

Макеты CSS Grid

Макеты

- Хронологически:
 - использовали таблицы,
 - обтекания (floats),
 - позиционирование и инлайновые блоки (inline-block),
 - Flexbox (но предназначен для «одномерных» макетов, а не для сложных двумерных).

Grid Layout

- Набор свойств CSS Grid Layout предлагает систему компоновки на основе сетки, состоящую из строк и столбцов, что упрощает проектирование веб-страниц без использования плавающих элементов и позиционирования.

Grid Layout

- 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.
- Поддержка браузерами

В 2020 году поддержка браузерами достигает 94 %

Grid Layout

- Элемент HTML становится **grid**-контейнером при установке свойства **display**
 - **grid** - формирует сетку как блок;
 - **inline-grid** - формирует сетку как инлайновый блок;

`<style>`

```
.grid-container {  
  display: grid;  
}
```

`</style>`

- Все прямые дочерние элементы сетки автоматически становятся элементами сетки.

Grid Layout

```
<body>  
<div class="row">  
  <div class="row__item header">  
    <h1>Header</h1> </div>  
  <div class="row__item nav">  
    <h1>Navbar</h1> </div>  
  <div class="row__item article">  
    <h1>Article</h1> </div>  
  <div class="row__item ads">  
    <h1>Ads</h1> </div>  
</div>  
</body>
```

Grid Layout

- .row

```
{ display: grid; margin: auto;
grid-template-rows: 60px 1fr ;
grid-template-columns: 20% 1fr 15%;
grid-gap: 10px;
width: 1000px; height: 1000px;
justify-items: center; justify-content:
space-between; grid-template-areas:
"header header header" "nav article ads"; }
```

- **grid-template-rows** — это CSS свойство, которое определяет названия линий и путь размера функции **grid rows**.

CSS свойство **grid-row** определяет с какой строки в макете сетки будет начинаться элемент, сколько строк будет занимать элемент, или на какой строке завершится элемент в макете сетки. Является сокращенным свойством для свойств **grid-row-start** и **grid-row-end**.

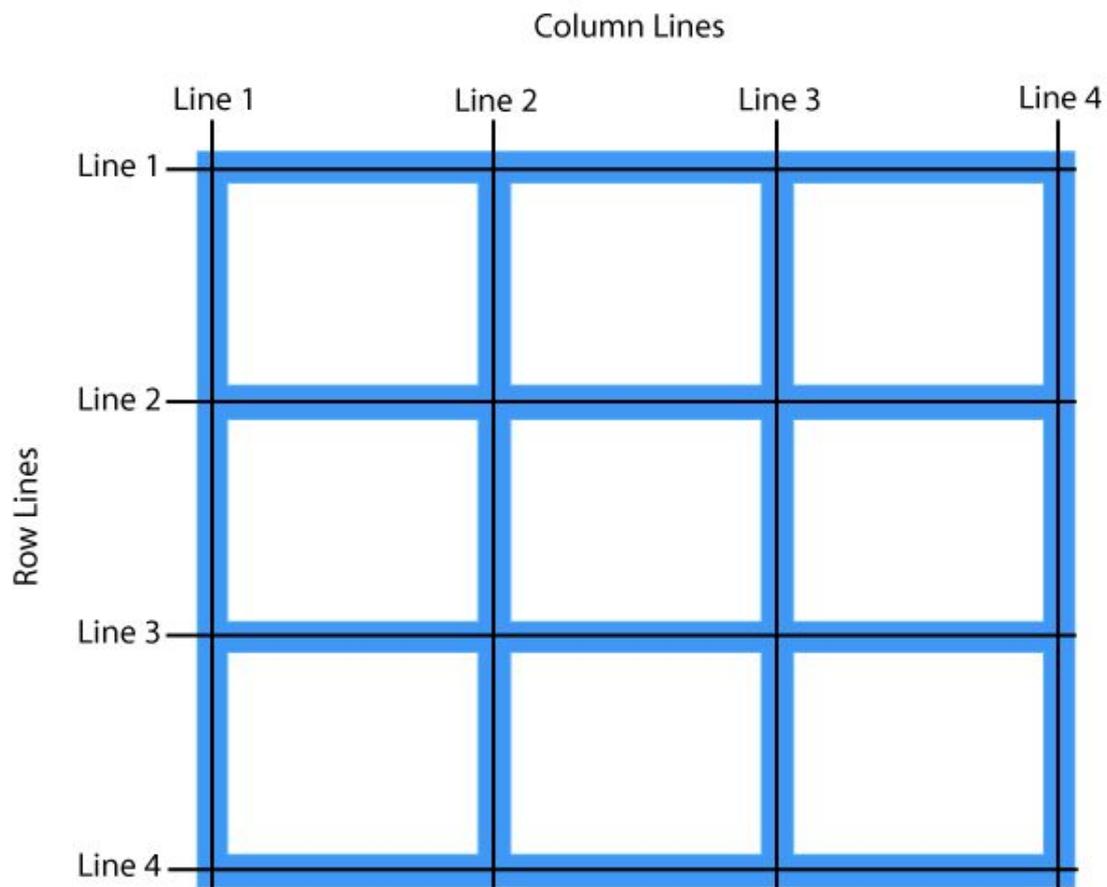
- Отступы между рядами и колонками сетки
 - `grid-column-gap`
 - `grid-row-gap`
 - `grid-gap` – свойство, определяющее желоба между строками и столбцами сетки.

```
.container{grid-gap: 10px 15px; }
```

