

# HTML



# CSS



## HTML и CSS (CSS3 Grid Layout)

# CSS-СЕТКА

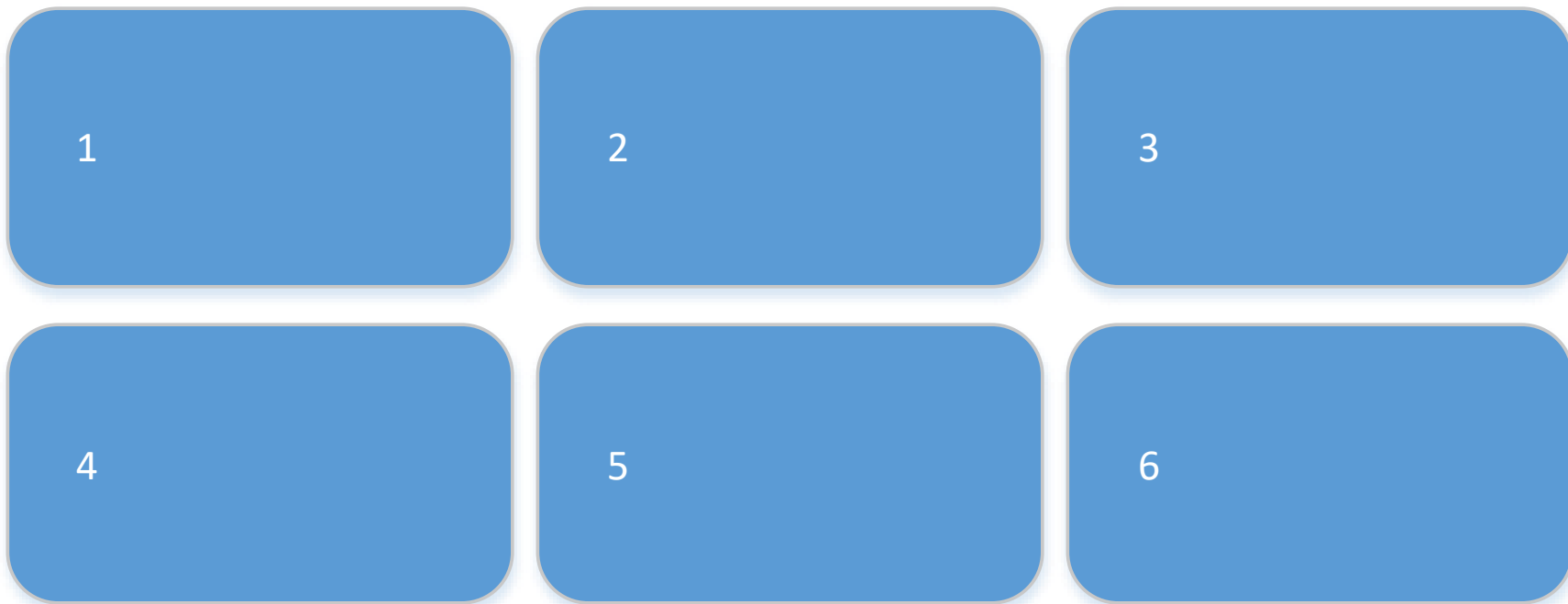
---

CSS-сетка позволяет определить двумерную разметку колонок и строк, а затем поместить в нее элементы. Некоторые из них заполняют только одну ячейку сетки, а другие способны расширяться и захватывать соседние колонки и строки. Можно задать размер сетки точно или позволить ей автоматически определять его по мере необходимости, чтобы уместить весь нужный контент.

Поддержка браузерами CSS-сеток началась в 2017 году.

# ПРИМЕР ПРОСТЕЙШЕЙ CSS-СЕТКИ

Проверим поддержку вашим браузером CSS-сеток.  
Для этого реализуем простейший пример:



# ПРИМЕР ПРОСТЕЙШЕЙ CSS-СЕТКИ

Код вёрстки:

Контейнер сетки

`<div class="grid">`

`<div class="1">1</div>`

`<div class="2">2</div>`

`<div class="3">3</div>`

`<div class="4">4</div>`

`<div class="5">5</div>`

`<div class="6">6</div>`

`</div>`

Потомки контейнера  
становятся элементами  
сетки

# ПРИМЕР ПРОСТЕЙШЕЙ CSS-СЕТКИ

Код стиля:

```
.grid {
```

```
  display: grid;
```

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

```
  grid-template-rows: 1fr 1fr;
```

```
  grid-gap: 10px;
```

```
}
```

```
.grid > * {
```

```
  background-color: darkgray;
```

```
  color: white;
```

```
  padding: 40px;
```

```
  border-radius: 10px;
```

```
}
```

