**CSS Grid Layout**

CSS Grid Layout is the most powerful layout system available in CSS

CSS Grid Layout - самая мощная система компоновки, доступная в CSS.

**CSS Grid Layout** excels at dividing a page into major regions or defining the relationship in terms of size, position, and layer, between parts of a control built from HTML primitives.

**CSS Grid layout** отлично подходит для того, чтобы разделить страницу на основные области или определить взаимосвязь размера, позиционирования и уровня между частями контента, состоящего из HTML примитивов.

Like tables, grid layout enables an author to align elements into columns and rows. However, many more layouts are either possible or easier with CSS grid than they were with tables.

Как и таблицы (<table>), grid layout позволяет выравнивать элементы в столбцы и строки. Тем не менее, с помощью CSS grid работать с элементами гораздо проще, чем с таблицами.

When working with CSS Grid, you can apply CSS rules to parent elements and children.

При работе с CSS Grid вы можете применяете правила CSS к родительским элементам и к дочерним элементам.

Properties for the container Grid.

Свойства для контейнера

Ways to declare a grid container

Способы объявления grid-контейнера

Display property

Свойство display

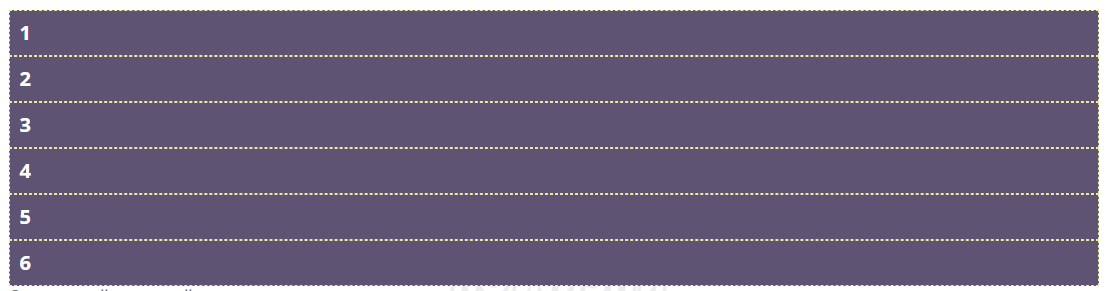
display: **grid**;

generates a block grid генерирует блочную сетку

.grid {

   display: grid;

}



.grid {

   display: inline-grid;

}



Defining an Explicit Grid

**CSS property grid-template**-columns sets the number of "columns" in the grid and can determine the width of each of them

CSS свойство grid-template-columns задаёт количество столбцов "колонок" в сетке и может определять ширину каждой из них

grid-template-rows it is a CSS property that determines the number of lines in the grid and can determine the height of each line

grid-template-rows - это CSS свойство, которое определяет количество строк в сетке и может определять высоту каждой из них

fr - a fraction of the free space in the grid

доля свободного места в сетке

fr - a unit of length that allows you to create flexible elements

fr — единица длины, которая позволяет создавать гибкие эллемент

.grid {

   display: grid;

   grid-template-columns: 1fr 1fr 1fr;

   grid-template-rows: 1fr 1fr;

}

You can combine two properties

Вы можете объединить два свойства grid-template: **1fr 1fr** / **1fr 1fr 1fr**;

The CSS **repeat ()** function presents a repeating section of a tracklist, allowing you to write a large number of columns or rows in a compact form.

Функция CSS repeat () представляет повторяющийся фрагмент списка дорожек, позволяя записывать большое количество столбцов или строк в компактной форме.

.grid {

   display: grid;

   grid-template-columns: repeat(3, 1fr);

   grid-template-rows: repeat(2, 1fr);

}

grid-template: **repeat(2,1fr)** / **repeat(3,1fr)**;



Percentage dimensions are calculated from the width or height of the grid container.

Размеры в процентах вычисляются от ширины или высоты контейнера-сетки.

.grid{

background-color: *#eee*;display: grid;grid-template-columns: 20% 30% 20%;grid-template-rows: 80% 120%;

}

grid-template: 80% 120% / 20% 30% 20%;

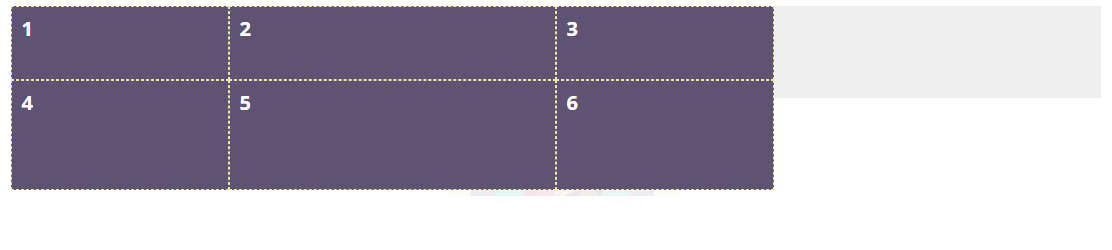
}

You can set the size of the grid items using non-flexible units like **px.**  
Вы можете установить размер элементов сетки, используя негибкие единицы, такие как пиксели.

