицей измерения в обычном ее понимании, поэтому не может быть представлена или объединена с другими типами единиц в выражениях *calc()*.** Общий размер фиксированных строк или столбцов вычитается из доступного пространства контейнера-сетки. Оставшееся пространство делится между строками и столбцами с гибкими размерами пропорционально их коэффициенту.

Если сумма всех гибких размеров дорожек меньше 1, они будут занимать только **соответствующую долю оставшегося пространства**, а **не расширяться**, чтобы заполнить его полностью.

Если доступное пространство бесконечно (то есть, ширина или высота контейнера-сетки не заданы), дорожки сетки гибкого размера **масштабируются по своему содержимому**, сохраняя при этом их соответствующие пропорции.

grid-template-rows, grid-template-columns

Ключевое слово ***max-content*** устанавливает для дорожки размер, который занимает **максимально необходимое пространство с учетом содержимого элемента сетки**.

min-content позволяет занимать **минимальное пространство, необходимое для этого содержимого**, при этом ширина элемента ориентируется на самое длинное слово или на самое широкое изображение.

Функция ***minmax****(min,max)* определяет **диапазон размеров**, больше или равный *min* и меньше или равный *max*. Если *max* < *min*, то *max* игнорируется, а *minmax(min,max)* обрабатывается как *min*. Значения в *fr* можно устанавливать только как максимальное.

- *minmax(длина или min-content или max-content или auto, длина или fr или min-content или max-content или auto);*
- *minmax(длина, длина или fr или min-content или max-content или auto);*
- *minmax(длина или min-content или max-content или auto, длина);*

grid-template-rows, grid-template-columns

Значение ***auto*** ориентируется на **содержимое элементов сетки одной дорожки**. Как минимум, рассматривается как минимальный размер элемента сетки, как определено *min-width* или *min-height*. Как максимум, обрабатывается так же, как и *max-content*. Может растягиваться за счет свойств *align-content* и *justify-content*.

Размеры дорожек можно задавать с помощью значения ***fit-content***(длина или %), представляющее собой формулу *min(maximum size*, *max(minimum size, argument))*, которая вычисляется как *minmax(auto, max-content)*, то есть *auto*. При этом, **размер дорожки ограничивается значением, указанным в скобках**, и если оно больше, чем автоматический минимум.

grid-template-rows, grid-template-columns

Нотация ***repeat()*** представляет **повторяющийся фрагмент списка дорожек**, что позволяет записать в более компактной форме большое количество одинаковых по размерам столбцов или строк. Общая форма синтаксиса следующая:

repeat(число или auto-fill или auto-fit, повторяющаяся дорожка);

Первый аргумент задает **количество повторений**, которое может быть задано с помощью положительного целого числа или ключевых слов. Второй аргумент - **размер повторяющейся дорожки**. Однако, существуют некоторые ограничения:

- Нотация *repeat()* не может быть вложенной.
- Значения *auto-fill* или *auto-fit* не могут быть совмещены с *min-content*, *max-content*, *auto*, *fit-content()* или *fr*.

*Используя значение *auto-fill*, вы всегда получите хотя бы один столбец, даже если он по какой-то причине не помещается в контейнер-сетку. Если вы используете *auto-fit*, то дорожки, которые не содержат элементы сетки, будут сброшены.

Именованные области: grid-template-areas

Свойство ***grid-template-areas*** определяет именованные области сетки, которые не связаны с каким-либо конкретным элементом сетки, но на которые можно ссылаться из свойств размещения сетки. Синтаксис свойства обеспечивает визуализацию структуры сетки, облегчая понимание общего макета контейнера-сетки. Свойство не наследуется.

