



# CSS GRID LAYOUT

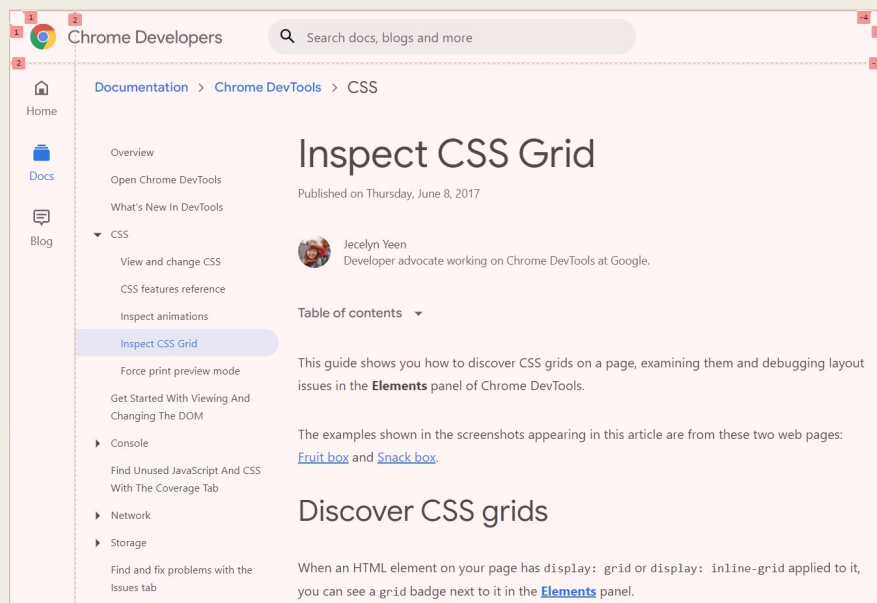
Двумерная сетка CSS



# Grid контейнер

**display: grid;**

Добавление данного стиля автоматически сделает всех прямых потомков этого элемента — grid элементами.

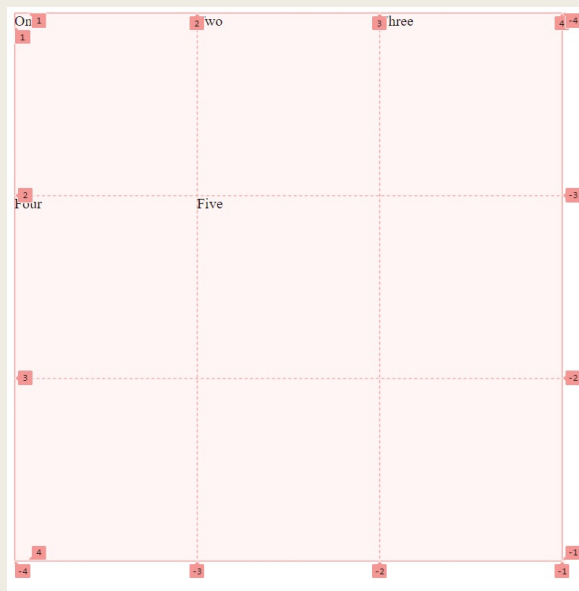


Отображение grid-элементов в браузере возможно в Chrome Dev Tools и Firefox Developer Edition (Grid Inspector)

# Grid Tracks

```
display: grid;  
grid-template-columns: 200px 200px 200px;  
grid-template-rows: 200px 200px 200px;
```

```
<div class="wrapper">  
  <div>One</div>  
  <div>Two</div>  
  <div>Three</div>  
  <div>Four</div>  
  <div>Five</div>  
</div>
```



Определение рядов и колонок в сетке.