.grid{

background-color: *#eee*;display: grid;grid-template-columns: 200px 300px 200px;grid-template-rows: 80px 120px;

}



The minmax (min, max) function defines a size range greater than or equal to min and less than or equal to max. If max <min, then max is ignored. The fr values ​​can only be set as maximum.

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

.grid {

   display: grid;

   grid-template-columns: minmax(200px, 300px) minmax(100px, 200px) minmax(250px, 150px);

   grid-template-rows: minmax(100px, 110%) minmax(50px, 1fr);

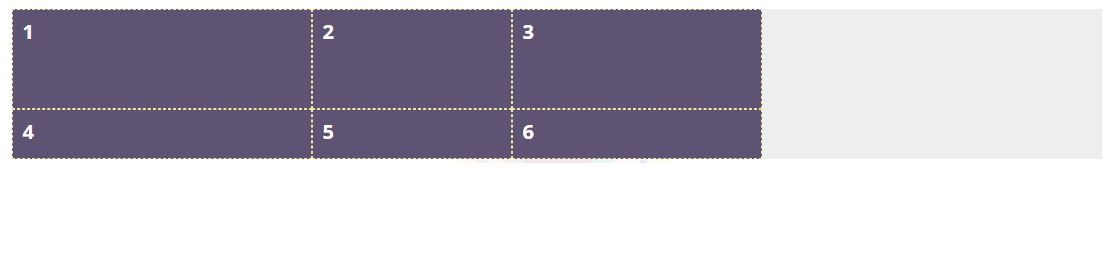
}

.grid {

   display: grid;

   grid-template: minmax(100px, 110%) minmax(50px, 1fr) / minmax(200px, 300px) minmax(100px, 200px) minmax(250px, 150px);

}



grid-template-areas

The grid-template-areas property defines named grid areas. The property syntax provides visualization of the grid structure, making it easier to understand the overall layout of the grid container.

Свойство grid-template-areas определяет именованные области сетки. Синтаксис свойства обеспечивает визуализацию структуры сетки, облегчая понимание общего макета контейнера-сетки.

.grid {

   display: grid;

   grid-template-columns: 150px 1fr;

   grid-template-areas:

      'header header'

      'sidebar content'

      'block-1 block-2'

      'footer footer';

}

.grid\_\_item:nth-child(1) {

   grid-area: header;

}

.grid\_\_item:nth-child(2) {

   grid-area: sidebar;

}

.grid\_\_item:nth-child(3) {

   grid-area: content;

}

.grid\_\_item:nth-child(4) {

   grid-area: block-1;

}

.grid\_\_item:nth-child(5) {

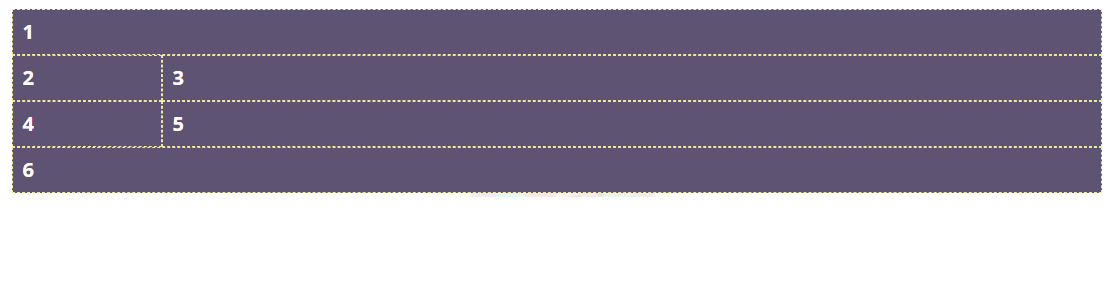
   grid-area: block-2;

}

.grid\_\_item:nth-child(6) {

   grid-area: footer;

}



Определение неявной сетки

grid-template-columns: **auto**;

grid-template-rows: **auto**;

grid-template-columns: **repeat(auto-fill,minmax(150px, 1fr))**;

grid-template-columns: repeat(auto-fit,minmax(150px, 1fr));

}

Defining an Implicit grid

Определение неявной сетки

Properties: grid-auto-columns, grid-auto-rows, grid-auto-flow

### Свойства: grid-auto-columns, grid-auto-rows, grid-auto-flow

If a grid item is positioned in a row or column that is not explicitly sized by grid-template-rows or grid-template-columns, implicit grid tracks are created to store it

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

Grid items that are not explicitly placed are automatically placed in free space in the grid container using an auto-nesting algorithm.