grid-template-areas		
Значения:		
none		Указывает, что никакие именованные области сетки и никакие явные дорожки сетки не определены этим свойством (хотя явные дорожки сетки все еще могут быть созданы с помощью *grid-template-columns* или *grid-template-rows*). При отсутствии явной сетки любые строки/столбцы будут генерироваться неявно, а их размер будет определяться свойствами *grid-auto-rows* и *grid-auto-columns*. Значение по умолчанию.
строка +		Последовательность идентификаторов, определяющая, как должны отображаться строки и столбцы. grid-template-areas: "header header" "sidebar content" "sidebar content";

grid-template-areas

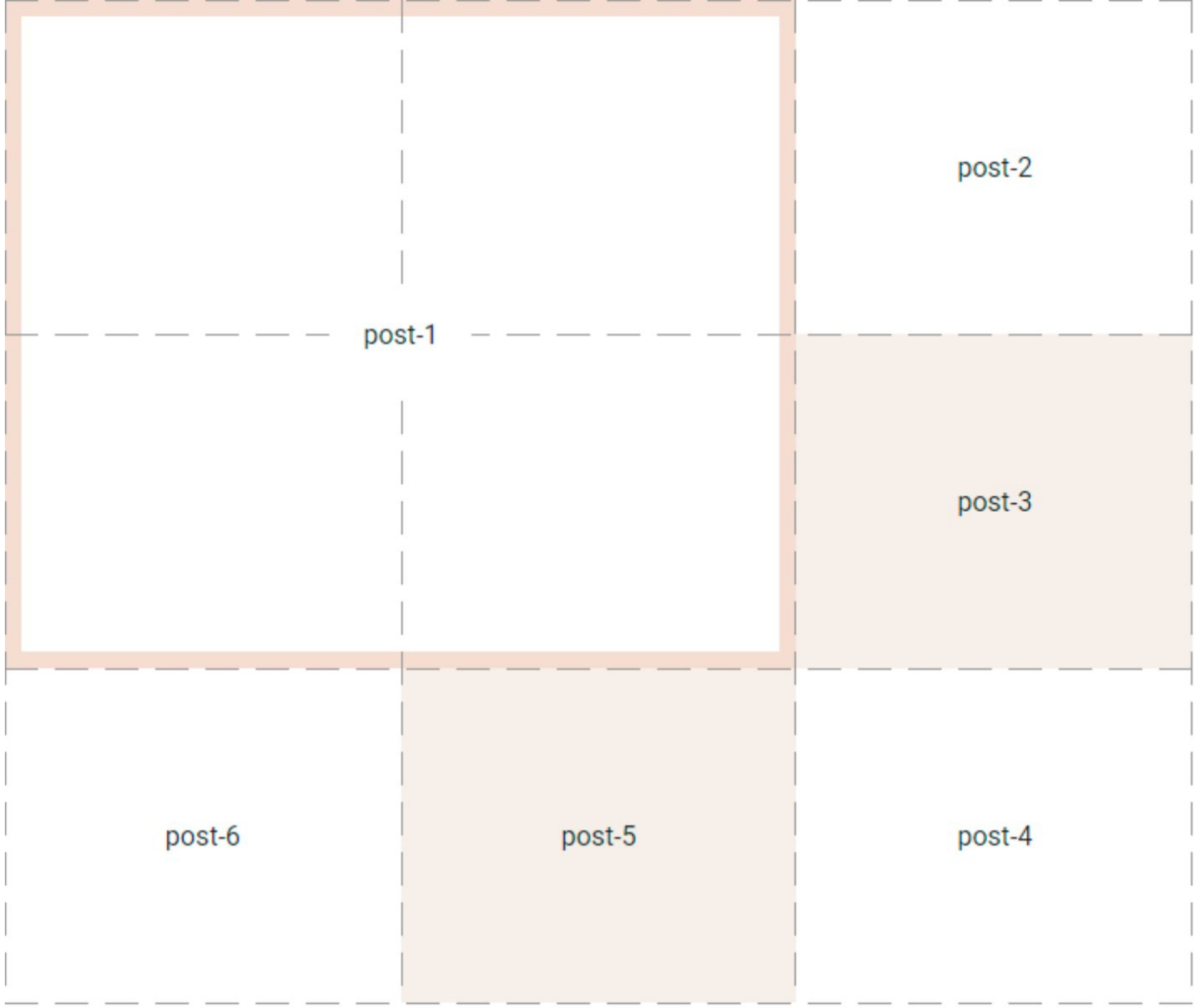
Каждый идентификатор сетки в значении *grid-template-areas* соответствует ячейке сетки. Как только все ячейки идентифицированы, браузер объединяет все смежные ячейки с одинаковыми именами в одну область, которая охватывает все их, при условии, что они описывают область прямоугольной формы. Если вы попытаетесь настроить более сложные области, весь шаблон будет недействительным и области сетки не будут определены.

