

Шаг 4 – Пример применения

Исходники – папка step4

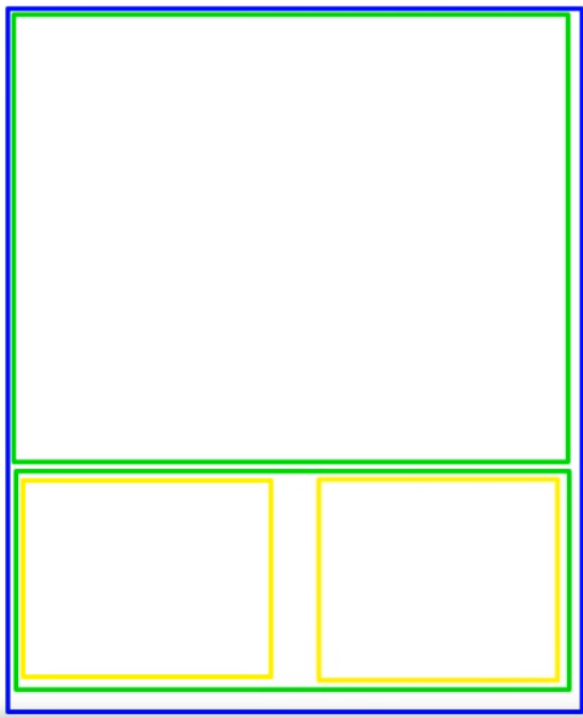
Задача: сверстать адаптивную страницу под разные разрешения экрана (широкоформатный, планшетный и мобильный экран) по макетам.

МАКЕТЫ

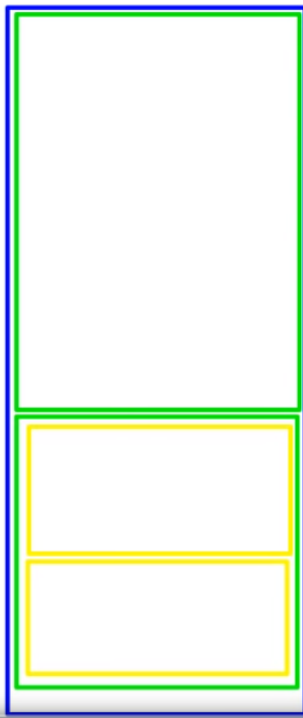
Широкоформатный экран:



Планшет:



Смартфон:



1. Широкоформатный экран

Разметка (структура документа):

```
<main>
  <div class="content">

  </div>
  <aside>
    <div class="side-block">

    </div>
    <div class="side-block">

    </div>
  </aside>
</main>
```

Наполним класс контент текстовой информацией. Для этого необходимо перейти в папку шаг 1 и скопировать содержимое контейнера с классом content виз файла index.html:

```
<div class="content">
  <h1>Lorem ipsum dolor sit amet</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipis
  <p>Sed pharetra bibendum turpis, ac posuere urna
  <p>Donec a diam ut sem hendrerit vestibulum. Viva
  <p>Lorem ipsum dolor sit amet, consectetur adipis
  <p>Sed pharetra bibendum turpis, ac posuere urna
  <p>Donec a diam ut sem hendrerit vestibulum. Viva
</div>
```

Далее заполним блоки боковой колонки aside. Добавляем в них по 4 коротких абзаца:

```
<aside>
  <div class="side-block">
    <p>Donec a diam ut sem hendrerit vestibulum</p>
    <p>Donec a diam ut sem hendrerit vestibulum</p>
    <p>Donec a diam ut sem hendrerit vestibulum</p>
    <p>Donec a diam ut sem hendrerit vestibulum</p>
  </div>
  <div class="side-block">
    <p>Donec a diam ut sem hendrerit vestibulum</p>
    <p>Donec a diam ut sem hendrerit vestibulum</p>
    <p>Donec a diam ut sem hendrerit vestibulum</p>
    <p>Donec a diam ut sem hendrerit vestibulum</p>
  </div>
</aside>
```

Зададим основные стили:

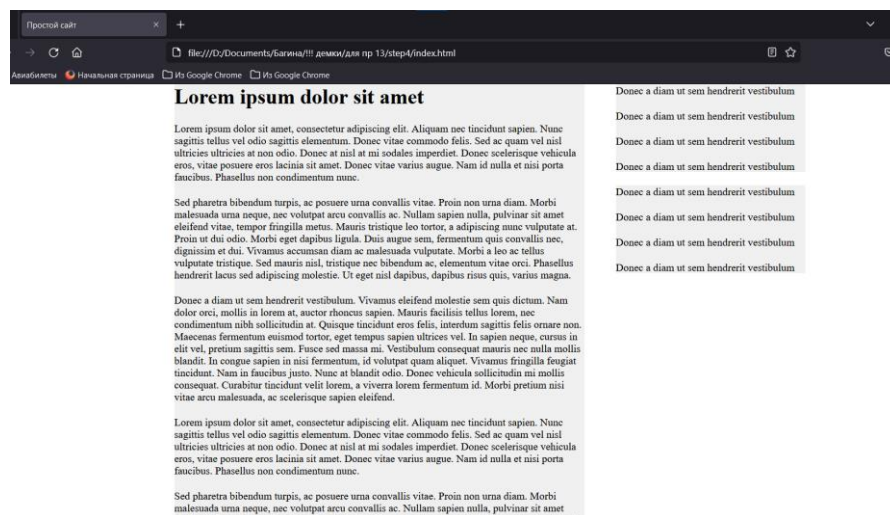
```
# style.css > ...
1  /*обнуляем отступы у всех элементов:*/
2  * {
3    padding: 0;
4    margin: 0;
5  }
6  h1, p {
7    margin-bottom: 20px;
8  }
9  main {
10   max-width: 960px;
11   margin: 0 auto;
12 }
```

Разбиваем текстовую часть в соответствии с макетом на две колонки:

- Задаем родительскому блоку свойство `display` со значением `flex`, также указываем отступ между блоками.
- Для блоков устанавливаем ширину
- Для текстовых блоков задаем фоновый цвет

```
2  * {
3      padding: 0;
4      margin: 0;
5  }
6  h1, p {
7      margin-bottom: 20px;
8  }
9  main {
10     max-width: 960px;
11     margin: 0 auto;
12     display: flex;
13     gap: 3%;
14 }
15 .content {
16     width: 60%;
17     background: #f0f0f0;
18 }
19 aside {
20     width: 30%;
21 }
22 .side-block {
23     background: #f0f0f0;
24 }
```

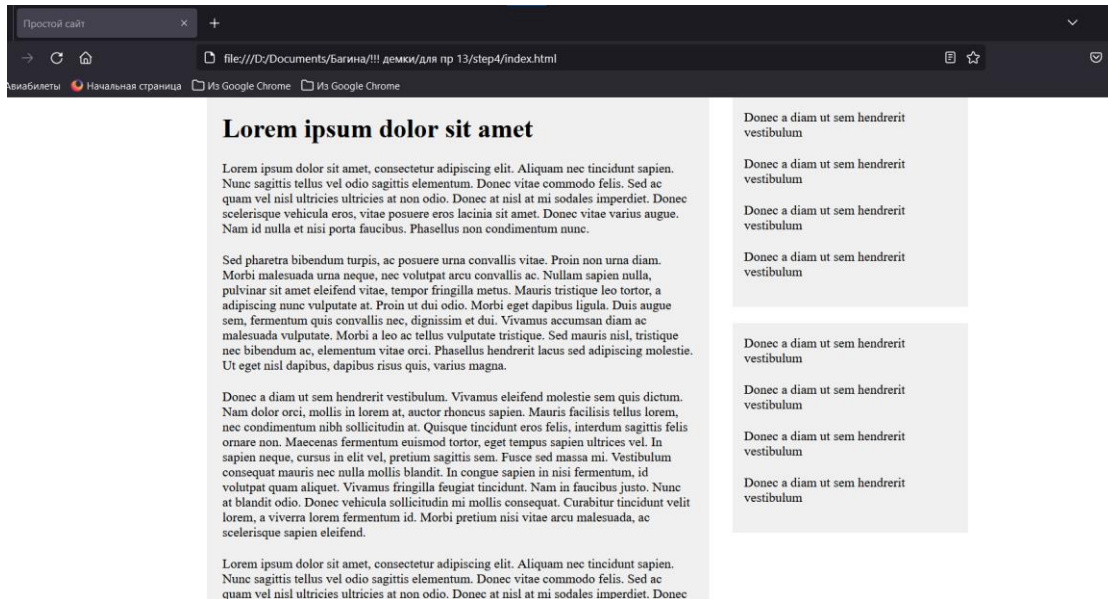
Итог:



Добавим **внутренние отступы** в контейнерах, чтобы текст не прилипал к границам блоков:

- Для класса `content` 2% со всех сторон
- Для класса `side-block` 5% со всех сторон, также для данного класса необходимо установить **внешний отступ** снизу в 20 пикселей

Итог:



2. Планшетный макет

Добавляем свойство flex-wrap со значением wrap.