Элементы сетки, которые не размещены явно, автоматически помещаются в свободное пространство в контейнере-сетке с помощью алгоритма автоматического размещения.

grid-auto-columns



.grid {

   display: grid;

   grid-template: 1fr / repeat(3, 200px);

   grid-auto-columns: 50px;

   grid-auto-flow: column;

}

По умолчанию эти автоматически добавляемые дорожки имеют минимальный необходимый размер. Свойства grid-auto-rows и grid-auto-columns позволяют контролировать размер неявных дорожек сетки.

By default, these auto-added tracks are at the minimum required size. The grid-auto-rows and grid-auto-columns properties allow you to control the size of the implicit grid tracks.

grid-auto-rows



.grid {

   display: grid;

   grid-template: 1fr / repeat(3, 200px);

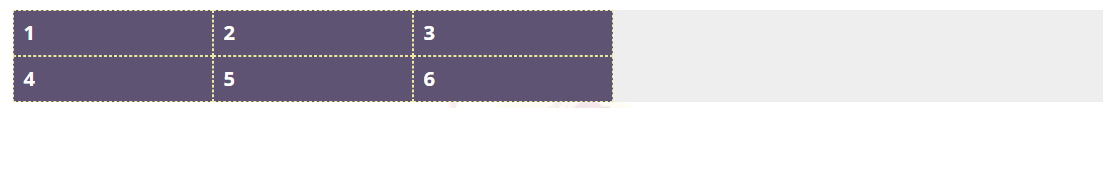
   grid-auto-rows: 100px;

}

By default, these auto-added tracks are at the minimum required size. The grid-auto-rows property allow you to control the size of the implicit grid tracks.

grid-auto-flow: **row**;  
The auto-nesting algorithm places items by filling each row from left to right, adding new rows as needed. Default value.

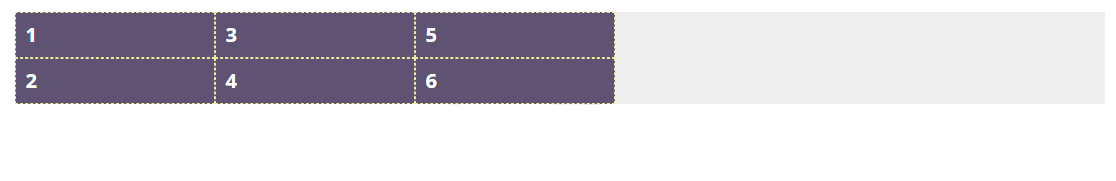
Алгоритм автоматического размещения размещает элементы, заполняя каждую строку по очереди слева-направо, добавляя новые строки по мере необходимости. Значение по умолчанию.



grid-auto-flow: **column**;

The algorithm places items by filling each column in turn from top to bottom, adding new columns as needed.

Алгоритм размещает элементы, заполняя каждый столбец по очереди сверху вниз, добавляя новые столбцы по мере необходимости.



grid-auto-flow: **dense**;

Algorithm for "dense" packing of elements. If necessary, it can change the order of the elements, filling empty spaces with larger elements

Алгоритм "плотной" укладки элементов. При необходимости может менять порядок следования элементов, заполняя пустые места более крупными элементами



### Align content along the horizontal axis

### justify-content

Выравнивание содержимого вдоль горизонтальной оси

let's look at the properties of aligning content along the horizontal axis

давайте рассмотрим свойства выравнивания содержимого вдоль горизонтальной оси

justify-content: **start**;

.grid {

   display: grid;

   height: 300px;

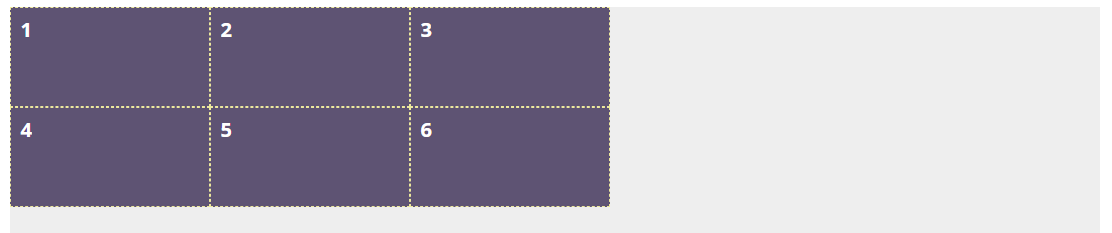
   grid-template: repeat(2, 100px) / repeat(3, 200px);

   justify-content: start;

}

Aligns the grid to the left side of the container.

Выравнивает сетку по левой стороне контейнера.



justify-content: **end**;  
Aligns the grid to the right side of the container.

Выравнивает сетку по правой стороне контейнера.