Все строки должны содержать **одинаковое количество столбцов**. Если вы хотите определить только некоторые ячейки как часть области сетки, вы можете использовать одну или несколько **.** для заполнения этих безымянных ячеек. При определении областей сетки идентификаторы можно перечислять через единичный пробел, без разрыва строки. Или же выровнять с помощью пробелов/табуляции и перевода строки для большей наглядности.

Области сетки полезны для определения семантических отношений между различными частями макета страницы, позволяя указать, какая часть страницы включает в себя верхний колонтитул, боковую панель, область содержимого и нижний колонтитул.

После того, как вы создали области сетки, элементы сетки могут быть назначены непосредственно, чтобы занимать эти области, используя свойство *grid-area*.

```
.grid-container {
display: grid;
grid-template-areas: "post-1 post-1 post-2»
                     "post-1 post-1 post-3"
                     "post-6 post-5 post-4";
 grid-template-rows: repeat(3, 200px);
grid-template-columns: repeat(3, 1fr);
}
.post-1 {
grid-area: post-1;
 }
.post-2 {
 grid-area: post-2;
}
 .post-3 {
grid-area: post-3;
 }
.post-4 {
grid-area: post-4;
 }
.post-5 {
grid-area: post-5;
}
.post-6 {
grid-area: post-6;
 }
```



Краткая запись явной сетки

*Свойство ***grid-template*** является сокращением для установки *grid-template-rows*, *grid-template-columns* и *grid-template-areas* в одном объявлении.

grid-template	
Значения:	
none	Устанавливает для всех трех свойств начальные значения *none*.
grid-template-rows/grid-template-columns	Устанавливает *grid-template-rows* и значение *grid-template-columns* в указанные значения, а *grid-template-areas* в значение *none*. *grid-template: repeat(3, 200px)/repeat(3, 1fr);*
имена линий сетки или последовательность идентификаторов, заключенная в кавычки и размер дорожки или именованные линии сетки или + /явный список дорожек	Устанавливает *grid-template-areas* для перечисленных последовательностей идентификаторов. Устанавливает для *grid-template-rows* указанные значения размеров дорожек, следующие за каждой последовательностью идентификаторов (выставляя *auto* для любых отсутствующих размеров), и объединяет в именованных линиях сетки, определенных до/после каждого размера. Устанавливает *grid-template-columns* в список дорожек, указанный после косой черты (или ни одного, если не указан). *grid-template: [start] "post-1 post-1 post-2" 200px [row2]* *[row2] "post-1 post-1 post-3" 200px [row3]* *[row3] "post-6 post-5 post-4" 200px [row-end] / 1fr 1fr 1fr;*