Линия сетки

- Разделительные линии, составляющие структуру для сетки. Они могут быть вертикальными ("линии колонок") или горизонтальными ("линии строк") и располагаться по обе стороны от строки или столбца.

Линия сетки



Линия сетки

- Трек сетки
- Ячейка сетки
- Область сетки

Свойство `grid-template-areas`

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

Повторение названия области приводит к тому, что содержимое охватывает эти ячейки. Точка означает пустую ячейку. Сам синтаксис предоставляет визуализацию структуры сетки.

С помощью свойства **`grid-area`** мы можем назначить каждой из этих областей свое собственное имя. Именованное областей еще не создает никакого макета, однако теперь у нас есть именованные области, которые мы можем в нем использовать.

Свойство `grid-template-areas`

- `grid-template-columns`
- `grid-template-rows`
- Дополнительная единица измерения: "`fr` (fraction)".

СВОЙСТВО `grid-template-areas`

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

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

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

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

- Выравнивание по горизонтали:
- justify-content: start | end | center | stretch | space-around | space-between | space-evenly;
-
- Выравнивание по вертикали:
- align-content

- Выравнивание по горизонтали:
- justify-content: start | end | center | stretch | space-around | space-between | space-evenly;
-
- Выравнивание по вертикали:
- align-content

Создаем шаблон сайта с CSS Grid:



Grid:

Вы можете изменить шаблон просто перераспределив грид-области в `grid-template-areas`.

Таким образом, если мы сменим на это:

```
grid-template-areas: "nav header header" "nav article ads"; }
```

Grid:



Создаем простой CSS Grid размером 3 на 3 ячейки:

- Вот пример простого 3×3 грида с небольшими отступами между элементами.



A 3x3 grid of yellow cells, numbered 1 through 9. The cells are arranged in three rows and three columns, with small white gaps between them. The numbers are centered in each cell.

1	2	3
4	5	6
7	8	9

Создаем простой CSS Grid размером 3 на 3 ячейки:

- Заполняем `body` , добавляем 9 блоков внутрь блока с классом `grid`

```
<div id="grid">
```

```
<div>1</div>
```

```
<div>2</div>
```

```
<div>3</div>
```

```
<div>4</div>
```

```
<div>5</div>
```

```
<div>6</div>
```

```
<div>7</div>
```

```
<div>8</div>
```

```
<div>9</div>
```

```
</div>
```

Создаем простой CSS Grid размером 3 на 3 ячейки:

- **Заполняем style**

```
<style>
```

```
#grid { display: grid;  
grid-template-rows: 1fr 1fr 1fr;  
grid-template-columns: 1fr 1fr 1fr;  
grid-gap: 2vw; }  
#grid > div {font-size: 5vw;  
padding: .5em;  
background: gold;  
text-align: center; }
```

```
</style>
```

Создаем простой CSS Grid размером 3 на 3 ячейки:

- **Заполняем style**

```
<style>
```

```
#grid { display: grid; - Превращает элемент в grid контейнер.  
Это все, что нужно для того, что создать grid. Теперь у нас есть  
grid-контейнер и grid-элементы.
```

```
grid-template-rows: 1fr 1fr 1fr; - Выстраивает ряды в гриде.  
Каждое значение представляет размер ряда. В этом случае все  
значения равны 1fr (единица гибкости). для этого можно было бы  
использовать разные значения, такие как 100px, 7em, 30% и так  
далее.
```

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

```
grid-gap: 2vw; - Выставляет разрыв. То есть пробелы между  
grid элементами. Тут мы используем vw единицу, которая  
относительна ширине viewport, но также мы можем  
использовать 10px, 1em и т. д. }
```

```
</style>
```

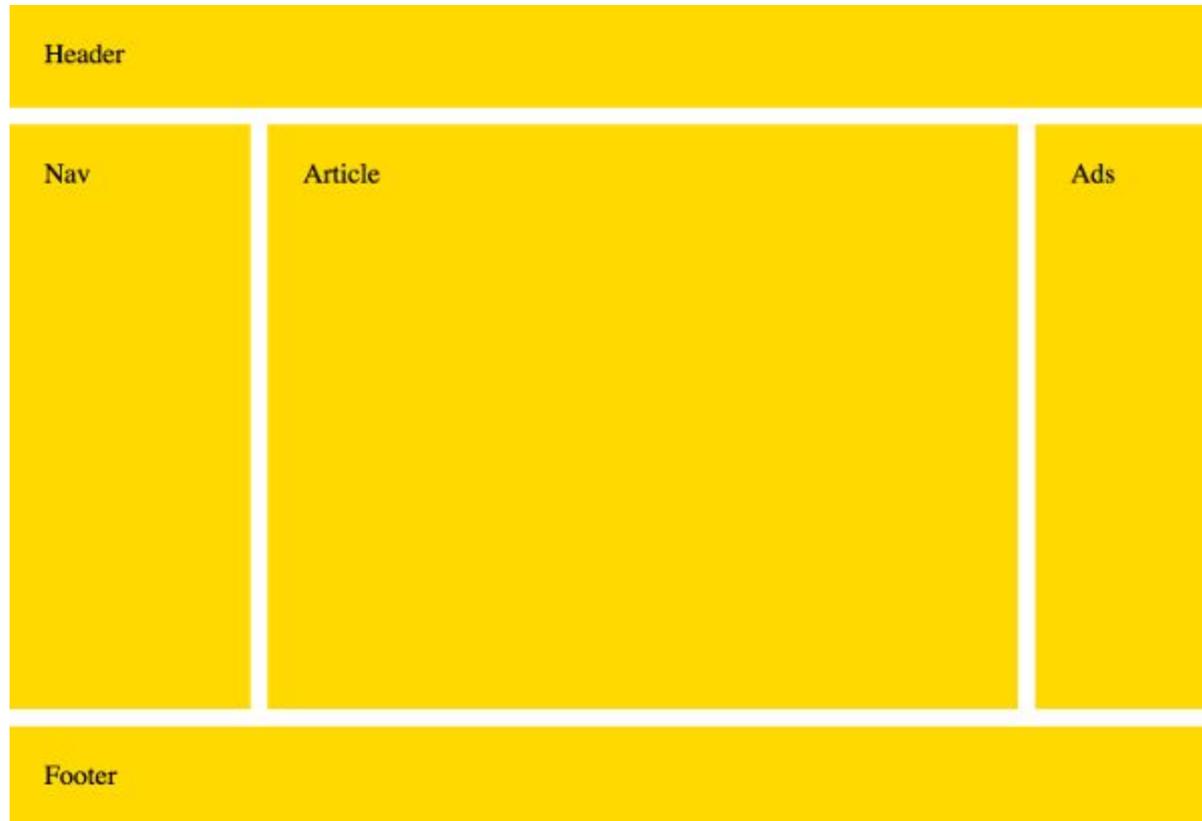
Создаем простой CSS Grid размером 3 на 3 ячейки:

- *Функция `repeat()`*
- Вы можете использовать функцию `repeat()` для повторяющихся объявлений значения размера элемента. Для примера, вместо того, чтобы делать это:
- `grid-template-rows: 1fr 1fr 1fr 1fr 1fr;` Мы можем сделать так:
- `grid-template-rows: repeat(5, 1fr);`

Создаем шаблон сайта с CSS Grid

- Гриды включают в себя интуитивный синтаксис, в котором вы можете виртуально *«видеть»* шаблон в коде, поэтому становится очень легко создавать и изменять сам шаблон.
- Даже значительные изменения могут быть сделаны за несколько секунд.
- Этот интуитивный синтаксис также помогает с адаптивным веб-дизайном. Создание разных шаблонов для разных устройств становится довольно несложной задачей при использовании грид разметки.

Создаем шаблон сайта с CSS Grid



Создаем шаблон сайта с CSS Grid

- Код шаблона. Заполняем `body`, используя теги `html 5`

```
<body>
```

```
<header
```

```
id="pageHeader">Header</header>
```

```
<article
```

```
id="mainArticle">Article</article>
```

```
<nav id="mainNav">Nav</nav>
```

```
<div id="siteAds">Ads</div>
```

```
<footer
```

```
id="pageFooter">Footer</footer>
```

```
</body>
```

Создаем шаблон сайта с CSS Grid

- Код шаблона. Прописываем стиль

```
<style>
```

```
body { display: grid; - делаем <body> грид-  
контейнером, таким образом все другие элементы станут  
грид-элементами.
```

```
grid-template-areas: "header header  
header"
```

```
"nav article ads"
```

```
"footer footer footer";
```

Этот кусок определяет наш шаблон. Просто смотря на код, мы можем видеть, что это 3x3 грид (три ряда и три колонки). Таким образом, у нас получается пять грид областей на девяти грид ячейках, так как некоторые наши грид-области занимают несколько ячеек.

Создаем шаблон сайта с CSS Grid

- Код шаблона. Прописываем стиль

```
<style>
```

```
body { display: grid;
```

```
grid-template-areas:
```

```
"header header header"
```

```
"nav article ads"
```

```
"footer footer footer";
```

```
grid-template-rows: 60px 1fr 60px; -
```

```
размеры строкам и колонкам
```

```
grid-template-columns: 20% 1fr 15%;
```

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

```
height: 100vh;
```

```
margin: 0;
```

```
}
```

Создаем шаблон сайта с CSS Grid

- Код шаблона. Прописываем стиль. Назначаем каждую из этих грид-областей каждому элементу:

```
#pageHeader {  
  grid-area: header; }  
#pageFooter {  
  grid-area: footer; }  
#mainArticle {  
  grid-area: article;}  
#mainNav {  
  grid-area: nav; }  
#siteAds {  
  grid-area: ads; } </style>
```

Свойство `grid-area` это сокращение свойства, которое позволяет вам размещать грид-элементы в гриде. В нашем случае, мы просто отсылаемся к названиям, которые мы уже предварительно указали в `grid-template-areas`.

Создаем шаблон сайта с CSS Grid

- Код шаблона. Прописываем стиль для основных тегов.

```
header, footer, article, nav, div {  
padding: 20px;  
background: gold; }
```

Изменяем шаблон сайта с CSS Grid

Изменяем шаблон

Вы можете изменить шаблон просто перераспределив грид-области в `grid-template-areas`.

Таким образом, если мы сменим на это:

```
grid-template-areas:  
"nav header header"  
"nav article ads"  
"nav footer ads";
```

Изменяем шаблон сайта с CSS Grid

То в результате получим такой шаблон:

