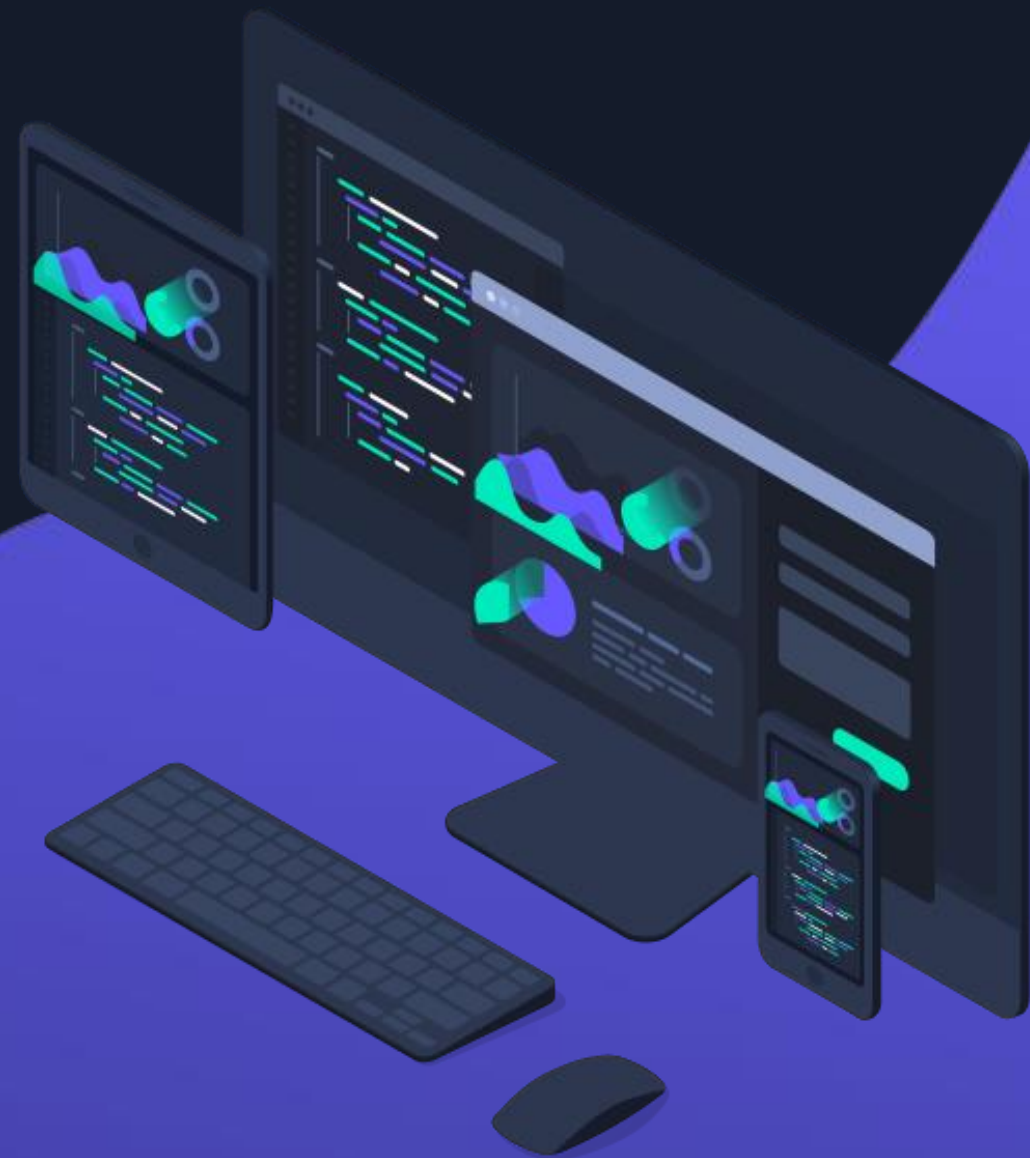


I T E A  
O N L I N E



# На сегодняшнем занятии:

1

Таблицы

2

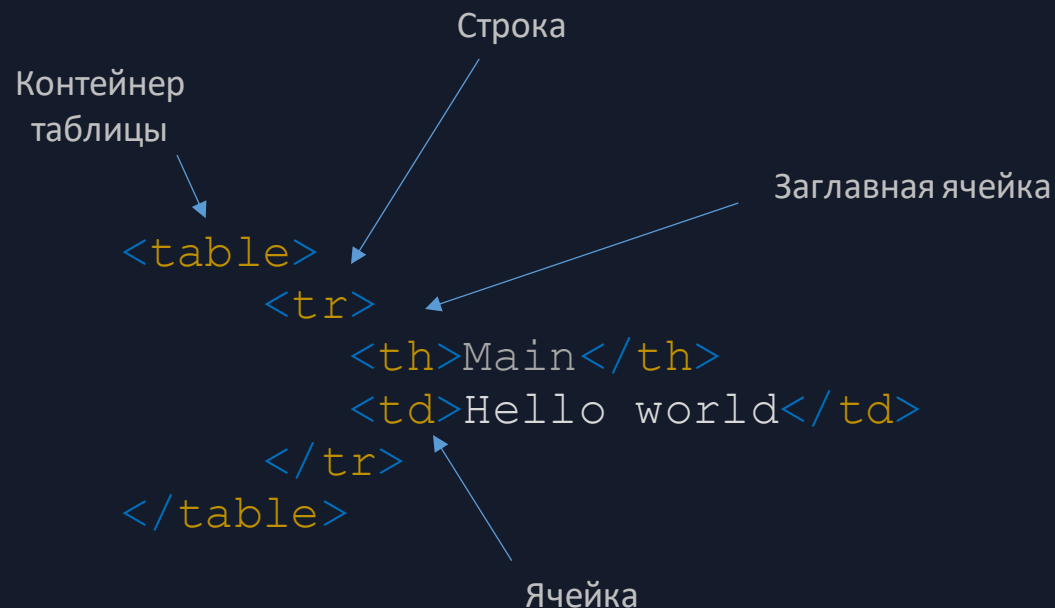
GRID Layout

# Таблицы

Элемент `<table>` служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек, которые задаются с помощью тегов `<tr>` и `<td>`.

Внутри `<table>` допустимо использовать следующие элементы:

```
<caption>
<col>
<colgroup>
<tbody>
<td>
<tfoot>
<th>
<thead>
<tr>
```



# Таблицы

Любая таблица состоит из строк и ячеек, которые задаются с помощью тегов `<tr>` и `<td>`.

Чтобы сделать ячейку заглавной, используется тег `<th>`.

```
<table border="1">
  <tr>
    <th>1</th>
    <td>2</td>
  </tr>
  <tr>
    <th>3</th>
    <td>4</td>
  </tr>
</table>
```

1	2
3	4

# Таблицы

Атрибуты таблицы

Атрибут	Назначение	Аналог в CSS
align	Выравнивание таблицы относительно родительского элемента	<code>margin: auto</code>
b bgcolor / background	фоновый цвет/рисунок в таблице	<code>background-image: url()</code> <code>background-color: #f00</code>
border	толщина рамки	<code>border-width: 1px</code>
bordercolor	цвет рамки	<code>border-color: blue</code>
cellpadding	отступ от рамки до содержимого ячейки	<code>padding: 10px</code> (для <code>td</code> )
cellspacing	расстояние между ячейками	<code>border-spacing: auto</code>
width / height	ширина/высота таблицы	<code>width: 1px; height: 1px</code>

# Таблицы

Для объединения двух и более ячеек в одну используются атрибуты `colspan` и `rowspan` тега `<td>`.

```
<table border="1">
  <tr>
    <th colspan="2">1</th>
    <td rowspan="2">2</td>
  </tr>
  <tr>
    <th>3</th>
    <td>4</td>
  </tr>
</table>
```

Количество ячеек, которое необходимо объединить

1		2
3	4	

# Таблицы

Теги для таблиц

Тег	Назначение
<caption></caption>	предназначен для создания заголовка к таблице и может размещаться только внутри контейнера <table> сразу после открывающего тега
<col>	задает характеристики одного или нескольких столбцов таблицы
<colgroup></colgroup>	задает характеристики одного или нескольких столбцов таблицы
<tbody></tbody>	предназначен для хранения одной или нескольких строк таблицы
<thead></thead>	предназначен для хранения одной или нескольких строк, которые расположатся вверху таблицы
<tfoot></tfoot>	предназначен для хранения одной или нескольких строк, которые расположатся внизу таблицы

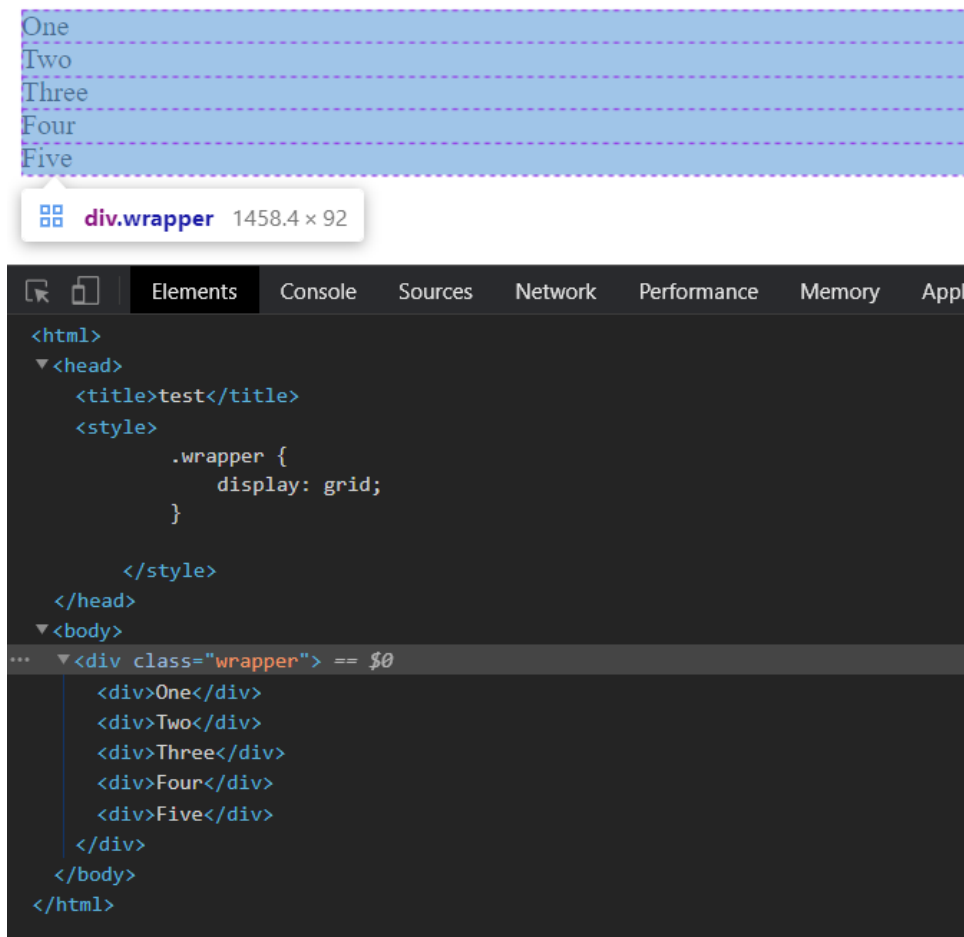


O N L I N E

# GRID Layout

CSS Grid Layout представляет двумерную сетку для CSS. Grid (здесь и далее подразумевается CSS Grid Layout ) можно использовать для размещения основных областей страницы или небольших элементов пользовательского интерфейса

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







O N L I N E

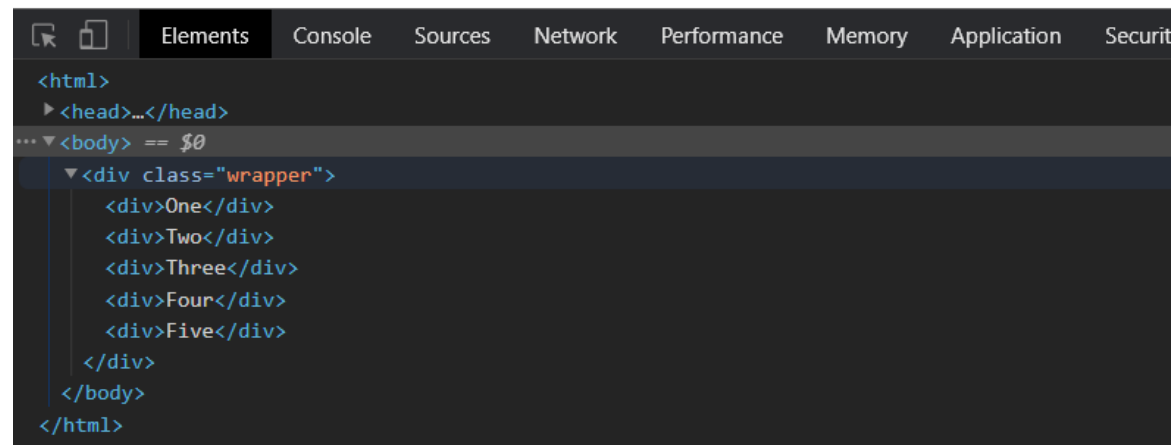
# Grid Tracks

Мы определяем ряды и столбцы в нашей сетке при помощи свойств `grid-template-columns` и `grid-template-rows`. Это определяет полосы сетки. Полоса сетки — это место между любыми двумя линиями сетки.

```
<style>
  .wrapper {
    display: grid;
    grid-template-columns: 200px 200px 200px;
  }
</style>
<div class="wrapper">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
  <div>Four</div>
  <div>Five</div>
</div>
```



div.wrapper 1458.4 × 36.8



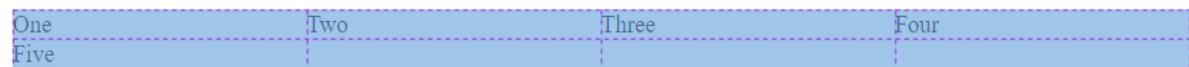


O N L I N E

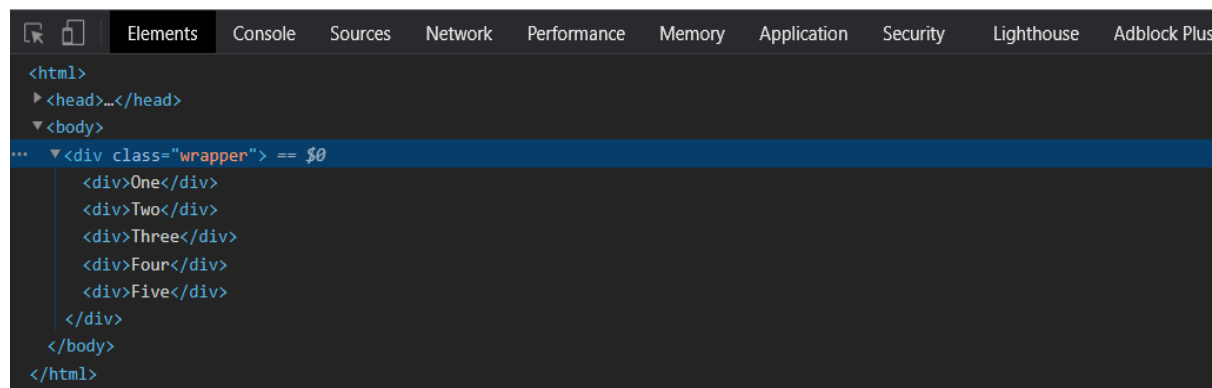
# Grid Tracks repeat()

В огромных гридах с большим количеством треков можно использовать нотацию `repeat()`, чтобы повторить структуру треков или ее часть.

```
<style>
  .wrapper {
    display: grid;
    grid-template-columns: repeat(4, 200px);
  }
</style>
<div class="wrapper">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
  <div>Four</div>
  <div>Five</div>
</div>
```



div.wrapper 1458.4 x 36.8



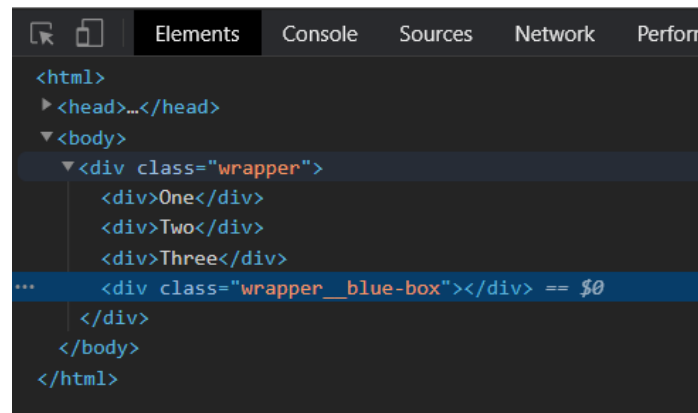
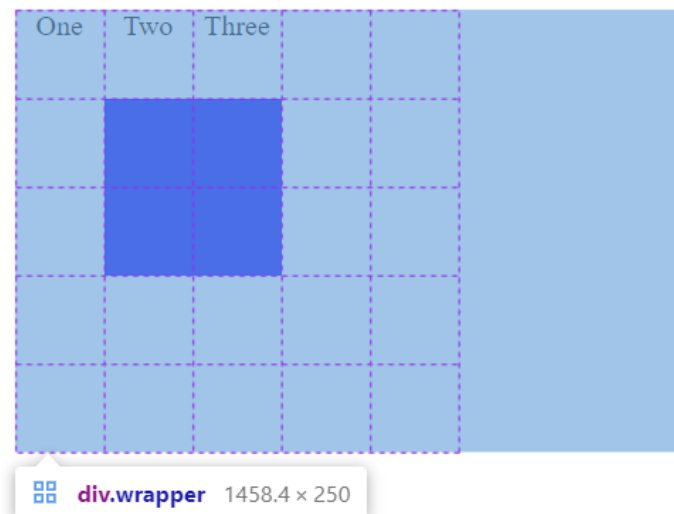