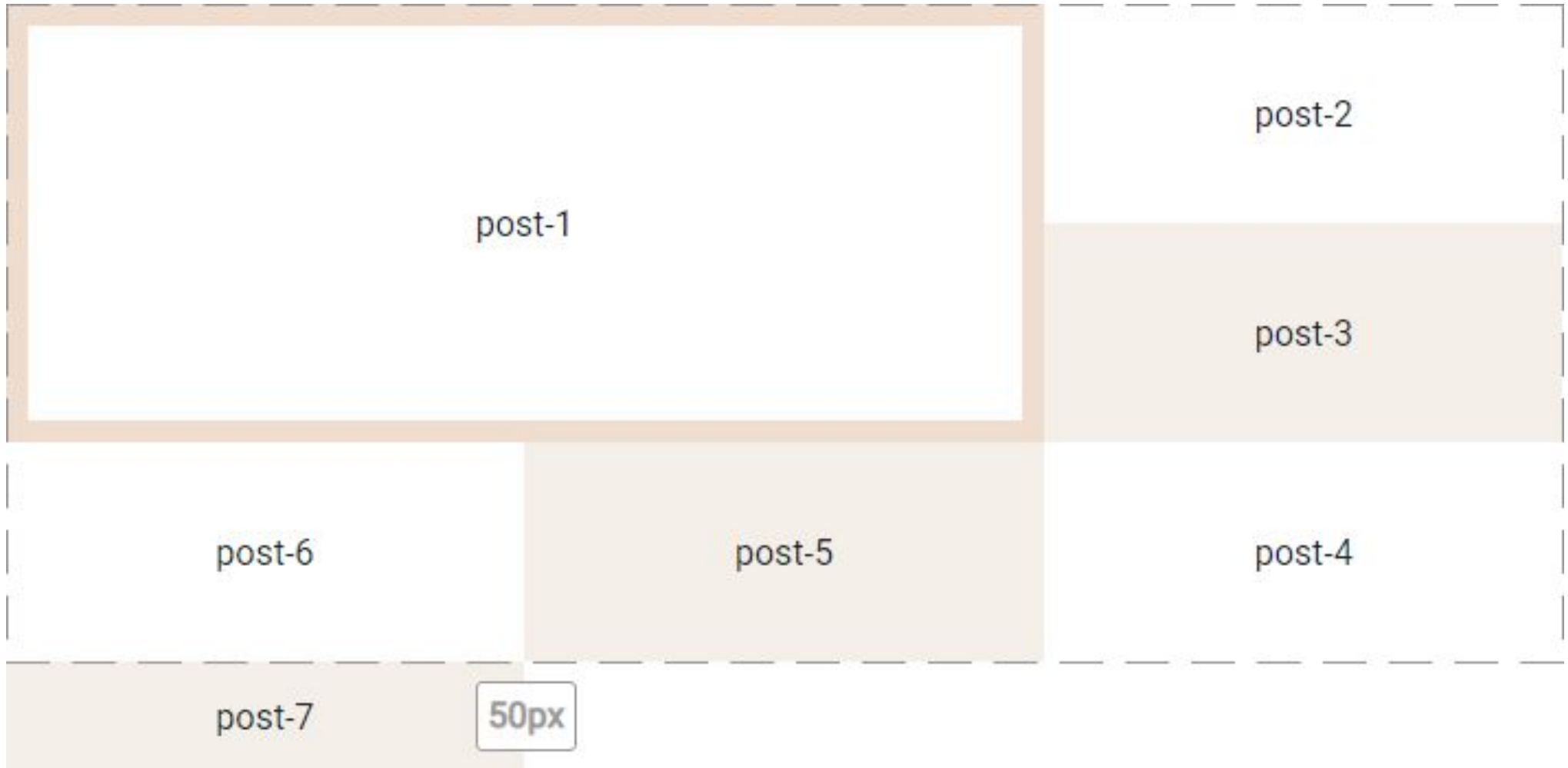
Неявная сетка

Если элемент сетки расположен в строке или столбце, размер которых не определен явно *grid-template-rows* или *grid-template-columns*, создаются **неявные дорожки сетки** для его хранения. Это может произойти в случае, если строка или столбец оказались за пределами установленных размеров сетки.

