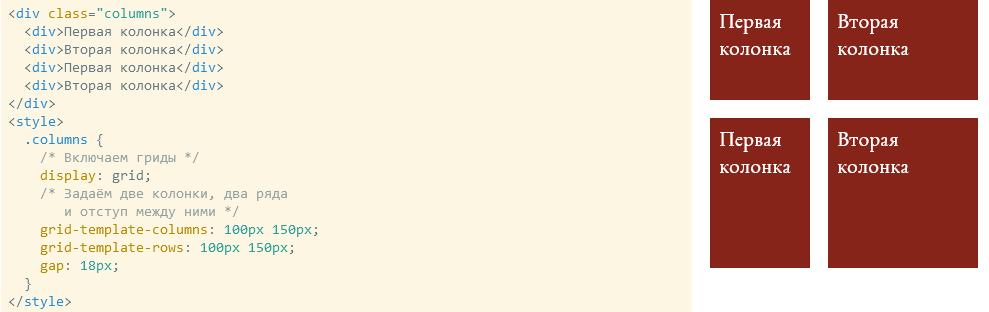
CSS grid layout или гриды — это набор свойств для раскладки элементов на странице или в отдельном элементе.

Чтобы включить гриды, нужно задать элементу display: grid и сетку: количество строк и столбцов, размеры и расстояния между ними. Элемент станет грид‑контейнером, а все вложенные в него элементы станут грид‑элементами и встанут по сетке:

<body>

    <div class="columns">

<div>Первая колонка</div>

        <div>Вторая колонка</div>

        <div>Первая колонка</div>

        <div>Вторая колонка</div>

      </div>

      <style>

        .columns {

          /\* Включаем гриды \*/

          display: grid;

          /\* Задаём две колонки, два ряда

             и отступ между ними \*/

          grid-template-columns: 100px 150px;

          grid-template-rows: 100px 150px;

          gap: 18px 6px;

        }

        .columns > div {

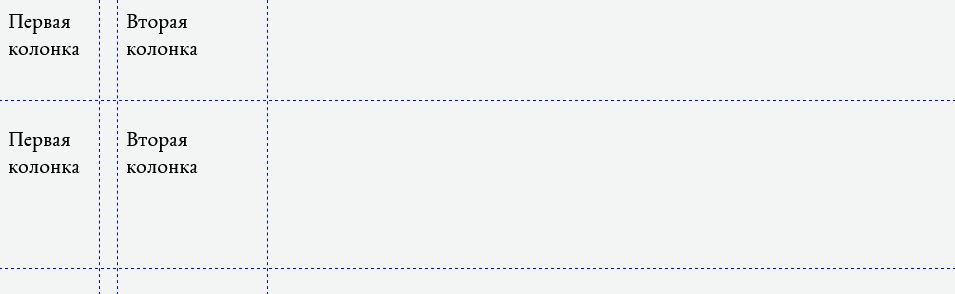
            background-color: aqua;

            padding: 9px 9px 18px;

        }

      </style>

Относитесь к гридам, как к прямоугольному формату, внутри которого можно определить направляющие, грид‑линии, задав таким образом колонки и строки, и разложить в получившиеся ячейки содержимое:



За вертикальные направляющие, а следовательно, за количество и размер колонок отвечает свойство grid-template-columns. Задаётся в любых величинах длины: пикселях, em, rem, vw, vh и процентах от ширины родителя:

Чтобы поддержать пропорциональные колонки, используют новую единицу измерения fr (от fraction). Работает также, как flex-grow во флексбоксах:

grid-template-columns: 1fr 1fr Две колонки поделят контейнер пополам

grid-template-columns: 2fr 1fr Две колонки поделят контейнер в соотношении 2:1

Единицы длины и fr можно смешивать. В этом случае пропорции посчитаются относительно свободного места, оставшегося после выделения фиксированных колонок:

.columns {

grid-template-columns: 1fr 100px 2fr;

}

Чтобы не дублировать описания одинаковых колонок, используют функцию repeat():

.columns {

grid-template-columns: repeat(12, 1fr);

/\* То же самое, что

grid-template-columns: 1fr 1fr ... 1fr \*/

gap: 18px;