.grid {

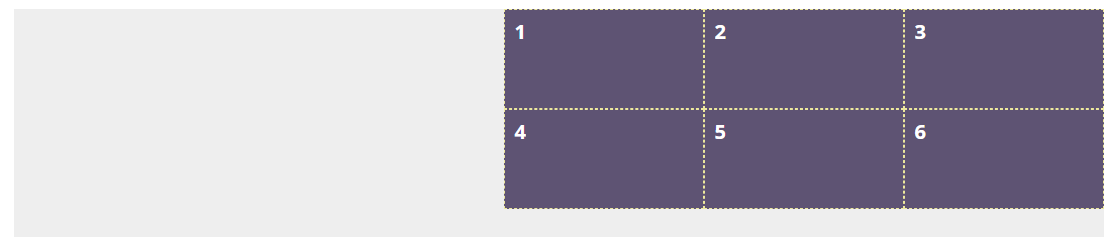
   display: grid;

   height: 300px;

   grid-template: repeat(2, 100px) / repeat(3, 200px);

   justify-content: end;

}



justify-content: **center**

Выравнивает сетку по центру контейнера.  
Aligns the grid to the center of the container.

.grid {

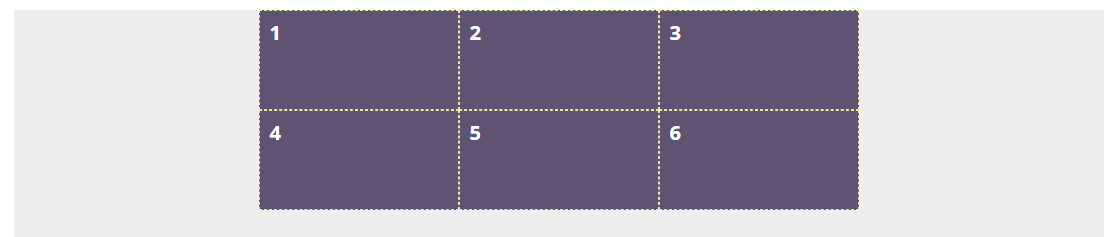
   display: grid;

   height: 300px;

   grid-template: repeat(2, 100px) / repeat(3, 200px);

   justify-content: center;

}



justify-content: **stretch**;

resizes the grid items to allow the grid to fill the full width of the grid container

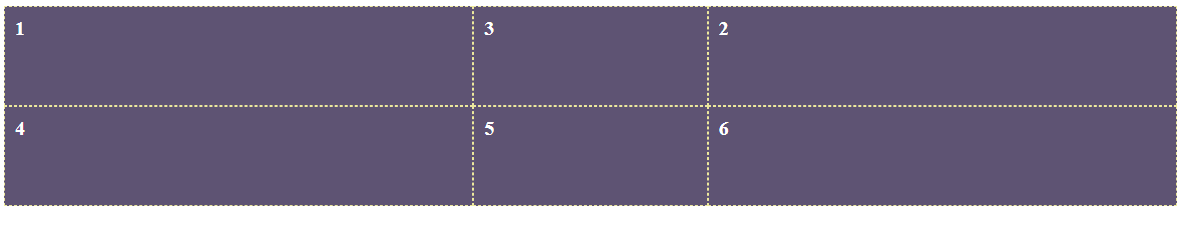
изменяет размер элементов сетки, чтобы сетка заполняла всю ширину контейнера сетки

.grid {

   display: grid;

   justify-content: stretch;

}



justify-content: **space-between**

Equal space between elements, no margins.

Одинаковое пространство между элементами, без отступов по краям.

.grid {

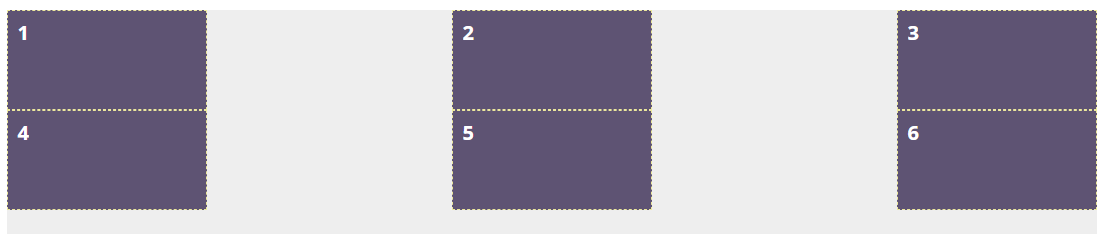
   display: grid;

   height: 300px;

   grid-template: repeat(2, 100px) / repeat(3, 200px);

   justify-content: space-between;

}



justify-content: **space-around**

это свойство делает одинаковое пространство между элементами, и полуразмерные отступы по краям.

this property makes equal space between elements, and half-size margins along the edges.

.grid {

   display: grid;

   height: 300px;

   grid-template: repeat(2, 100px) / repeat(3, 200px);

   justify-content: space-around;

}



justify-content: **space-evenly**

задает одинаковое пространство между элементами, и полноразмерные отступы по краям.

sets equal space between elements, and full-size margins.

.grid {

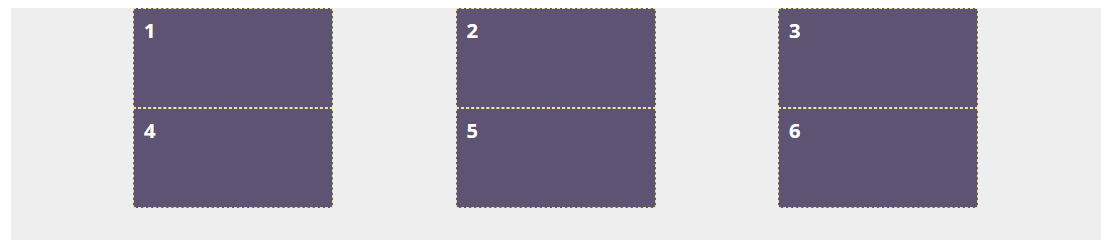
   display: grid;

   height: 300px;

   grid-template: repeat(2, 100px) / repeat(3, 200px);

   justify-content: space-evenly;

}



Align the grid along the vertical axis

Property: align-content

### Выравнивание сетки вдоль вертикальной оси Свойство: align-content

### align-content: **start**;

Aligns the grid to the top of the container.

### Выравнивает сетку по верхней стороне контейнера.

### align-content: start is the default

.grid {

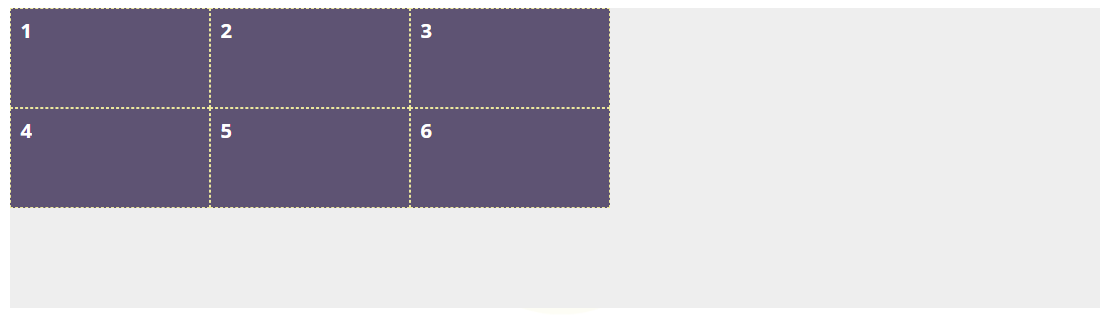
   display: grid;

   height: 300px;

   grid-template: repeat(2, 100px) / repeat(3, 200px);

   align-content: start;

}



align-content: **end**;

Aligns the grid to the top / underside of the container.

Выравнивает сетку по нижней стороне контейнера.

.grid {

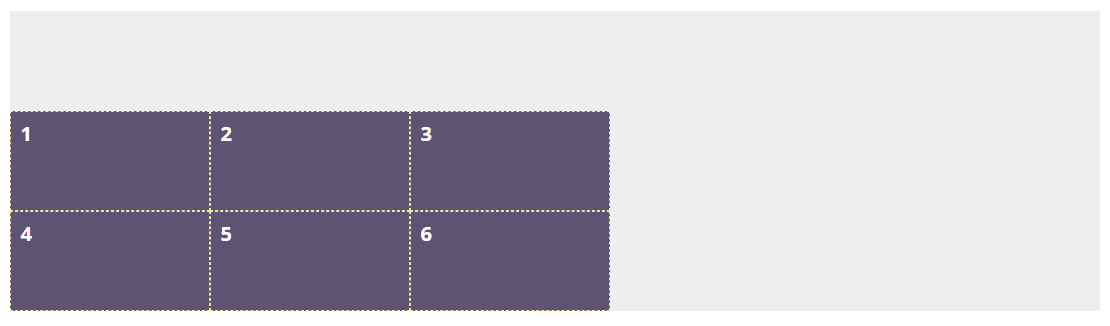
   display: grid;

   height: 300px;

   grid-template: repeat(2, 100px) / repeat(3, 200px);

   align-content: end;

}



align-content: **center**

Aligns the grid to the center of the container.

Выравнивает сетку по центру контейнера.

.grid {

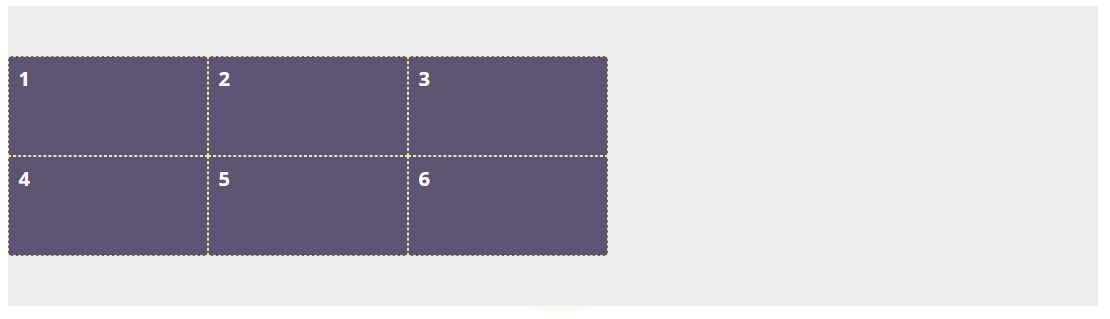
   display: grid;

   height: 300px;

   grid-template: repeat(2, 100px) / repeat(3, 200px);

   align-content: center;