По умолчанию эти автоматически добавляемые дорожки имеют минимальный необходимый размер.
Свойства *grid-auto-rows* и *grid-auto-columns* позволяют контролировать размер неявных дорожек сетки. **Если дано несколько размеров дорожек, шаблон повторяется по мере необходимости**, чтобы найти размер неявных дорожек. Первая неявная дорожка сетки после явной сетки получает первый

grid-auto-columns, grid-auto-rows	
Значения:	
auto	Значение по умолчанию.
размер дорожки +	В качестве размера дорожки может использоваться любое значение, допустимое для задания размеров дорожек сетки.

```
.grid-container {
max-width: 710px;
display: grid;
grid-template-columns: repeat(3,1fr);
grid-template-rows: repeat(3,100px);
 grid-auto-rows: 50px;
}
.post-1 {
grid-column: 1/3;
grid-row: 1/3;
 }
.post-2 {
grid-column: 3;
 grid-row: 1;
 }
 .post-3 {
grid-column: 3;
grid-row: 2;
}
.post-4 {
grid-column: 3;
grid-row: 3;
}
.post-5 {
grid-column: 2;
grid-row: 3;
}
 .post-6 {
grid-column: 1;
 grid-row: 3;
}
```

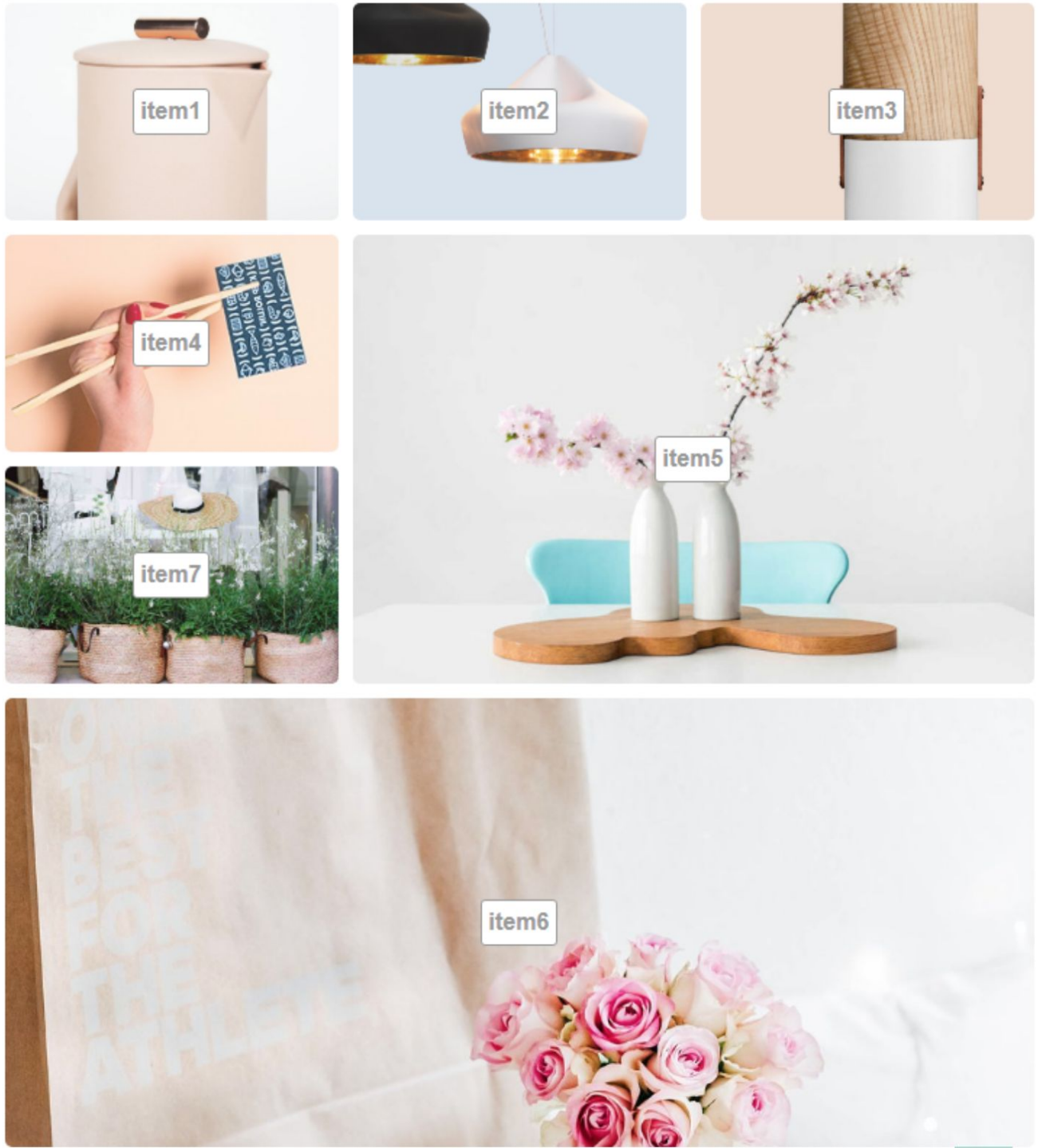


Автоматическое размещение

