

Конспект «Микросетки. Начало».

Раздел 2

Список на флексах

Свойство `justify-content` со значением `space-between` заставляет первый и последний элемент прижиматься к границам контейнера. Но если в ряду всего два элемента, то свободного пространства между ними может оказаться слишком много. В этом случае лучше использовать `margin`.

Чтобы убрать лишний отступ у последнего элемента в ряду, используют псевдокласс `:nth-child`. Он позволяет выбрать дочерний элемент по его порядковому номеру:

```
// Выберет второй элемент с классом item
.item:nth-child(2) { ... }

// Выберет каждый второй элемент с классом item
.item:nth-child(2n) { ... }
```

Если не известно, какой элемент окажется в ряду последним, этот способ не работает.

repeat

Если все колонки в грид-контейнере должны быть одинаковой ширины, то удобно использовать значение-функцию `repeat`. В скобках после `repeat` указывают количество колонок и их ширину. Значения разделяют запятой:

```
grid-template-columns: repeat(количество колонок, ширина колонки);
```

auto-fit

Если количество колонок зависит от ширины контейнера, используют специальное значение `auto-fit`. Его указывают в скобках после `repeat` вместо числа колонок:

```
grid-template-columns: repeat(auto-fit, ширина колонки);
```

repeat (auto-fit, 100px)

```
width: 200px;
width: 300px;
width: 400px;
```

minmax

Чтобы ширина колонок изменялась пропорционально свободному пространству в контейнере, используют значение-функцию `minmax`.

Его указывают в `repeat` вместо фиксированной ширины колонок. В скобках после `minmax` задают минимальный и максимальный размеры колонок, они разделяются запятой:

```
repeat(auto-fit, minmax(минимальный размер, максимальный размер));
```

В `minmax` в качестве максимального значения часто используют единицу измерения `fr`. Она позволяет колонкам увеличивать ширину до тех пор, пока свободного пространства в контейнере не хватит на ещё одну колонку.

minmax (100px, 1fr)

```
width: 200px;
width: 299px;
width: 300px;
```

Свойства grid-column и grid-row

Чтобы растянуть элемент на несколько колонок используют свойство `grid-column`. Число после ключевого слова `span` указывает число колонок, которые элемент должен занять:

```
.element {
  grid-column: span 2;
}
```

Растянуть элемент на несколько рядов можно с помощью свойства `grid-row`. Ключевое слово `span` в нём означает количество рядов, которые элемент должен занять:

```
.long-element {
  grid-row: span 2;
```

```
}
```

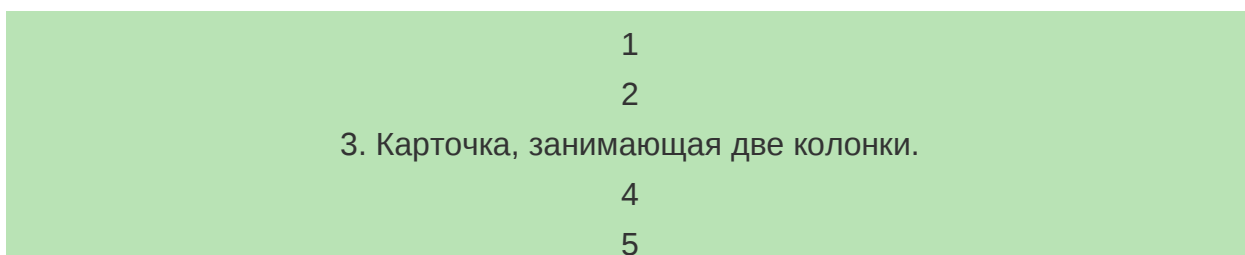
Свойства `grid-column` и `grid-row` можно использовать одновременно.

Свойство `grid-auto-flow`

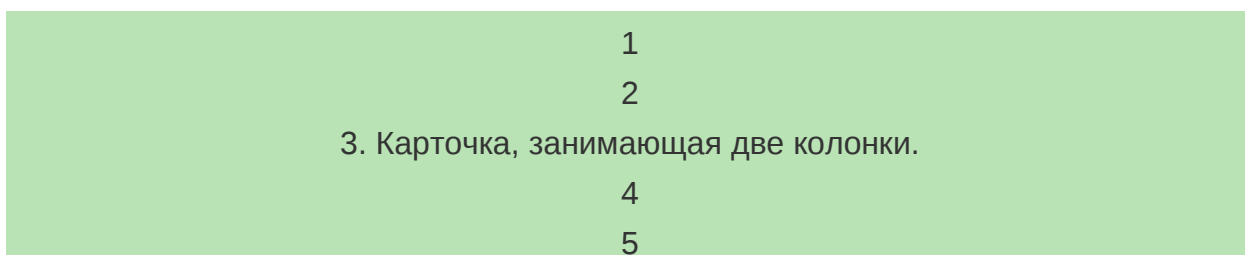
Свойство `grid-auto-flow` управляет автозаполнением грид-контейнера.

```
.grid-container {  
  display: grid;  
  grid-auto-flow: row;  
}
```

Значение по умолчанию `row` говорит располагать элементы в том порядке, в котором они идут в разметке, и при необходимости создавать новые ряды:



Но если указать значение `dense`, то контейнер будет заполняться так, чтобы не было пропусков:



Значение `dense` заставляет грид-контейнер заполнять пустые ячейки первым подходящим по размеру грид-элементом. При этом визуальный порядок на странице может отличаться от порядка элементов в разметке. Если порядок элементов важен, лучше это значение не использовать.