Элемент становится  
контейнером сетки

Определяются три  
колонки  
одинаковой  
ширины

Определяются две  
строки одинаковой  
высоты

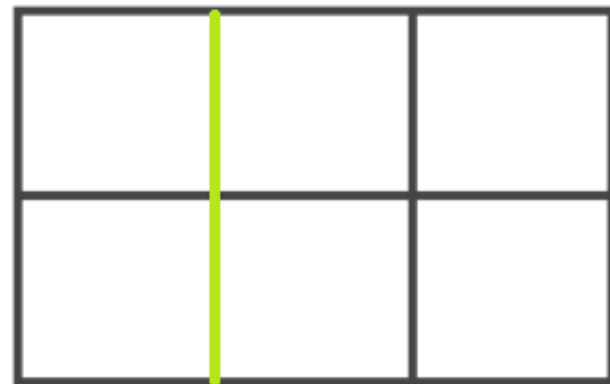
Создаются зазоры  
между всеми  
элементами сетки

# АНАТОМИЯ СЕТКИ

**Контейнер сетки** - элемент к которому применяется `display: grid`. Это прямой родитель для всех элементов сетки.

**Элемент сетки** - дочерние элементы (прямые потомки) контейнера.

**Линия сетки** формирует структуру сетки. Она может быть вертикальной или горизонтальной и лежать на любой из двух сторон строки или колонки.

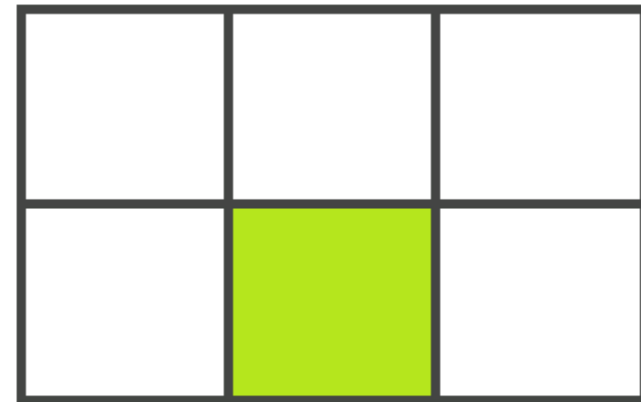


# АНАТОМИЯ СЕТКИ

**Полоса сетки** - расстояние между двумя соседними линиями сетки. У сетки есть горизонтальные (строки) и вертикальные (колонки) полосы.



**Ячейка сетки** - единичное пространство в сетке, где пересекаются горизонтальная и вертикальная полосы сетки.



# АНАТОМИЯ СЕТКИ

**Область сетки** - прямоугольный участок, составленный из одной или нескольких ячеек сетки. Область определяется между двумя вертикальными и двумя горизонтальными линиями сетки.





# СВОЙСТВА РОДИТЕЛЬСКОГО КОНТЕЙНЕРА

**display** - определяет элемент как контейнер и устанавливает новый контекст форматирования сетки для его содержимого.

Значения:

- ❖ **grid** - формирует сетку как блок (блочный элемент);
- ❖ **inline-grid** - формирует сетку как строчный элемент;
- ❖ **subgrid** - если ваш контейнер это ещё и элемент (вложенная сетка), то вы можете использовать это свойство для обозначения того, чтобы размеры строк/колонок были взяты из родительского элемента, а не определяли собственный.

# СВОЙСТВА РОДИТЕЛЬСКОГО КОНТЕЙНЕРА

## **grid-template-columns** и **grid-template-rows**

- определяет колонки и строки сетки с помощью списка значений разделённого пробелами. Значения представляют из себя размер полосы, а пробелы между ними представляют линии сетки (есть возможность задавать имена колонок в виде произвольного имени).

Например:

```
.container {  
  grid-template-columns: 40px 50px auto 50px 40px;  
  grid-template-rows: 25% 100px auto;  
}
```

Определяются пять колонок.

Определяются три строки.

