

УДК 004.514

ИСПОЛЬЗОВАНИЕ GRID CSS

Жданкин А. С., студент гр. ИТб – 151, IV курс
Алексеева Г.А., старший преподаватель
Научный руководитель: Чичерин И.В., к.т.н., доцент
Кузбасский государственный технический университет
имени Т.Ф. Горбачёва
г. Кемерово

Модуль Grid в CSS был разработан CSS Working Group для упрощения создания шаблонов в CSS. Основные браузеры начали его поддерживать в марте 2017 года [1].

Grid – это набор пересекающихся горизонтальных и вертикальных линий, которые задают размеры и позиционируют систему координат для контейнера в Grid контейнере.

Для создания Grid нужно просто выставить элементу `display: grid`; Это автоматически сделает всех потомков элемента – grid элементами.

Первым шагом является определение того, сколько будет колонок и рядов в создаваемом элементе.

На Рисунок 1 приведен пример с 3 рядами и 3 колонками. Он состоит из 9 Grid элементов, все объекты на рисунке – одного размера, но размер можно задать по желанию. Одни элементы могут охватывать несколько столбцов и рядов, другие – оставаться размеров с 1 ячейку.

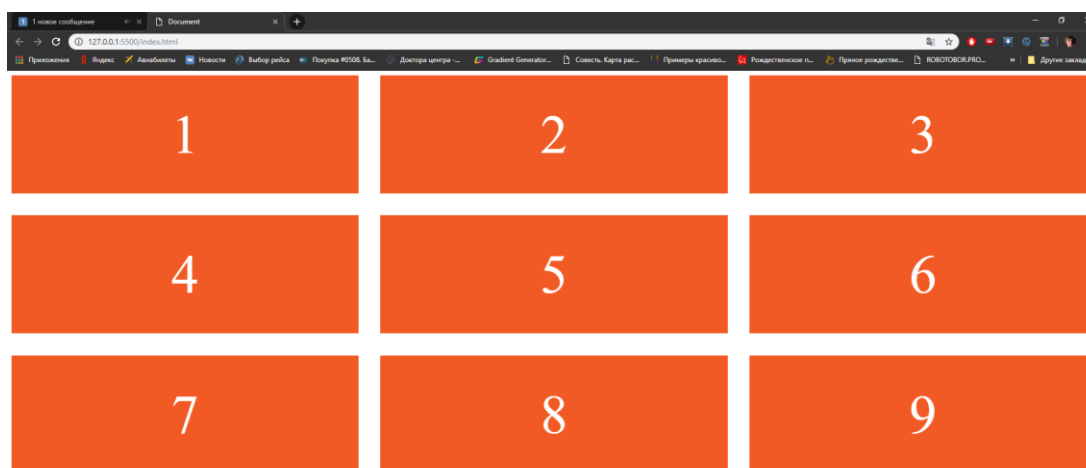


Рисунок 1

На Рисунок 2 представлен пример HTML разметки для CSS Grid: Это обычный снипет HTML, состоящий из элементов, вложенных во внешний элемент. В данном примере, внешний `div` это контейнер гридов, а элементы, вложенные в него, считаются Gridi элементами.

```
<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>
```

Рисунок 2

Но созданный элемент не может считаться полноценным гридом, пока не применен CSS. На рисунке 3 – CSS написан на SASS. Правила применяются к внешнему элементу div, которому задан id = "grid".

```
#grid
  display: grid
  grid-template-rows: 1fr 1fr 1fr
  grid-template-columns: 1fr 1fr 1fr
  grid-gap: 2vw
```

Рисунок 3

Объявление, приведенное на Рисунок 4, выстраивает ряды в гриде, каждое значение это размер ряда, размер можно указывать в пикселях (px), процентах (%), относительно высоты шрифта элемента (em), в данном случае это fr (единица гибкости – специальная единица измерения для grid).

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

Рисунок 4

На Рисунок 5 показано аналогичное объявление для колонок.