}



align-content: **stretch**;  
resizes the grid items to allow the grid to fill the full height of the grid container

изменяет размер элементов сетки, чтобы сетка заполняла всю высоту контейнера сетки

.grid {

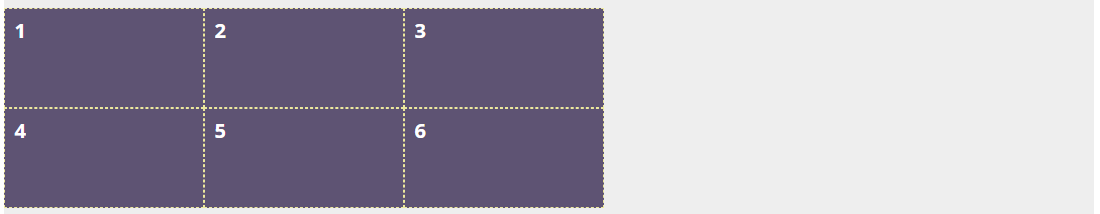
   display: grid;

   height: 300px;

   grid-template-columns: repeat(3, 200px);

   align-content: stretch;

}



align-content: **space-between**  
Equal space between elements, no margins.

Одинаковое пространство между элементами, без отступов по краям.

.grid {

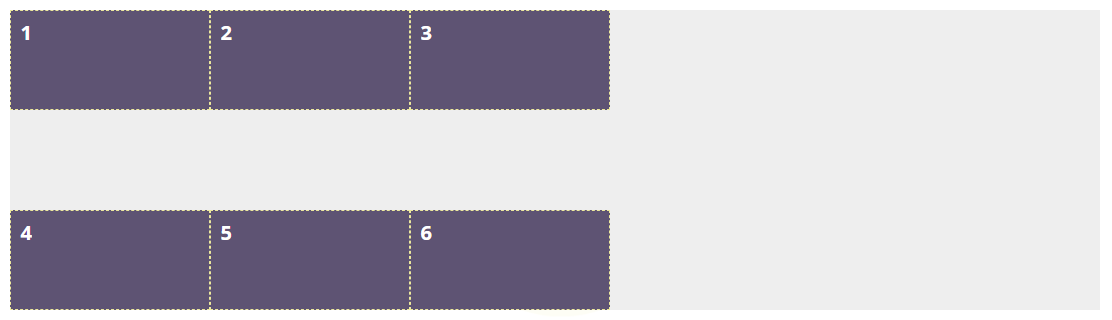
   display: grid;

   height: 300px;

   grid-template: repeat(2, 100px) / repeat(3, 200px);

   align-content: space-between;

}



align-content: **space-around**  
Equal space between elements, and half-size margins.

Одинаковое пространство между элементами, и полуразмерные отступы по краям.

.grid {

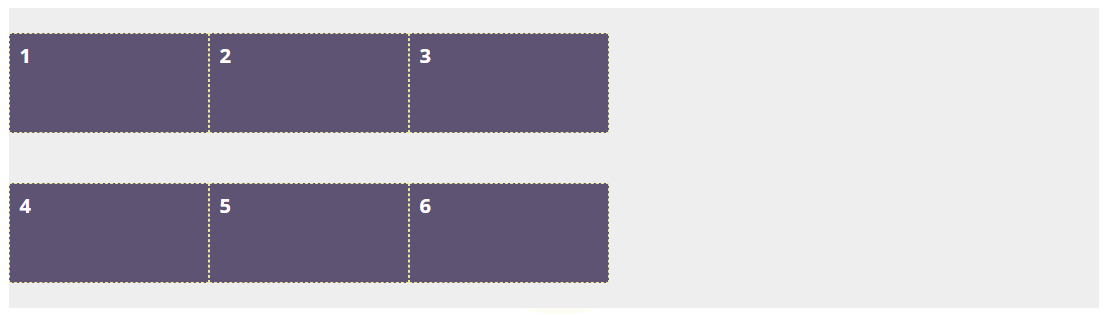
   display: grid;

   height: 300px;

   grid-template: repeat(2, 100px) / repeat(3, 200px);

   align-content: space-around;

}



align-content: **space-evenly**;