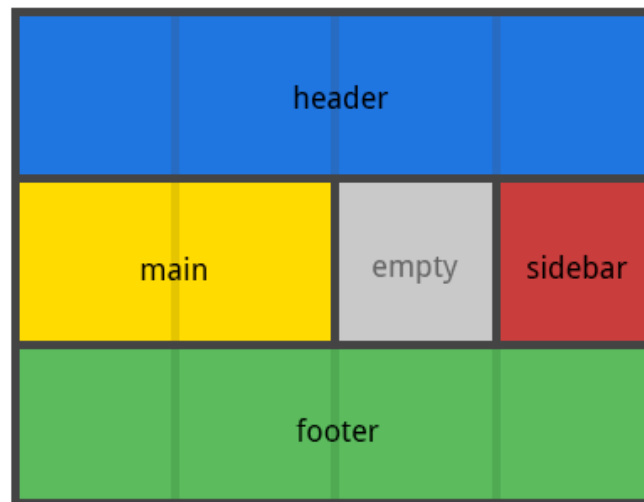
# СВОЙСТВА РОДИТЕЛЬСКОГО КОНТЕЙНЕРА

**grid-template-areas** - определяет шаблон сетки ссылаясь на имена областей, которые заданы с помощью свойства **grid-area**.

Значения:

- ❖ имя области заданное с помощью **grid-area**;
- ❖ . - точка обозначающая пустую ячейку;
- ❖ **none** - области не определены.

Рассмотрим пример:




# СВОЙСТВА РОДИТЕЛЬСКОГО КОНТЕЙНЕРА


Пример:

```
.container {  
  grid-template-columns: 50px 50px 50px 50px;  
  grid-template-rows: auto auto auto;  
  grid-template-areas:  
    "header header header header"  
    "main main . sidebar"  
    "footer footer footer footer";  
}
```

Определяются четыре колонки.



Определяется три строки.




Определяется шаблон сетки.



```
.header {  
  grid-area: header;  
}
```

Определяется название элемента, чтобы можно было ссылаться на него с помощью шаблона.



# СВОЙСТВА РОДИТЕЛЬСКОГО КОНТЕЙНЕРА

**grid-column-gap** и **grid-row-gap** - определяет размер ширины вертикальной и горизонтальной линий. Есть сокращение **grid-gap**.

Значения:

❖ значение размера.

Пример:

```
.container {
```

```
  grid-template-columns: 100px 50px 100px;
```

```
  grid-template-rows: 80px auto 80px;
```

```
  grid-column-gap: 10px;
```

```
  grid-row-gap: 15px;
```

```
}
```

Определяются три колонки.

Определяется три строки.

Ширина вертикальных линий.

Ширина горизонтальных линий.

# СВОЙСТВА РОДИТЕЛЬСКОГО КОНТЕЙНЕРА

**justify-items** - выравнивает содержимое вдоль оси **строки**. Это значение применяется ко всем элементам сетки внутри контейнера.

Значения:

- ❖ **start** - выравнивает содержимое по левой стороне области;
- ❖ **end** - выравнивает содержимое по правой стороне области;
- ❖ **center** - выравнивает содержимое по центру области;
- ❖ **stretch** - заполняет всю ширину области (по умолчанию).

# СВОЙСТВА РОДИТЕЛЬСКОГО КОНТЕЙНЕРА

**align-items** - выравнивает содержимое вдоль оси **столбца**. Это значение применяется ко всем элементам сетки внутри контейнера.

Значения:

- ❖ **start** - выравнивание элементов по верхней части области;
- ❖ **end** - выравнивание элементов по нижней части области;
- ❖ **center** - выравнивание элементов по центру области;
- ❖ **stretch** - заполняет всю высоту области (по умолчанию).

# СВОЙСТВА ДОЧЕРНЕГО ЭЛЕМЕНТА

**grid-column** и **grid-row** - определяют местоположение в сетке ссылаясь на конкретные линии.

Значения: номера линий, их названия или количество полос.

Пример:

```
.item-c {  
  grid-column: 3 / span 2;  
  grid-row: third-line / 4;  
}
```

Определяются  
местоположение от 3  
линии колонки + ещё две  
полосы по горизонтали.

Определяются  
местоположение от  
именованной линии  
(third-line) до 4 линии по  
вертикале.



# ИСТОЧНИКИ

---

С другими свойствами родительского контейнера и дочерних элементов можно ознакомиться:

[https://developer.mozilla.org/ru/docs/Web/CSS/CSS\\_Grid/Layout](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Grid/Layout) - CSS Grid Layout

<https://tuhub.ru/posts/css-grid-complete-guide> -  
Полное руководство по CSS Grid

# ПРАКТИКА

Реализуем  
макет:

