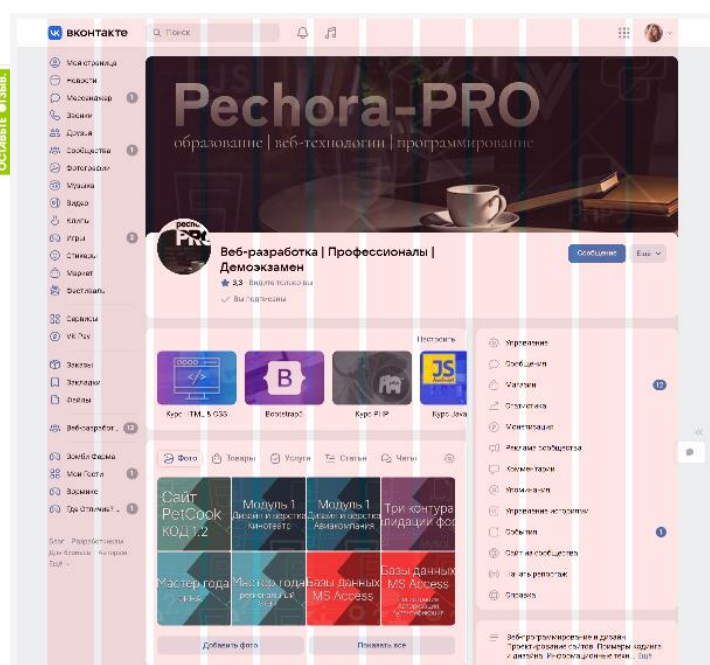
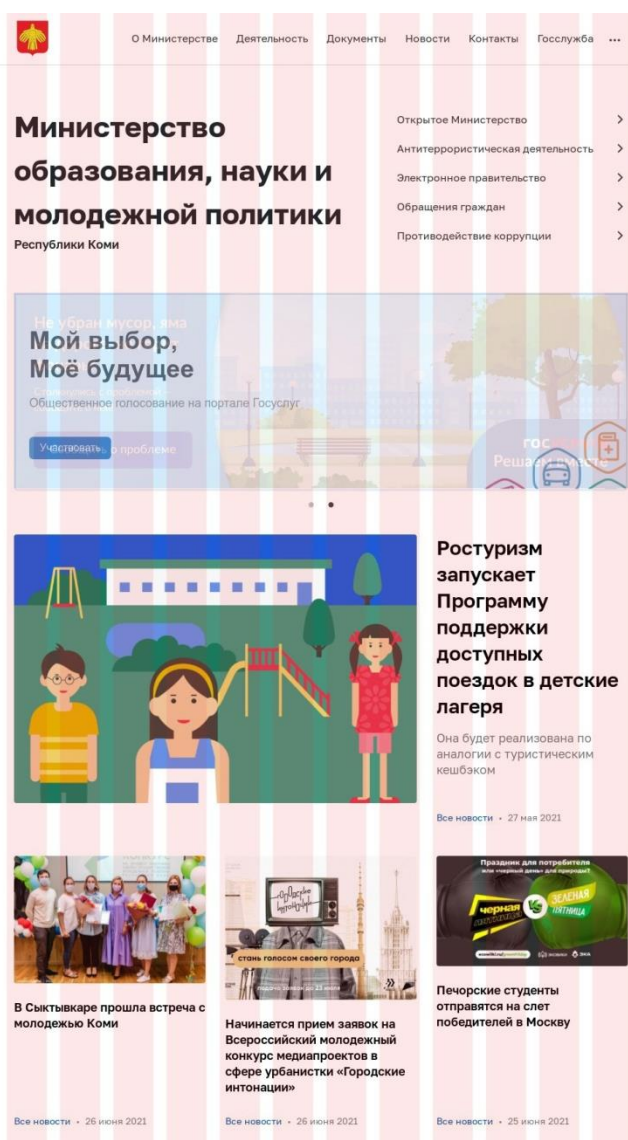


ЗАДАНИЕ 29 ГИБКАЯ СЕТКА

Очень многое в веб-дизайне пришло к нам из дизайна книжного и сетка не исключение.