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

Рисунок 5

Следующее объявление (Рисунок 6) задаёт размер разрыва между Grid элементами, это сокращение для grid-row-gap и grid-column-gap.

```
grid-gap: 2vw
```

Рисунок 6

Следующее правило (Рисунок 7) отвечает за обычное назначение стилей.

```
& > div
font-size: 5vw
padding: .5em
background: #F15A24
color: #fff
text-align: center
```

Рисунок 7

В объявлении может использоваться функция `repeat()`, которая служит для указания размера повторяющихся элементов, например запись на рисунке 4 может быть реализована, как показано на Рисунок 8. Это позволяет сократить объем кода, который необходимо написать [1].

```
grid-template-rows: repeat(5, 1fr)
```

Рисунок 8

На Рисунок 9 продемонстрирован шаблон сайта с использованием CSS Grid.

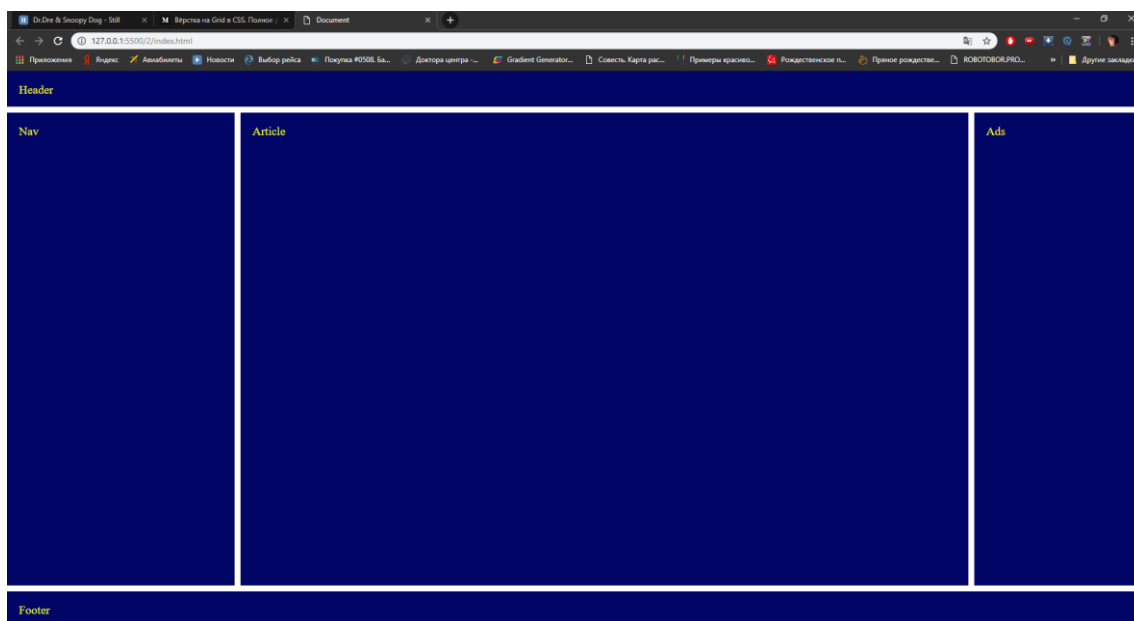


Рисунок 9

Можно виртуально видеть шаблон в коде, поэтому его легко создавать и изменять, а это значительно упрощает адаптивную вёрстку.

Пример HTML разметки представлен на Рисунок 10. В данном примере `body` сделан грид контейнером, а все другие элементы станут грид элементами.

```
<header class="header">Header</header>  
<article class="article">Article</article>  
<nav class="nav">Nav</nav>  
<div class="ads">Ads</div>  
<footer class="footer">Footer</footer>
```

Рисунок 10

ASCII график (для удобства можно его записывать вертикально) показан на Рисунок 11. Это определяет шаблон, где прослеживаются 3x3 грид, и в итоге получаются 5 грид областей на 9 грид ячейках, так как некоторые области занимают несколько ячеек. Изменить шаблон можно, просто перераспределив грид области.

```
grid-template-areas: "a a a" "b c d" "e e e"
```

Рисунок 11

Назначим каждую из грид областей (Рисунок 12). Свойство grid-area это сокращение свойства, которое позволяет размещать грид элементы в гриде. Происходит отсылка к названиям, которые предварительно указаны в grid-template-areas.

```
.header  
    grid-area: a  
.article  
    grid-area: c  
.nav  
    grid-area: b  
.ads  
    grid-area: d  
.footer  
    grid-area: e
```

Рисунок 12

Пример задания размеров строк и колонок можно увидеть на Рисунок 13.

```
grid-template-rows: 60px 1fr 60px  
grid-template-columns: 20% 1fr 15%
```

Рисунок 13

CSS Grid это двумерные макеты, а например Flexbox – одномерные. Если объект раскладывается в одном направлении, например ссылка, то применим flexbox. Это даст больше гибкости, чем в случае CSS Grid и в дальнейшем с этим будет проще работать. Однако если мы создаём целый макет в

двух измерениях, как с row и column то CSS Grid просто необходим. В этом случае достигается большая гибкости, а разметка становится проще. И в дальнейшем будет проще выполнить адаптации планшетов и телефонов [3].

На данный момент существует проблема с поддержкой CSS Grid браузерами (Рисунок 14). К браузерам, поддерживающим CSS Grid, относятся Chrome, Firefox, Safari. Edge поддерживает CCS Grid, но может отображать её некорректно. Opera mini не имеет поддержки CSS Grid [2].

| IE | Edge | Firefox | Chrome | Safari | Opera | IOS Safari | Opera Mini | Android Browser |
|-----|-------|---------|--------|-----------|-------|------------|------------|-----------------|
| | | 2-39 | 4-28 | | | | | |
| | | 40-51 | 29-56 | | 10-27 | | | |
| 6-9 | 12-15 | 52-53 | 57 | 3.1-10 | 28-43 | 3.2-10.2 | | |
| 10 | 16-17 | 54-65 | 58-72 | 10.1-11.1 | 44-57 | 10.3-11.4 | | 2.1-4.4.4 |
| 11 | 18 | 66 | 73 | 12 | 58 | 12.1 | all | 67 |
| | | 67-68 | 74-76 | 12.1-TP | | 12.2 | | |

| Blackberry Browser | Opera Mobile | Chrome for Android | Firefox for Android | IE Mobile | UC Browser for Android | Samsung Internet | QQ Browser | Baidu Browser |
|--------------------|--------------|--------------------|---------------------|-----------|------------------------|------------------|------------|---------------|
| | | | | | | | | |
| | | | | | | 4-5.4 | | |
| 7 | 12-12.1 | | | 10 | | 6.2-7.4 | | |
| 10 | 46 | 71 | 64 | 11 | 11.8 | 8.2 | 1.2 | 7.12 |
| | | | | | | | | |

| | | |
|------------------|---------------------|-------------------|
| Полная поддержка | Частичная поддержка | Не поддерживается |
|------------------|---------------------|-------------------|

Рисунок 14

В заключение следует отметить, что плюсы и минусы CSS Grid – для каждого свои. Но в общем случае применение CSS Grid позволяет упростить разметку и увеличить гибкость при разработке макета в двух измерениях, а также значительно упрощает адаптацию к различным устройствам.

Список литературы:

1. Полное руководство по CSS Grid [Электронный ресурс]. – <https://tuhub.ru/posts/css-grid-complete-guide>
2. Wikipedia – The Free Encyclopedia [Электронный ресурс]. – https://en.wikipedia.org/wiki/CSS_grid_layout
3. Обучение программированию [Электронный ресурс]. – <https://ru.hexlet.io/blog/posts/css-grid-is-better>