O N L I N E

# Grid Tracks

Размещение элементов по линиям  
мы размещаем первые два элемента в нашем  
гриде из трех колоночных треков с помощью  
свойств `grid-column-start`, `grid-column-end`,  
`grid-row-start` и `grid-row-end`.

```
<style>
.wrapper {
  display: grid;
  grid-template-columns: repeat(5, 50px);
  grid-template-rows: repeat(5, 50px);
  text-align: center;
}
.wrapper__blue-box {
  background-color: blue;
  grid-column-start: 2;
  grid-column-end: 4;
  grid-row-start: 2;
  grid-row-end: 4;
}
</style>
```





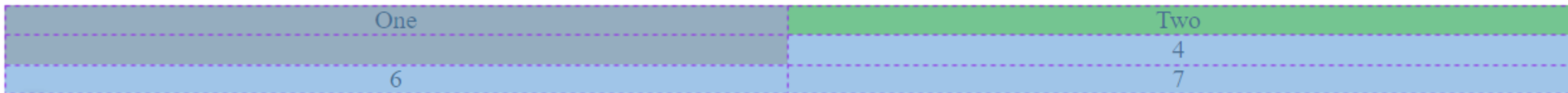
O N L I N E

# Grid Layout

## Объединение ячеек

Используя span мы не привязаны к тому, чтобы указать, где заканчивается область, вместо этого мы определяем, на сколько линий должен распространяться элемент

```
<style>
  .wrapper {
    display: grid;
    grid-column: 5;
    grid-template-columns: repeat(3, 1fr);
    text-align: center;
  }
  .item-1 {
    grid-row: 1 / span 2;
    background-color: burlywood;
  }
  .item-2 {
    background-color: chartreuse;
  }
</style>
```





O N L I N E

# Grid Layout

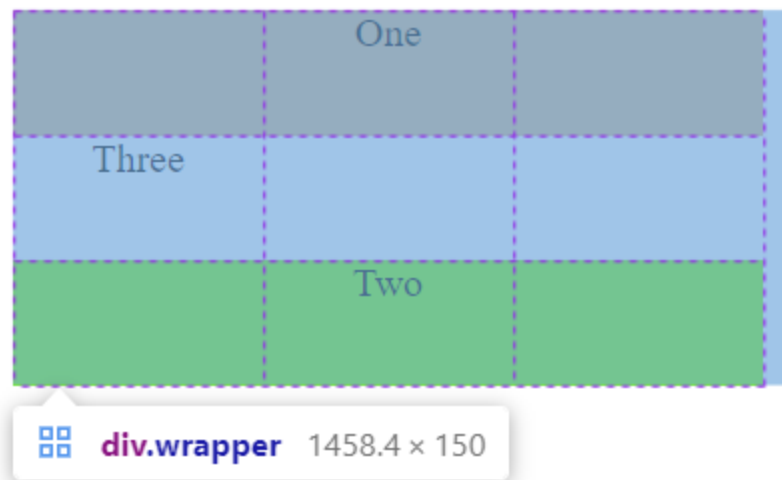
## Позиционирование элементов

Наш первый элемент заголовка, охватывая все три столбца. Второй элемент предназначен для основного контента, третий — подвал

```
grid-template-areas: "header header header"
                    "main  main  sidebar"
                    "footer footer footer";
```

```
.item-1 {
  background-color: burlywood;
  grid-area: header;
}
```

```
.item-2 {
  background-color: chartreuse;
  grid-area: footer;
}
```



```
<div class="wrapper">
  <div class="item-1">One</div>
  <div class="item-2">Two</div>
  <div>Three</div>
</div>
```



O N L I N E

---

# Q&A

I T E A  
O N L I N E