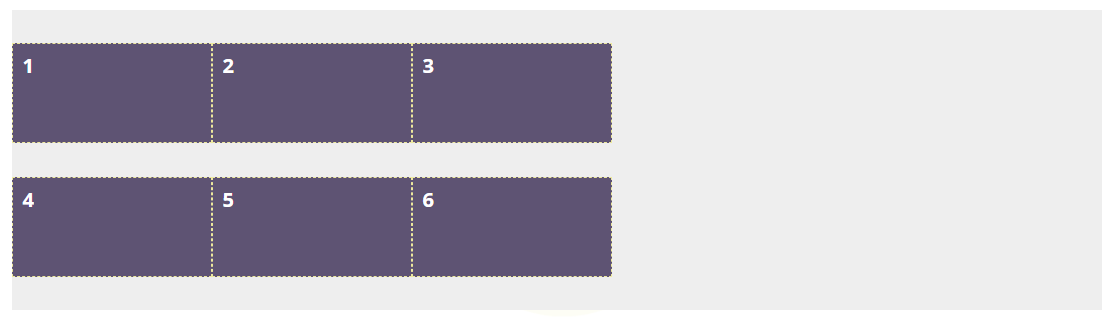
Equal space between elements, and full-size margins.

Одинаковое пространство между элементами, и полноразмерные отступы по краям.

.grid{

background-color: *#eee*;display: grid;height: 300px;grid-template: repeat(2,100px) / repeat(3,200px);align-content: space-evenly;

}



#### justify-items

Aligns the grid items along the horizontal axis. This value applies to all grid items inside the container.

Выравнивает элементы сетки по горизонтальной оси. Это значение ко всем элементам сетки внутри контейнера.

justify-items: **stretch**

fills the whole width of the cell (this is the default)  
заполняет всю ширину ячейки (по умолчанию)

.grid {

   background-color: #eee;

   display: grid;

   grid-template: repeat(2, 100px) / repeat(3, 1fr);

   justify-items: stretch;

}



justify-items: **start**

Aligns content to the left side of their cell.

Выравнивает содержимое по левому краю ячейки.

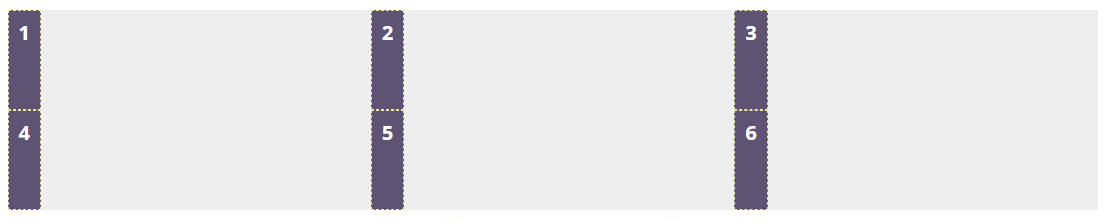
.grid {

   display: grid;

   grid-template: repeat(2, 100px) / repeat(3, 1fr);

   justify-items: start;

}



justify-items: **end**

Aligns the content to the right of the cell.

Выравнивает содержимое по правому краю ячейки.

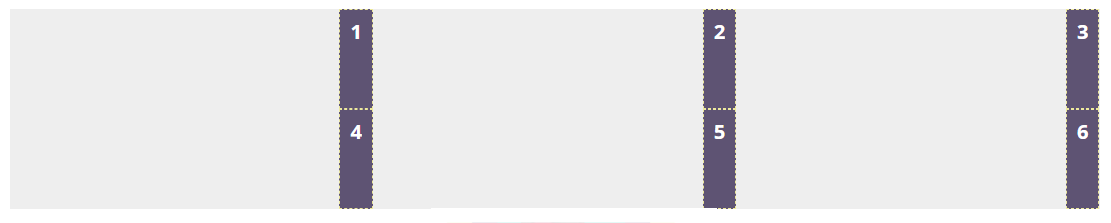
.grid {

   display: grid;

   grid-template: repeat(2, 100px) / repeat(3, 1fr);

   justify-items: end;

}



justify-items: **center**

aligns items in the center of their cell

выравнивает элементы по центру своей ячейки

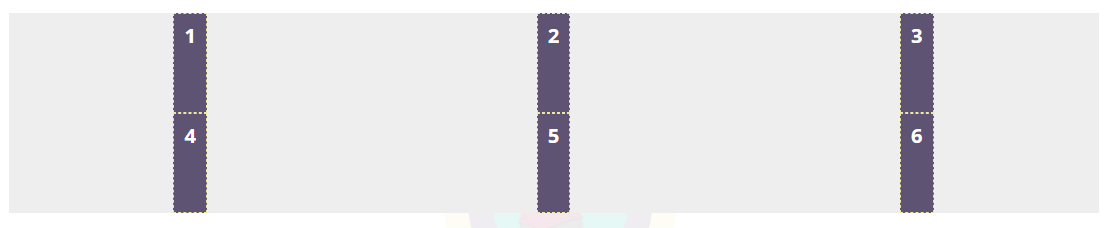
.grid {

   display: grid;

   grid-template: repeat(2, 100px) / repeat(3, 1fr);

   justify-items: center;

}



### *align-items*

*Align grid items along the vertical axis*Выровнять элементы сетки по горизонтальной оси

Aligns grid items along the horizontal axis  
Выравнивает элементы сетки по горизонтальной оси

This value applies to all grid items inside the container.