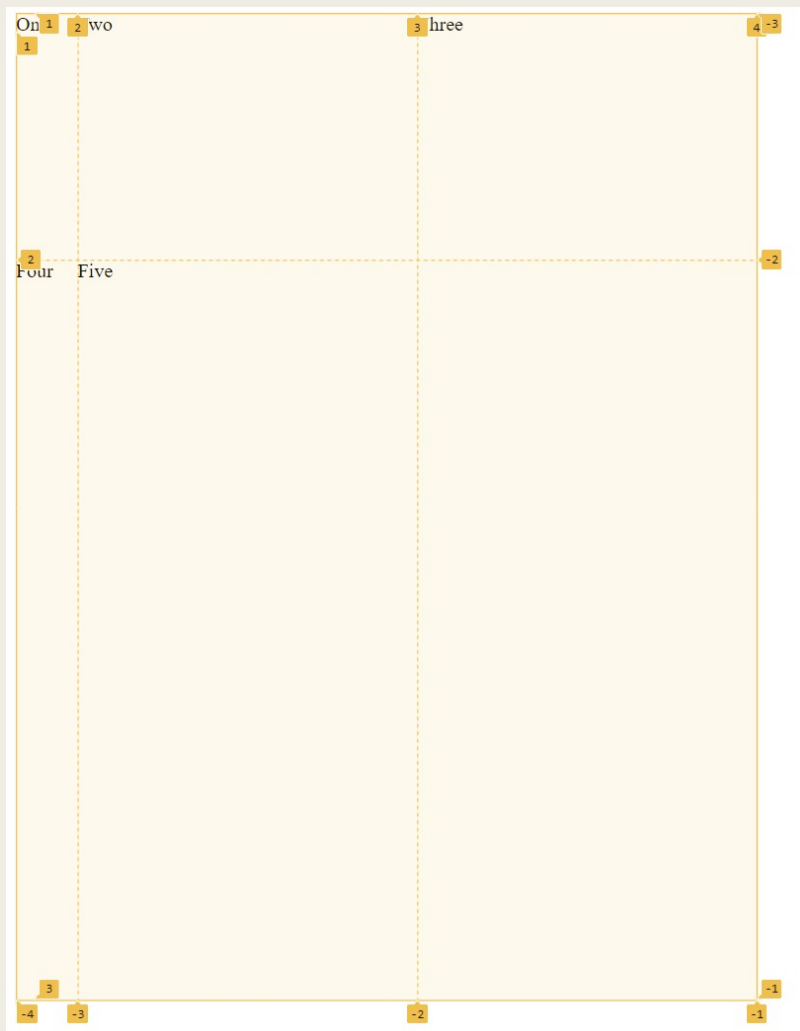
Отображение результата в браузере (Google Dev Tools).

# fr – fraction

```
<div class="wrapper">  
  <div>One</div>  
  <div>Two</div>  
  <div>Three</div>  
  <div>Four</div>  
  <div>Five</div>  
</div>
```

```
display: grid;  
width: 600px;  
height: 800px;  
grid-template-columns: 50px 1fr 1fr;  
grid-template-rows: 1fr 3fr;
```

Единица fr предназначена для создания эластичной grid-сетки и представляет собой долю доступного пространства в grid-контейнере.



# Функция repeat()

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

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

```
display: grid;  
grid-template-columns: 20px 1fr 1fr 10%;
```

```
display: grid;  
grid-template-columns: 20px repeat(2, 1fr) 10%;
```

Функция добавлена для быстрого создания сеток с колонками или строками одинаковой ширины (или высоты).

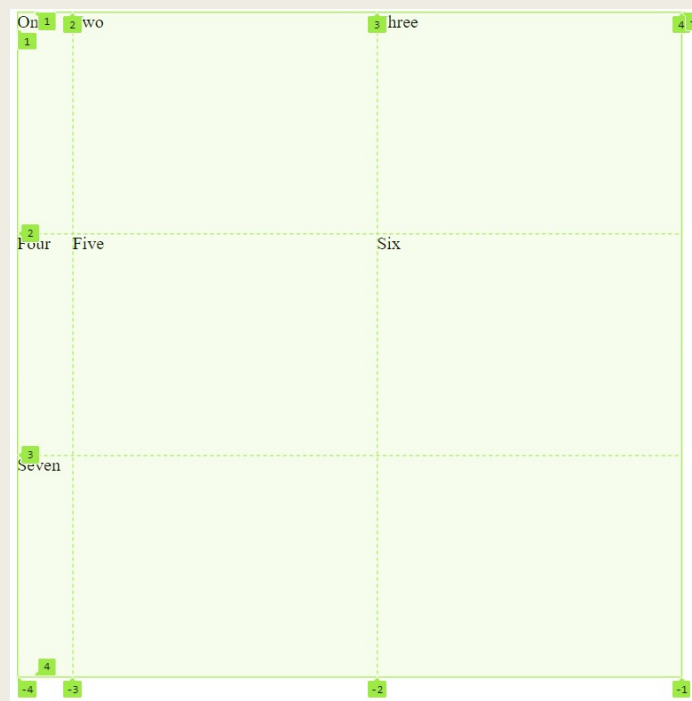
# Auto rows / auto columns

```
display: grid;  
width: 600px;  
height: 800px;  
grid-template-columns: 50px repeat(2, 1fr);  
grid-auto-rows: 300px;
```

```
<div class="wrapper">  
  <div>One</div>  
  <div>Two</div>  
  <div>Three</div>  
  <div>Four</div>  
  <div>Five</div>  
  <div>Six</div>  
  <div>Seven</div>  
</div>
```

Неявная горизонтальная сетка.