*Элементы сетки, которые не размещены явно, **автоматически помещаются в незанятое пространство в контейнере-сетке с помощью алгоритма автоматического размещения**. Свойство *grid-auto-flow* управляет автоматическим размещением элементов сетки без явного положения. После заполнения явной сетки (или если явной сетки нет) автоматическое размещение также приведет к генерации неявных дорожек сетки. Свойство не наследуется.

grid-auto-flow	
Значения:	
row	Алгоритм автоматического размещения размещает элементы, заполняя каждую строку по очереди слева-направо (для LTR-языков), добавляя новые строки по мере необходимости. Значение по умолчанию.
column	Алгоритм размещает элементы, заполняя каждый столбец по очереди сверху-вниз, добавляя новые столбцы по мере необходимости.
dense	Алгоритм "плотной" укладки элементов. При необходимости может менять порядок следования элементов, заполняя пустые места более крупными элементами.

```
.grid-container {
max-width: 710px;
margin: 10px auto;
display: grid;
grid-template-columns:
repeat(3, 1fr);
grid-gap: 10px;
grid-auto-rows: 200px;
grid-auto-flow: dense;
 }
.item5 {
grid-column: span 2;
 grid-row: span 2;
 }
.item6 {
grid-column: span 3;
grid-row: span 2;
 }
```



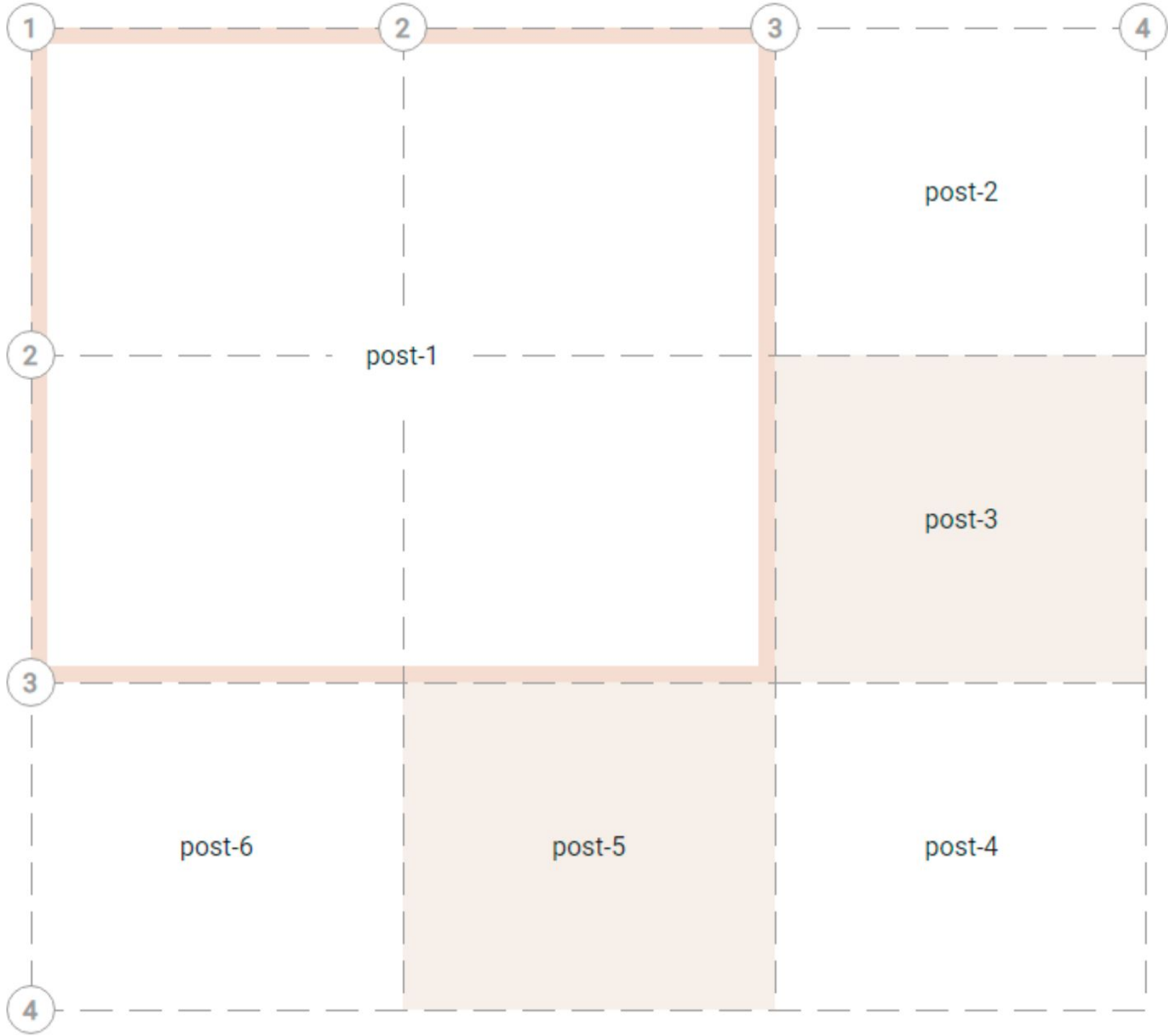
Размещение и переупорядочивание элементов сетки

Свойства размещения на сетке
- ***grid-row-start***, ***grid-row-end***, ***grid-column-start*** и ***grid-column-end*** и их краткая запись ***grid-row***, ***grid-column*** и ***grid-area*** позволяют определить размещение элемента сетки, предоставив любую (или ноль) из следующих шести частей информации:

	Строка	Столбец
Начало	Начальная линия строки	Начальная линия столбца
Конец	Конечная линия строки	Конечная линия столбца
Диапазон	Диапазон строк	Диапазон столбца

grid-row-start, grid-column-start, grid-row-end, grid-column-end		
Значения:		
auto		Свойство не влияет на размещение элемента сетки, указывая на автоматическое размещение или диапазон по умолчанию, равный единице.
имя линии		Начальная и конечная линия строки/столбца задаются в именованных линий сетки.
целое число и имя линии		Начальная и конечная линия строки/столбца задаются с помощью целого числа (отрицательное порядковый номер линии сетки будет отсчитываться с противоположного края явной сетки) и (необязательно) имени линии.
span и целое число или имя линии		Ключевое слово *span* и целое положительное число/имя линии задают диапазон ячеек для размещения элемента сетки.

```
.grid-container {
display: grid;
grid-template-rows: 200px 200px 200px;
grid-template-columns: 1fr 1fr 1fr;
}
.post-1 {
 grid-row-start: 1;
 grid-row-end: 3;
grid-column-start: 1;
grid-column-end: 3;
 }
.post-2 {
grid-row-start: 1;
 grid-column-start: 3;
}
 .post-3 {
grid-row-start: 2;
 grid-column-start: 3;
 }
.post-4 {
grid-row-start: 3;
 grid-column-start: 3;
 }
.post-5 {
grid-row-start: 3;
 grid-column-start: 2;
}
 .post-6 {
 grid-row-start: 3;
 grid-column-start: 1;
}
```



