**1.** fr - гибкая единиц измерения, которая рассчитывается исходя из количества доступного пространства в grid-контейнере.

**2**. grid-template-columns: repeat(5, 20%);

grid-template-columns: 20% 20% 20% 20% 20%;

**3**. **auto-fill** - вместо указания количество колонок и сколько раз им повторяться, можно сказать браузеру, чтобы он уместил как можно больше колонок с учетом указанной длины. Строка автоматически заполняется таким количеством колонок, как это возможно с учетом заданной ширины.

auto-fill используется в связке с repeat () таким образом: grid-template-columns: repeat (auto-fill, 100px); - каждая строка заполнена возможным количеством колонок, учитывая их длину в 100px. Когда в строке не остается места, то элементы переходят на следующую строку.

**auto-fit** - различие с auto-fill появится тогда, когда мы укажем браузеру, что делать с оставшимся местом в строке. Если есть только одна строка и есть место для дополнительных колонок, но нет элементов, которые можно вставить эти колонки, то тогда он сделает ширину пустых колонок равную нулю. А если выставить максимальную ширину колонок на 1fr, то браузер разделит оставшееся место поровну между ними.

**4.** <style>

.container{

display: grid;

grid-template-columns: 100px 30% 1fr;

grid-gap: 1em;

grid-template-rows: 200px 100px;

}

.container > div{

padding: 1em;

background-color: rgb(70, 224, 224);

}

</style>

</head>

<body>

<div class="container">

<div class="item item1"></div>

<div class="item item2"></div>

<div class="item item3"></div>

<div class="item item4"></div>

<div class="item item5"></div>

<div class="item item6"></div>

</div>

</body>

**5.** <style>

.container{

display: grid;

justify-content: space-between;

grid-template-columns: 100px 80px 100px;

gap: 1%;

grid-template-rows: 80px 140px 80px;

}

.container > div{

padding: 1em;

background-color: rgb(70, 224, 224);

}

</style>

</head>

<body>

<div class="container">

<div class="item item1"></div>

<div class="item item2"></div>

<div class="item item3"></div>

<div class="item item4"></div>

<div class="item item5"></div>

<div class="item item6"></div>

<div class="item item7"></div>

<div class="item item8"></div>

<div class="item item9"></div>

</div>

</body>

**6.** Свойство grid-area определяет размер и расположение элемента сетки в макете сетки и является сокращенным свойством для следующих свойств:

* [grid-row-start](https://html5css.ru/cssref/pr_grid-row-start.php) - Указывает, в какой строке следует начать отображение элемента.
* [grid-column-start](https://html5css.ru/cssref/pr_grid-column-start.php) - Указывает, в каком столбце следует начать отображение элемента.
* [grid-row-end](https://html5css.ru/cssref/pr_grid-row-end.php) - Указывает, на какой строке строки прекратить отображение элемента, или сколько строк для Span.
* [grid-column-end](https://html5css.ru/cssref/pr_grid-column-end.php) -Указывает, на какой столбец-строку остановить отображение элемента или сколько столбцов для Span.
* Itemname- Задает имя элемента сетки
* grid-template-areas- определяет шаблон сетки ссылаясь на имена областей, которые заданы с помощью свойства grid-area

**7.** .container {

grid-template-columns: 50px 50px 50px 50px;

grid-template-rows: auto;

grid-template-areas:

'header header header header'

'main main . sidebar'

'footer footer footer footer';

}

получиться сетка из 4 колонок и 3 строк. Верхняя строка будет состоять из области header. Строка по середине будет состоять из области main, занимающей две колонки, пустой ячейки и области sidebar, которая занимает одну колонку. Последняя строка будет состоять только из области footer

**8.** auto-fit

**9.** Можно дать наименование каждой линии grid, присвоив ей какое-либо имя в квадратных скобках и затем, используя это имя, позиционировать элементы. Между именами указывается ширина столбца или высота строки, которые находятся между этими линиями.

Например:

grid-template-columns: [col1start] 1fr [col1end] 10px

[col2start] 1fr [col2end] 10px

[col3start] 1fr [col3end];

grid-template-rows: [row1start] 1fr [row1end] 10px [row2start] 1fr [row2end];

используя эти названия, позиционируем элементы между определенными линиями:

.special-item{

grid-column: col1start / col2end;

grid-row: row1start;

background-color: #bbb;}

**10.** grid-template-columns: repeat(12, 20%);