Мы не определяем точную высоту каждой строки, а задаём все строки одного размера.

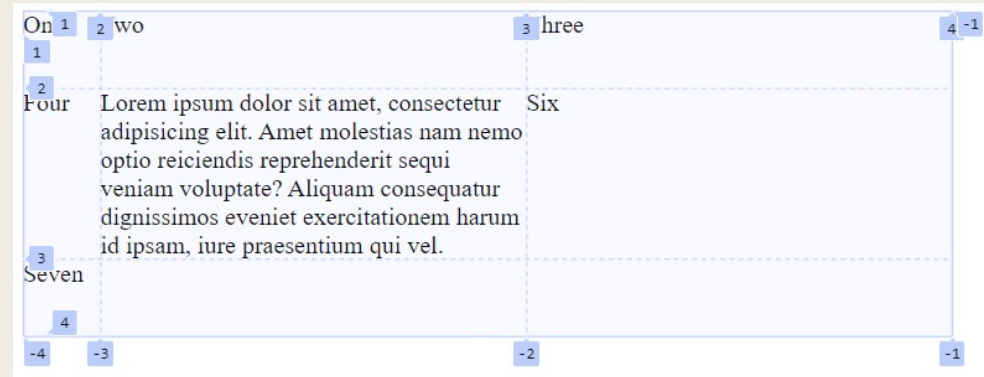


# Масштабирование с помощью minmax()

```
display: grid;  
width: 600px;  
grid-template-columns: 50px repeat(2, 1fr);  
grid-auto-rows: minmax(50px, auto);
```

```
<div class="wrapper">  
  
  <div>One</div>  
  
  <div>Two</div>  
  
  <div>Three</div>  
  
  <div>Four</div>  
  
  <div>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Amet  
molestias nam nemo optio reiciendis reprehenderit sequi veniam voluptate?  
Aliquam consequatur dignissimos eveniet exercitationem harum id ipsam, iure  
praesentium qui vel.</div>  
  
  <div>Six</div>  
  
</div>
```

С помощью функции minmax() можно установить наименьший и наибольший размер, который может принимать значение.



# Grid-линии

```
display: grid;
width: 600px;
grid-template-columns: repeat(3, 1fr);
grid-auto-rows: 100px;
```

При объявлении сетки, становятся доступными номера линий-разделителей (их мы видели ранее на изображениях).

На данные линии можно опираться при расстановке блоков.

```
<div class="wrapper">

  <div class="container-1">One</div>

  <div class="container-2">Two</div>

  <div class="container-3">Three</div>

  <div class="container-4">Four</div>

  <div class="container-5">Lorem ipsum
dolor sit amet, consectetur adipisicing elit. Amet
molestias nam nemo optio reiciendis reprehenderit
sequi veniam voluptate? Aliquam consequatur
dignissimos eveniet exercitationem harum id ipsam,
iure praesentium qui vel.</div>

  <div class="container-6">Six</div>

</div>
```

```
.container-1 {
  grid-column-start: 1;
  grid-column-end: 4;
  grid-row-start: 1;
  grid-row-end: 3;
}

.container-2 {
  grid-column-start: 1;
  grid-row-start: 3;
  grid-row-end: 5;
}
```





