



УРОК №16-17

CSS Grid



Размеры

`minmax()` - минимальный и максимальный размер трека

`minmax(1fr, 200px)`

Имена треков

При создании строки или столбца сетки вы можете указать имя:

```
grid-template-row: [first] 200px [second] 300px
```

Сделает две строки с именами `first` и `second`

Имя можно будет использовать при указании места начала дочернего элемента

```
grid-row-start: first;
```

```
grid-row-end: second;
```

grid-template-areas

Для упрощения разметки сетки имена также даются и для областей. Для этого используется свойство `grid-template-areas`. Запись представляет собой визуальное представление сетки.

```
grid-template-areas: "header header header"  
                    "body body  
body"  
                    "body body  
body"  
                    ". footer ad";
```

grid-template-areas

В кавычках записывается строка Grid, внутри которых через пробел записываются имена областей. Повторяющиеся имена станут областью.

ВАЖНО! Область обязательно должна быть прямоугольной.

Для указания ячейки без области можно поставить точку (.)

Если не указывать `grid-template-rows` и `grid-template-columns`, то сетка генерируется на основе `grid-template-area`

grid-area

Применяется для элементов. Данное свойство используется в двух вариантах:

1. Как указание области для элемента из `grid-template-areas`
2. Как сокращение `grid-row-start / grid-column-start / grid-row-end / grid-column-end`

Выравнивание сетки

Следующие свойства по выравниванию применяются для ВСЕГО КОНТЕЙНЕРА

- `justify-items`: выравнивание по горизонтали для элементов сетки внутри ячейки
- `align-items`: выравнивание по вертикали для элементов сетки внутри ячейки
- `place-items`: сокращение `align-items` / `justify-items`

Значения: `start`, `end`, `center`, `stretch`

- `justify-content`: выравнивание треков по горизонтали сетки (контейнера)
- `align-content`: выравнивание треков по вертикали сетки (контейнера)
- `place-content`: сокращение `align-content` / `justify-content`

Значения: `start`, `end`, `center`, `stretch`, `space-between`, `space-around`, `space-evenly`

Размер автострок и автостолбцов

Когда ваши элементы появляются за пределами обозначенными сеткой, то добавляются дополнительные строки и столбцы. Вы можете указать размеры для таких строк и столбцов при помощи следующих свойств

- `grid-auto-columns`
- `grid-auto-rows`

Ссылка

<https://tuhub.ru/posts/css-grid-complete-guide>

CSS grid layout

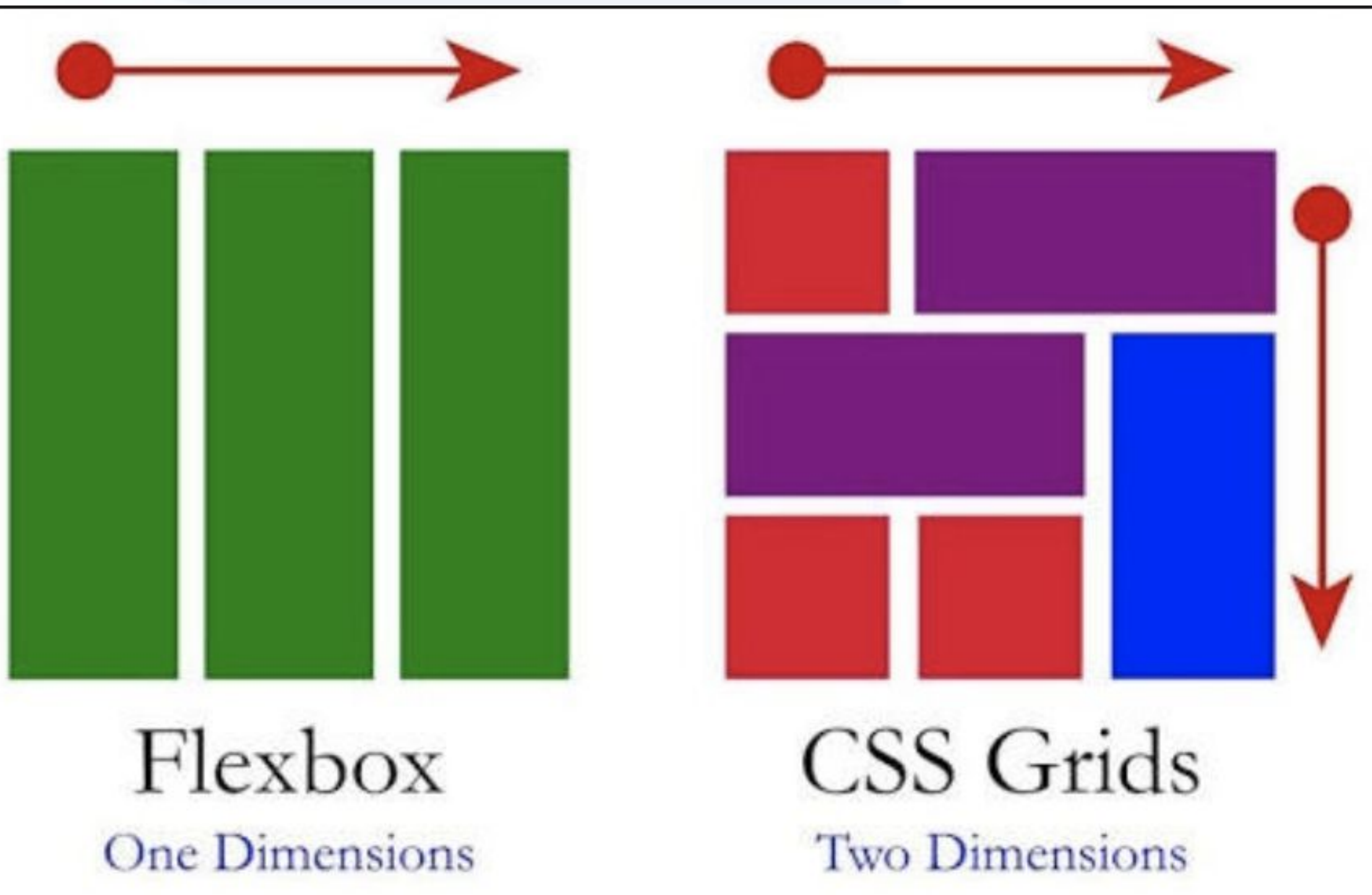


CSS grid layout

Ещё один способ расположения элементов.

Grid представляет собой пересекающийся набор горизонтальных и вертикальных линий - один набор определяет столбцы, а другой строки. Элементы могут быть помещены в сетку, соответственно строкам и столбцам. Строки и столбцы в Grid называют треками.

Для создания сетки нужно прописать для контейнера `display: grid` или `display: inline-grid`



grid tracks

Для задания столбцов и строк сетки нужно использовать свойства `grid-template-columns` и `grid-template-rows`. Эти свойства задаются для контейнера и задают ширину столбцов и высоту строк (в %, px или auto).

`grid-template-columns: 300px 400px 300px; //Сделает 3 столбца с заданными размерами.`

`grid-template-rows: 400px auto; //Сделает 2 строки, на первую выдадут 400 пикселей, на вторую - оставшееся место в контейнере.`

grid-template

- Свойство для сокращенной записи `grid-template-rows` и `grid-template-columns`

`grid-template: grid-template-rows / grid-template-columns`

fr и repeat

fr (fraction) - единица измерения для Grid треков, доля. Если для треков задать размеры в 1fr 2fr - то всё доступное пространство будет разделено на 2 части, причем второй трек будет в 2 раза больше чем первый.

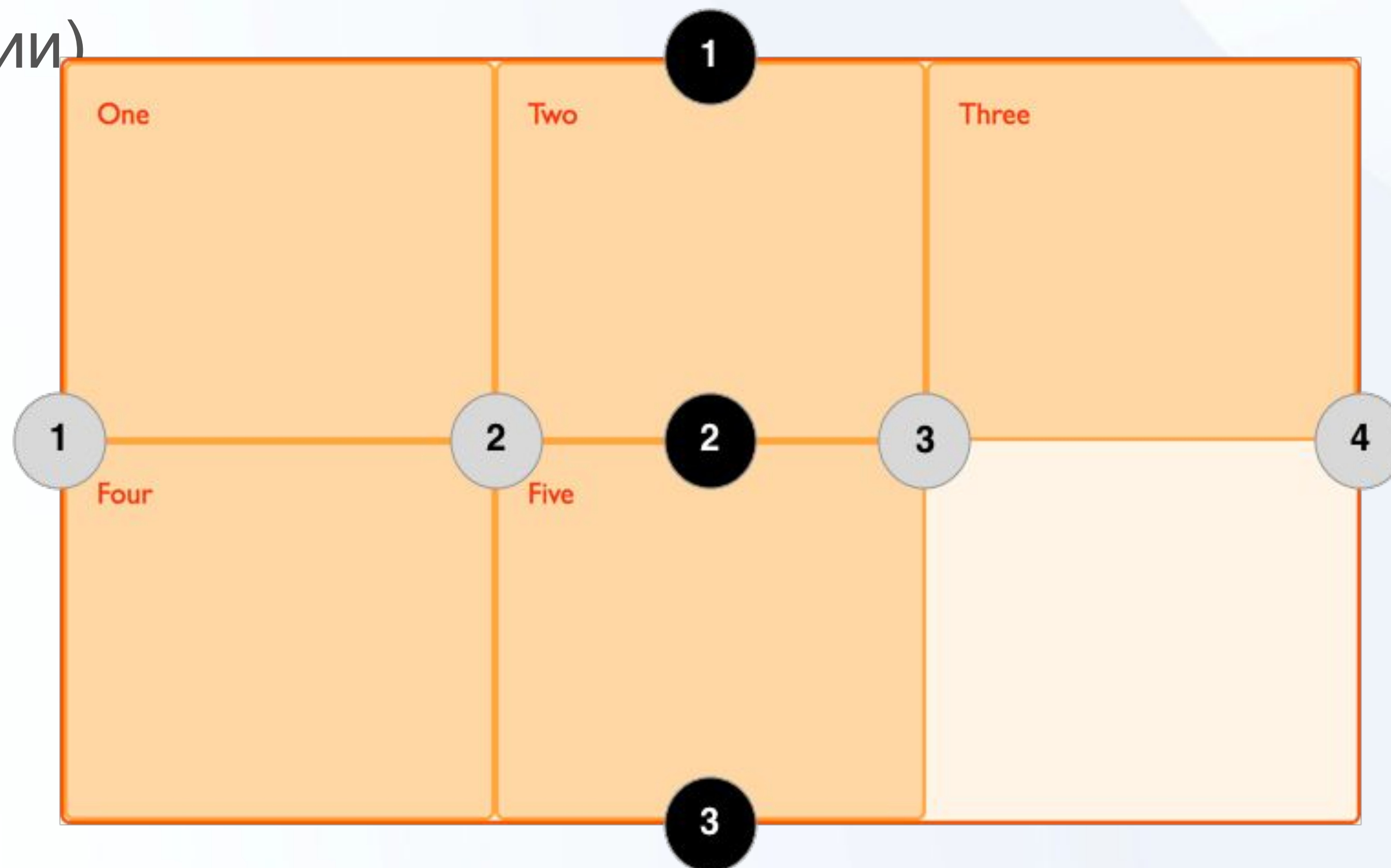
repeat - специальная запись для повторяющихся значений в треках.

```
repeat(3, 300px);
```

Аналогично с 300px 300px 300px

Grid-линии

При задании строк и столбцов создаются Grid-линии (линии нумерации)



Промежутки между треками

row-gap: расстояние между строками

column-gap: расстояние между столбцами

gap: row-gap column-gap - два свойства сразу

Расположение элементов

Свойства `grid-column-start`, `grid-column-end`, `grid-row-start` и `grid-row-end` задаются для дочерних элементов сетки и указывают расположение этих элементов.

Для `grid-column-start` и `grid-row-start` указывается номер линии начала, для `...-end` - номер линии конца

```
grid-column-start: 2;
```

```
grid-column-end: 3;
```

Элемент начнется со второй линии по вертикали и закончится на третьей (будет занимать один столбец).

Сокращенная запись и span

Для сокращения используются `grid-column` и `grid-row`, где начало и конец записываются через `/` (начало/конец, или `1/3`)

`span` можно использовать, если нужно указать размеры в треках.

Например, такая запись:

```
grid-row: 1 / span 2;
```

Будет указывать, что элемент начинается на линии 1 и занимает 2 строки.

Выравнивание элемента по ячейке

`justify-self`: выравнивание вдоль строки. Значения `start`, `end`, `center`, `stretch`

`align-self`: выравнивание вдоль столбца (вертикально).
Значения `start`, `end`, `center`, `stretch`

`place-self`: `align-self justify-self`

Два свойства вместе.

Дополнительные ссылки

<https://cssgridgarden.com/#ru>

https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout

<https://fls.guru/grid.html>

Конец

ПОСЛЕСЛОВИЕ

Давайте подведем итоги урока!
Чему мы научились? Что мы использовали?
К чему мы пришли?