Краткая запись свойств размещения элементов сетки

Свойства ***grid-row*** и ***grid-column*** являются сокращенными именами для свойств *grid-row-start*/*grid-row-end* и *grid-column-start*/*grid-column-end* соответственно.

Если заданы два значения, **первое** (до косой черты) устанавливается для параметра *grid-row-start*/*grid-column-start*, **второе** - для *grid-row-end*/*grid-column-end*. Если второе значение опущено, а первое указано в формате пользовательского идентификатора, то *grid-row-end*/*grid-column-end* также устанавливается в пользовательское имя сетки. В противном случае, оно вычисляется в *auto*.

Для свойства **grid-area** если указано четыре значения, **первое** устанавливается для *grid-row-start*, **второе** - для *grid-column-start*, **третье** - для *grid-row-end*, **четвертое** - для *grid-column-end*.

Если *grid-column-end*/*grid-row-end* **не указан**, а *grid-column-start*/*grid-row-start* **указан в форме пользовательского имени**, то для *grid-column-end*/*grid-row-end* также устанавливается значение пользовательского имени линии; в противном случае он установлен на *auto*.

Когда *grid-column-start* опущен, а значение *grid-row-start* указан в форме пользовательского имени, оно устанавливается для всех четырех значений. В противном случае оно устанавливается на auto.

Выравнивание элементов сетки

Для выравнивания элементов сетки можно использовать свойство *margin*, аналогично, как работает это свойство для блочных элементов.

По умолчанию элементы сетки растягиваются, чтобы заполнить свою область сетки. Тем не менее, если *justify-self* или *align-self* вычисляют значение, отличное от *stretch* или задано *margin: auto*, элементы сетки будут автоматически изменяться в соответствии с их содержимым.

Выравнивание с помощью *margin: auto*

При расчете размеров дорожек сетки *margin: auto* обрабатываются как 0. Они поглощают положительное свободное пространство, предшествующее выравниванием с помощью свойств выравнивания. Переполняющиеся элементы игнорируют свои автоматические поля и переполнение, как указано в их свойствах выравнивания блоков.

Выравнивание по оси строки

Элементы сетки могут быть выровнены в направлении оси строки (по горизонтали для LTR-языков) с помощью свойства *justify-self* или свойства *justify-items* (заданного для контейнера-сетки).

Выравнивание по оси столбца

Элементы сетки могут выровнены в направлении, перпендикулярном оси строки с помощью свойства *align-self* или свойства *align-items*, заданного для контейнера-сетки.

Промежутки между элементами сетки

Свойства ***row-gap*** и ***column-gap*** (и их сокращенная запись ***gap***), если они указаны в контейнере сетки, определяют промежутки ***между строками и столбцами сетки***. При определении размера дорожки каждый промежуток рассматривается как **дополнительная пустая дорожка указанного размера**. Дополнительный промежуток также может быть добавлен между дорожками за счет свойств *justify-content* и *align-content*.

Промежутки добавляются только между двумя дорожками сетки, то есть они **не добавляются перед первой и после последней дорожки**.

row-gap, column-gap	
Значения:	
normal	Вычисляется как *0px*. Значение по умолчанию.
длина или %	Процентное значение вычисляется относительно размеров области сетки. Отрицательные значения не используются.