# Gutters

```
display: grid;
width: 600px;
grid-column-gap: 10px;
grid-row-gap: 1rem;
grid-template-columns: repeat(3, 1fr);
grid-auto-rows: 100px;
```

```
<div class="wrapper">

  <div class="container-1">One</div>

  <div class="container-2">Two</div>

  <div class="container-3">Three</div>

  <div class="container-4">Four</div>

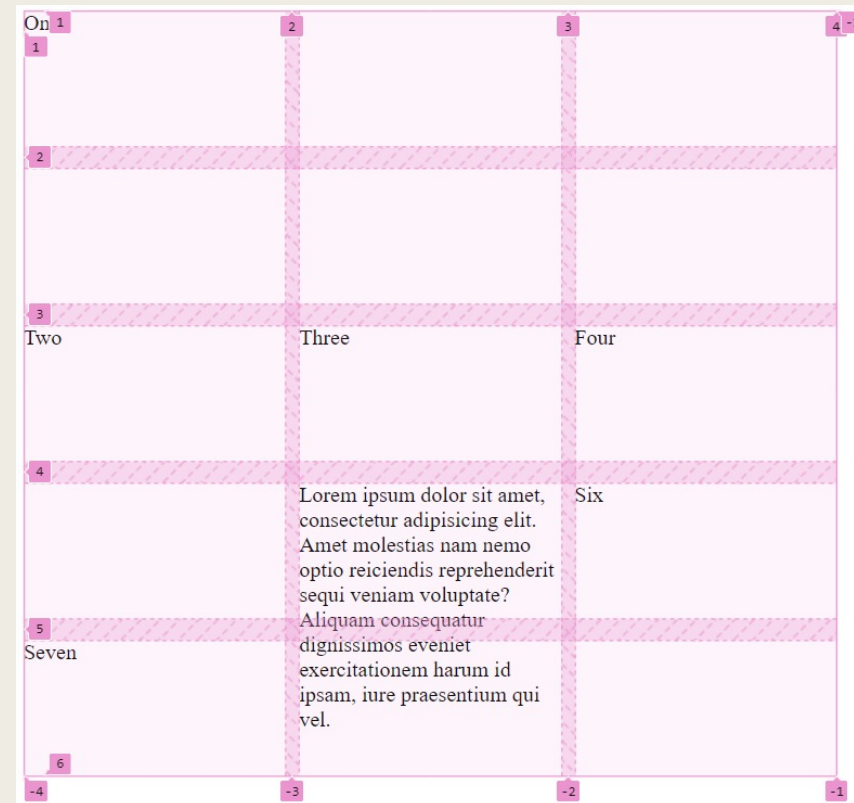
  <div class="container-5">Lorem ipsum dolor sit amet,
consectetur adipisicing elit. Amet molestias nam nemo optio reiciendis
reprehenderit sequi veniam voluptate? Aliquam consequatur dignissimos eveniet
exercitationem harum id ipsam, iure praesentium qui vel.</div>

  <div class="container-6">Six</div>

  <div class="container-7">Seven</div>

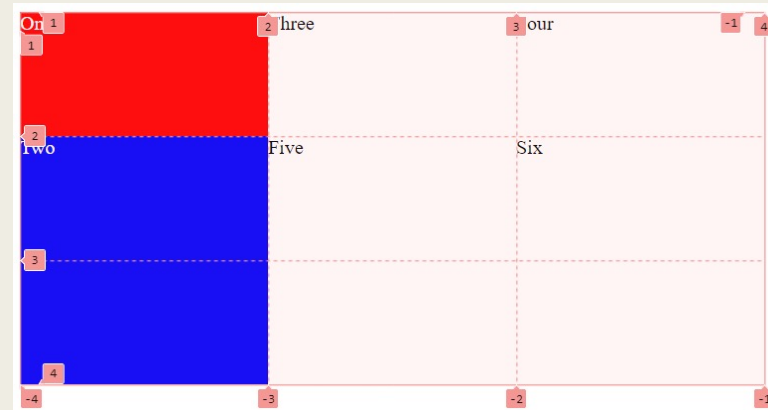
</div>
```

Расстояние между ячейками задаются с помощью свойств grid-column-gap или grid-row-gap (сокращенно – grid-gap).



# Grid-area

```
display: grid;
width: 600px;
grid-template-columns: repeat(3, 1fr);
grid-auto-rows: 100px;
```



```
<div class="wrapper">
  <div class="container-1">One</div>
  <div class="container-2">Two</div>
  <div class="container-3">Three</div>
  <div class="container-4">Four</div>
  <div class="container-5">Five</div>
  <div class="container-6">Six</div>
</div>
```

```
.container-1 {
  grid-area: 1 / 1 / 4 / 2;
  background: red;
  color: white;
}

.container-2 {
  grid-area: 2 / 1 / 4 / 2;
  background: blue;
  color: white;
}
```

Grid-area или область сетки – это прямоугольная область, которую можем расположить внутри grid-линий.

При пересечении областей регулируем расположение объектов с помощью z-index.

# Grid-area

Для упрощения использования grid-area стало возможным добавление названий областям.

С помощью grid-template-areas можно семантически обозначить ячейки, которые занимает область.

```
display: grid;
width: 600px;
grid-template-columns: repeat(3, 1fr);
grid-auto-rows: 100px;
grid-template-areas:
    «red red red»
    «green green yellow»
    «green green blue»
    «. . blue»;
```

```
.container-1 {
    grid-area: red;
    background: red;
    color: white;
}
```

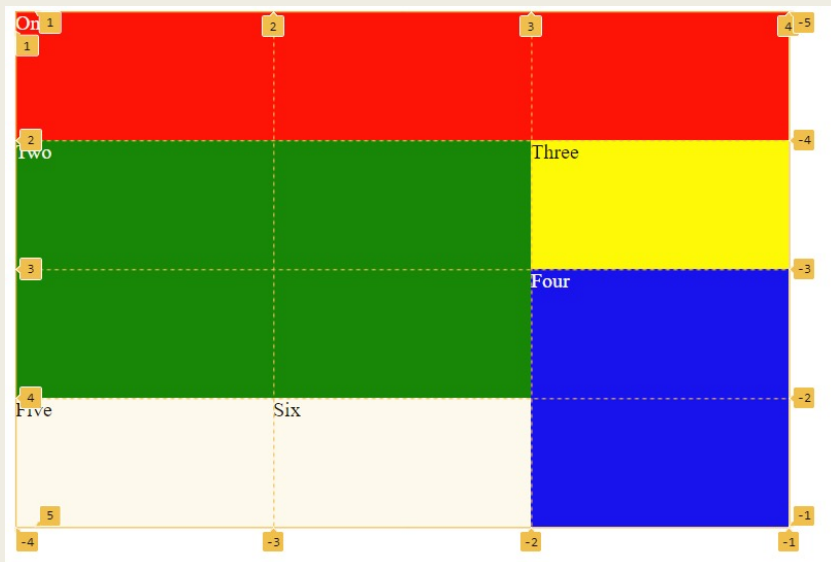
```
.container-2 {
    grid-area: green;
    background: green;
    color: white;
}
```

```
.container-3 {
    grid-area: yellow;
    background: yellow;
}
```

```
.container-4 {
    grid-area: blue;
    background: blue;
    color: white;
}
```

```
<div class="wrapper">
    <div class="container-1">One</div>
    <div class="container-2">Two</div>
    <div class="container-3">Three</div>
    <div class="container-4">Four</div>
    <div class="container-5">Five</div>
    <div class="container-6">Six</div>
</div>
```

# Grid-area



```
.container-1 {
```

```
  grid-area: red;
```

```
  background: red;
```

```
  color: white;
```

```
}
```

```
.container-2 {
```

```
  grid-area: green;
```

```
  background: green;
```

```
  color: white;
```

```
}
```

```
.container-3 {
```

```
  grid-area: yellow;
```

```
  background: yellow;
```

```
}
```

```
.container-4 {
```

```
  grid-area: blue;
```

```
  background: blue;
```

```
  color: white;
```

```
}
```

```
<div class="wrapper">
```

```
  <div class="container-1">One</div>
```

```
  <div class="container-2">Two</div>
```

```
  <div class="container-3">Three</div>
```

```
  <div class="container-4">Four</div>
```

```
  <div class="container-5">Five</div>
```

```
  <div class="container-6">Six</div>
```

```
</div>
```

# Полезные материалы

- [https://developer.mozilla.org/ru/docs/Web/CSS/CSS\\_Grid\\_Layout/Basic\\_Concepts\\_of\\_Grid\\_Layout](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout)
- [https://developer.mozilla.org/ru/docs/Web/CSS/CSS\\_Grid\\_Layout/Relationship\\_of\\_Grid\\_Layout](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Grid_Layout/Relationship_of_Grid_Layout)
- [https://developer.mozilla.org/ru/docs/Web/CSS/CSS\\_Grid\\_Layout](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Grid_Layout)
- [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Grid\\_Layout/Grid\\_Template\\_Areas](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Grid_Template_Areas)
- <https://goo-gl.ru/5hk2>
- <https://css-tricks.com/snippets/css/complete-guide-grid/>
- <https://medium.com/webbdev/grid-f81f2dd422bc>
- <https://dev.to/kenbellows/css-grid-areas-are-amazing-1gha>
- <https://ishadeed.com/article/grid-area/>
- <https://css-tricks.com/snippets/css/complete-guide-grid/>
- <https://gridbyexample.com/>