Если мы внимательно посмотрим на практически любой современный сайт, то сможем отметить некоторую упорядоченность расположения элементов... а если не менее внимательно посмотрим и во второй, то увидим, что элементы на странице выстраиваются в строчки и столбцы.



Эти столбцы и есть основа нашей сетки. Количество столбцов и размеры этой сетки могут быть разными.

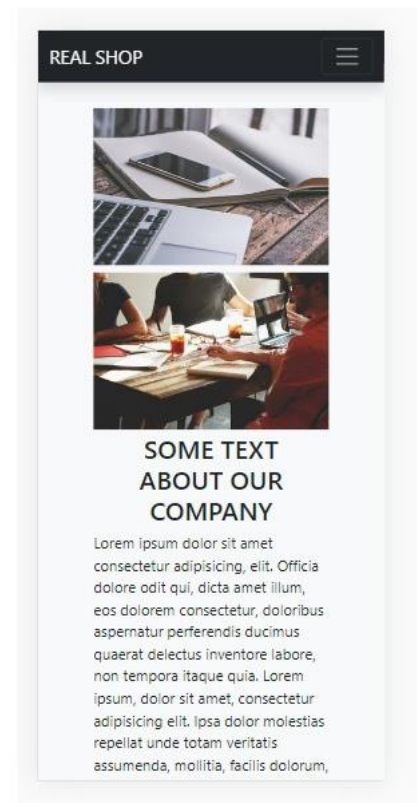
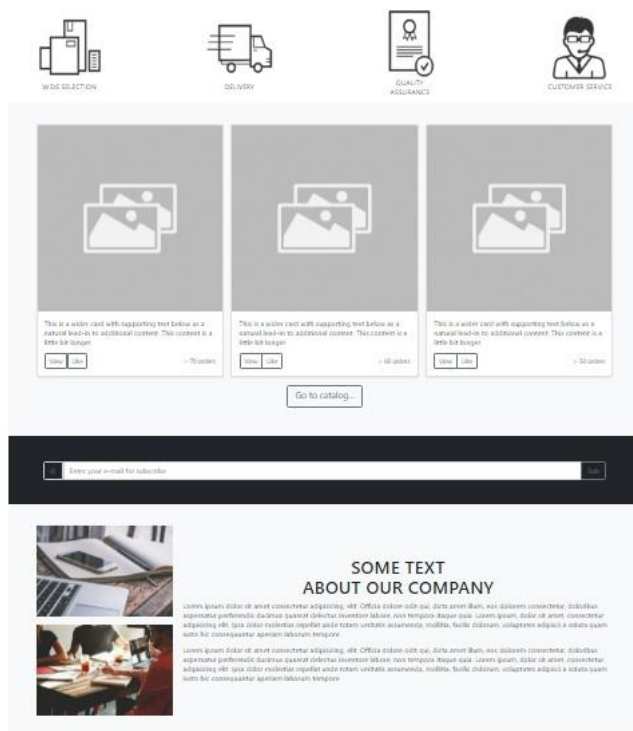
Раньше стандартом макета, выдаваемого дизайнером верстальщику был размер 960 px, потом популярен стал размер 1140 px. Если же говорить о количестве столбцов в макете, то здесь многое зависит от его сложности, от количества элементов в нем. Это могут быть и достаточно стандартные 12 (тоже пришедшие из книгопечатанья), так и 9, 16 или даже 24 столбца.

С появлением адаптивной верстки размеры этих сеток, а точнее столбцов в них перестали быть фиксированными. Задачу верстальщика это несомненно усложнило, но оно того стоило.

Итак, в адаптивной или, как еще говорят, гибкой сетке с изменением размера экрана будет изменяться и размер составляющего ее столбца. Для этого ширину столбца будем задавать не в строгих пикселях, а более гибких процентах.

Мало того, содержимое в этих столбцах может перегруппироваться... Например, сначала на экране компьютера блок занимал 3 столбца 12-ти колоночной сетки, потом 4 или 6 на планшете, а на мобильном устройстве все 12.

В качестве примера, посмотрите, как изменяется структура нашего макета Магазина при переходе от десктопной версии к мобильной.



Надо понимать, что существует большое количество готовых решений для построения сеток, например, <https://960.gs/> 960 Grid System или <https://1440px.com/> 1440 Grid Template или фреймворков, использующих их же – наш любимый Bootstrap или менее популярный Foundation.

Но мы с вами стремимся к пониманию, а не бездумному копированию. Поэтому на этапе обучения создадим стек самостоятельно.

Алгоритм работы

1. В файле **index.html** создадим структуру, на которой будем тестировать возможности сетки. Вся она будет находиться внутри контейнера и состоять из колонок.

```
<div class="container">

  <div class="cols">Колонка</div>
  <div class="cols">Колонка</div>
  <div class="cols">Колонка</div>

  <div class="cols">Колонка</div>
  <div class="cols">Колонка</div>
```

```

    <div class="cols">Колонка</div>
    <div class="cols">Колонка</div>

    <div class="cols">Колонка</div>

</div>

```

- В файле **style.css** определим общие настройки документа, настройки контейнера и колонок.

```

*{
  margin:0;
  padding:0;
  box-sizing: border-box;
}

.container {
  background: #d2d2d2;
  width: 90%;
  margin: 0 auto;
}

.container .cols {
  float: left;
  margin: 0 0 1em;
  padding: 0 1em;
  background: #fd4;
  text-align: center;
}

```

Итак, наша сетка будет состоять из 12 столбцов, а ширина родительского контейнера – 100%.

- Вычислим размеры одного столбца $100\%/12 = 8.333333333\%$
Аналогичным образом вычислим значения для блоков шириной 2, 3 и т.д. столбца и запишем их в файл со стилями.

```

.container .cols.col-1 { width: 8.33333333%;}
.container .cols.col-2 { width: 16.66666666%;}
.container .cols.col-3 { width: 25%;}
.container .cols.col-4 { width: 33.33333333%;}
.container .cols.col-5 { width: 41.66666666%;}
.container .cols.col-6 { width: 50%;}
.container .cols.col-7 { width: 58.33333333%;}
.container .cols.col-8 { width: 66.66666667%;}
.container .cols.col-9 { width: 75%;}
.container .cols.col-10 { width: 83.33333333%;}
.container .cols.col-11 { width: 91.66666667%;}
.container .cols.col-12 { width: 100%;}

```

4. Допишем необходимые классы в разметку и проверим работает ли наш код.

```
<div class="container">

  <div class="cols col-4">Колонка</div>
  <div class="cols col-4">Колонка</div>
  <div class="cols col-4">Колонка</div>

  <div class="cols col-3">Колонка</div>
  <div class="cols col-3">Колонка</div>
  <div class="cols col-3">Колонка</div>
  <div class="cols col-3">Колонка</div>

  <div class="cols col-12">Колонка</div>

</div>
```

5. Добавим в одну из наших колонок большое количество текста. Проверим работоспособность верстки в этом случае.

```
<div class="cols col-4">Колонка шириной 33.33333333%</div>

<div class="cols col-4">Колонка шириной 33.33333333% НО содержит большее
количество текста, которое ломает всю нашу структуру без внесения
дополнительных оберток и стилей</div>

<div class="cols col-4">Колонка шириной 33.33333333%</div>
```

Как видим сетка работает, но не очень корректно, поэтому следующими шагами исправим ситуацию.

6. Добавляем дополнительную «обертку» класс `<div class="row">` для каждой «строки».

```
<div class="container">

  <div class="row">
    <div class="cols col-4">...</div>
    <div class="cols col-4">...</div>
    <div class="cols col-4">...</div>
  </div>

  <div class="row">
    <div class="cols col-3">...</div>
    <div class="cols col-3">...</div>
    <div class="cols col-3">...</div>
    <div class="cols col-3">...</div>
  </div>

</div>
```

```
<div class="row">
  <div class="cols col-12">...</div>
</div>

</div>
```

7. Для того чтобы гибкая сетка эффективно перестраивалась при наличии большого количества текста добавим в файл css настройки обтекания для контейнеров.

```
.container:before,
.container:after,
.row:before,
.row:after,
.clear:before,
.clear:after
{
  content: " ";
  display: table;
}

.container:after,
.row:before,
.row:after,
.clear:after
{
  clear: both;
}
```

Теперь наша сетка и выглядит, и ведет себя достойно даже при большом количестве текста.