}

Например, так можно сверстать расписание лекций курса, в котором в каждой строке должно быть 5 превьюшек одинакового размера:

|  |
| --- |
| style.css  .schedule {  grid-template-columns: repeat(5, 1fr);  /\* 18 пк между рядами, 9 пк между колонкамии\*/  gap: 18px 9px;  }  index.html  <div class="schedule">  <div><img src="..."></div>  <div><img src="..."></div>  <div><img src="..."></div>  <div><img src="..."></div>  <div><img src="..."></div>  <div><img src="..."></div>  <div><img src="..."></div>  <div><img src="..."></div>  <div><img src="..."></div>  <div><img src="..."></div>  <div><img src="..."></div>  <div><img src="..."></div> |

Помимо числа колонок функция repeat() умеет работать с ключевыми словами auto-fill и auto-fit. Они создают столько колонок, сколько поместится при текущей ширине грид‑контейнера:

style.css

.schedule {

grid-template-columns: repeat(auto-fill, 150px);

gap: 18px 9px;

}

Всё описанное выше справедливо и для рядов. За их количество и размер отвечает свойство grid-template-rows:

.columns {

/\* Первая строка получит высоту в 50 пк,

вторая — 20%, третья — всю оставшуюся

высоту контейнера \*/

grid-template-rows: 50px 20% auto;

}

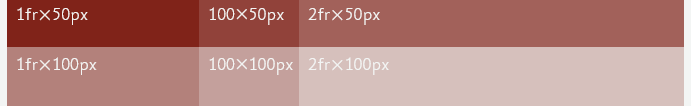


И конечно, одновременно можно задавать и колонки, и ряды:

.columns {

grid-template-columns: 1fr 100px 2fr;

grid-template-rows: 50px 100px;

}

## Неявно созданные колонки и ряды

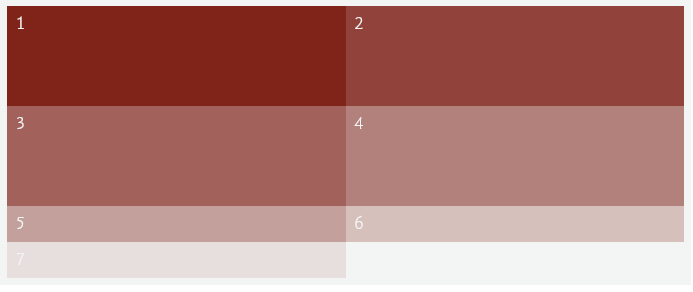
Если элементов внутри грид‑контейнера больше, чем доступных ячеек, то для них автоматически создадутся колонки и ряды:

style.css

.columns {

grid-template-columns: 1fr 1fr;

grid-template-rows: 100px 100px;

}

index.html

<div class="columns">

<div></div>

<div></div>

<div></div>

<div></div>

<div></div>

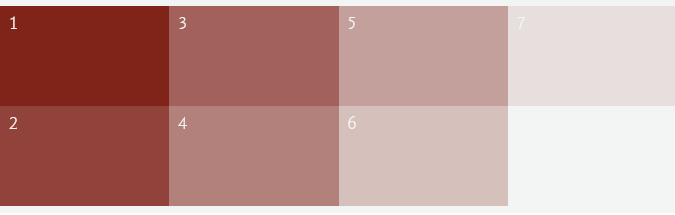
<div></div>

<div></div>

</div>

За их количество и размеры отвечают свойства grid-auto-columns и grid-auto-rows:

.columns {

grid-template-columns: 1fr 1fr;

grid-template-rows: 100px 100px;

/\* Пусть автоматические колонки

равномерно делят свободное пространство \*/

grid-auto-columns: 1fr;

/\* Пусть элементы раскладываются

по колонкам \*/

grid-auto-flow: column;

}

С помощью grid-auto-columns можно собрать адаптивные колонки, равномерно делящие доступное пространство:

style.css

.cols {

/\* Включаем гриды \*/

display: grid;

/\* Пусть неявные колонки

равномерно делят свободное пространство,

сохраняя минимальную ширину в 100 пк \*/

grid-auto-columns: minmax(100px, 1fr);

/\* Пусть лишние элементы выстраиваются

в колонки \*/

grid-auto-flow: column;

/\* Задаём межколонник в 18 пк \*/

gap: 18px;

/\* Растягиваем колонки по высоте \*/

height: 100%;

}

index.html

<div class="cols">

<div>Первая колонка</div>

<div>Вто­рая колонка</div>

<div>Тре­тья колонка</div>

<div>Чет­вёр­тая колонка</div>

</div>