Это значение применяется ко всем элементам сетки внутри контейнера.

align-items: **stretch**;

resizes the grid items to allow the grid to fill the full height of the grid container. default value

изменяет размер элементов сетки, чтобы сетка заполняла всю высоту контейнера сетки. Значение по умолчанию

.grid {

   display: grid;

   height: 200px;

   grid-template-columns: repeat(3, 1fr);

   align-items: stretch;

}



align-items: **start**

Aligns grid items to the top of of their cell

Выравнивает элементы сетки по верхнему краю их ячейки

.grid {

   display: grid;

   grid-template: repeat(2, 100px) / repeat(3, 1fr);

   align-items: start;

}



align-items: **end**

aligns items to the bottom of their cell

выравнивает элементы по нижней части их ячейки

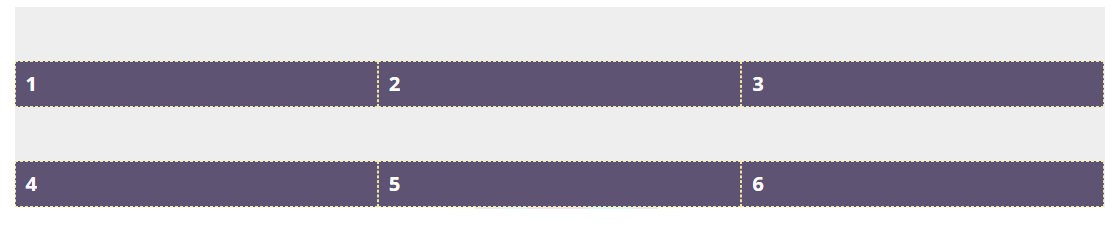
.grid {

   display: grid;

   grid-template: repeat(2, 100px) / repeat(3, 1fr);

   align-items: end;

}



align-items: **center**

aligns grid items in the center of their cell

выравнивает элементы сетки по центру своей ячейки

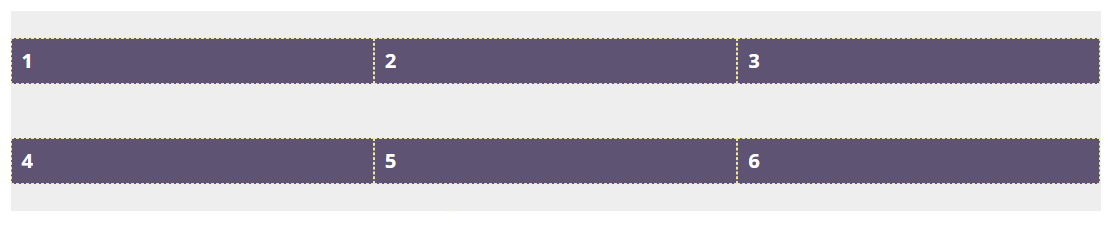
.grid {

   display: grid;

   grid-template: repeat(2, 100px) / repeat(3, 1fr);

   align-items: center;

}



#### column-gap row-gap grid-column-gap grid-row-gap

Specifies the size of the grid lines. You can think about it like setting the width of the space between the columns and rows.

Задает размер линий сетки. Вы можете думать об этом как об установке ширины промежутка между столбцами и строками.

Note: The grid- prefix will be removed and grid-column-gap and grid-row-gap renamed to column-gap and row-gap. The unprefixed properties are already supported in Chrome 68+, Safari 11.2 Release 50+ and Opera 54+.  
Примечание. Префикс grid- будет удален, а grid-column-gap и grid-row-gap переименованы в column-gap и row-gap. Свойства без префиксов уже поддерживаются в Chrome 68+, Safari 11.2 Release 50+ и Opera 54+.

column-gap row-gap

The gutters are only created between the columns/rows, not on the outer edges.

Промежутки создаются только между столбцами / строками, а не на внешних краях.

.grid {

   row-gap: 20px;

   column-gap: 10px;

}



#### gap grid-gap

gap: <grid-row-gap> <grid-column-gap>;

is a shorthand for [row-gap](https://css-tricks.com/snippets/css/complete-guide-grid/#prop-grid-column-row-gap) and [column-gap](https://css-tricks.com/snippets/css/complete-guide-grid/#prop-grid-column-row-gap)

это сокращение для row-gap и column-gap

.grid {

   gap: 20px 10px;

}



If you want to set the same spacing between columns and rows, you can write another shorthand like this.

Если вы хотите задать одинаковые промежутки между столбцами и строками, можно записать еще одно сокращение как это.

.grid {

   gap: 20px;

}