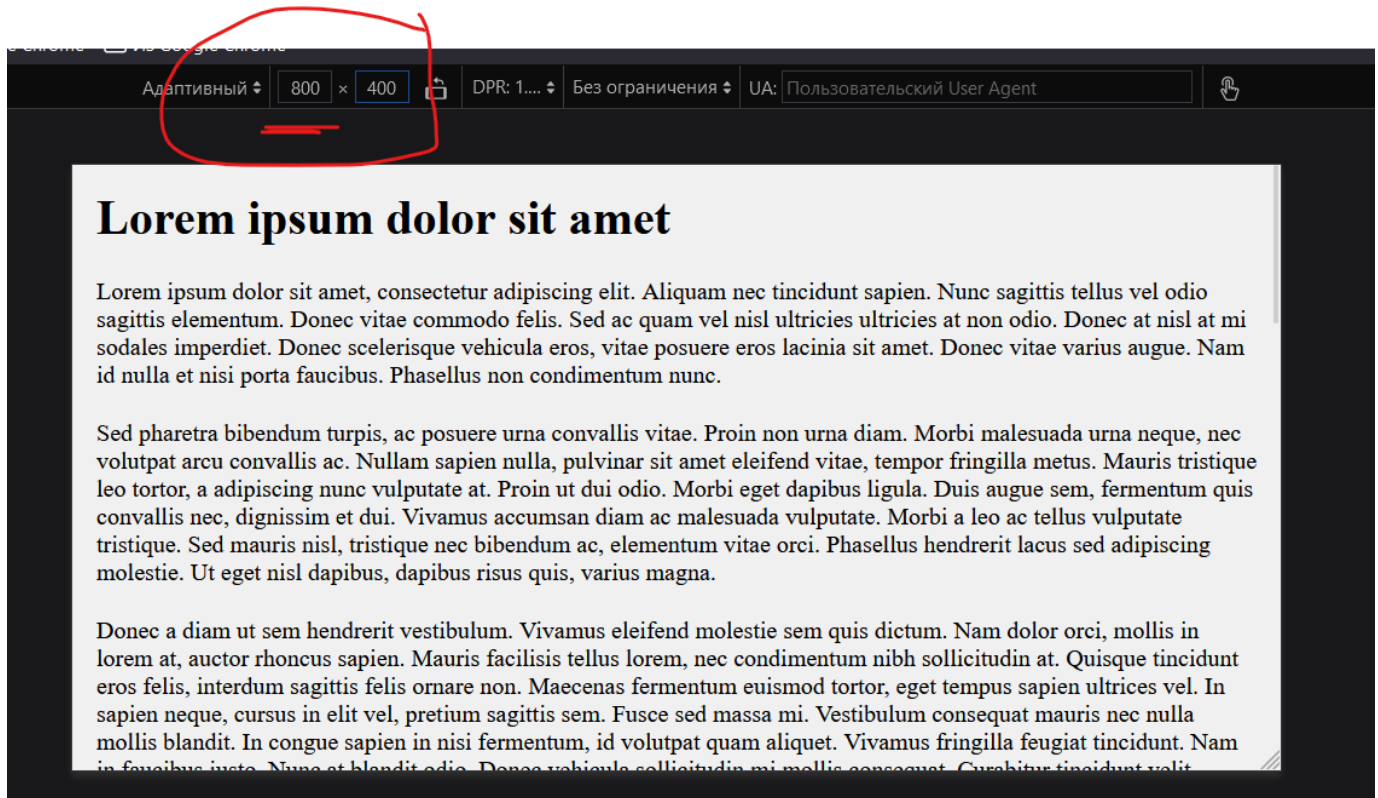
```
9   main {
10     max-width: 960px;
11     margin: 0 auto;
12     display: flex;
13     ✓ flex-wrap: wrap;
14     gap: 3%;
15 }
```

В конце кода CSS прописываем медиазапрос для планшетной версии сайта. Внутри медиазапроса мы изменяем ширину блока content до максимальной (100%). После этого запроса будет срабатывать свойство flex-wrap, которое мы написали ранее.

```
30  @media screen and (max-width: 800px) {
31    .content {
32      width: 100%;
33    }
34 }
```

Проверяем в режиме адаптивного дизайна в браузере Mozilla Firefox (другие браузеры некорректно работают с медиазапросами!)

Укажите ширину в 800 пикселей и нажмите Enter



Затем также проверьте ширину в 801 пиксель. Все элементы должны вернуться на свои места.

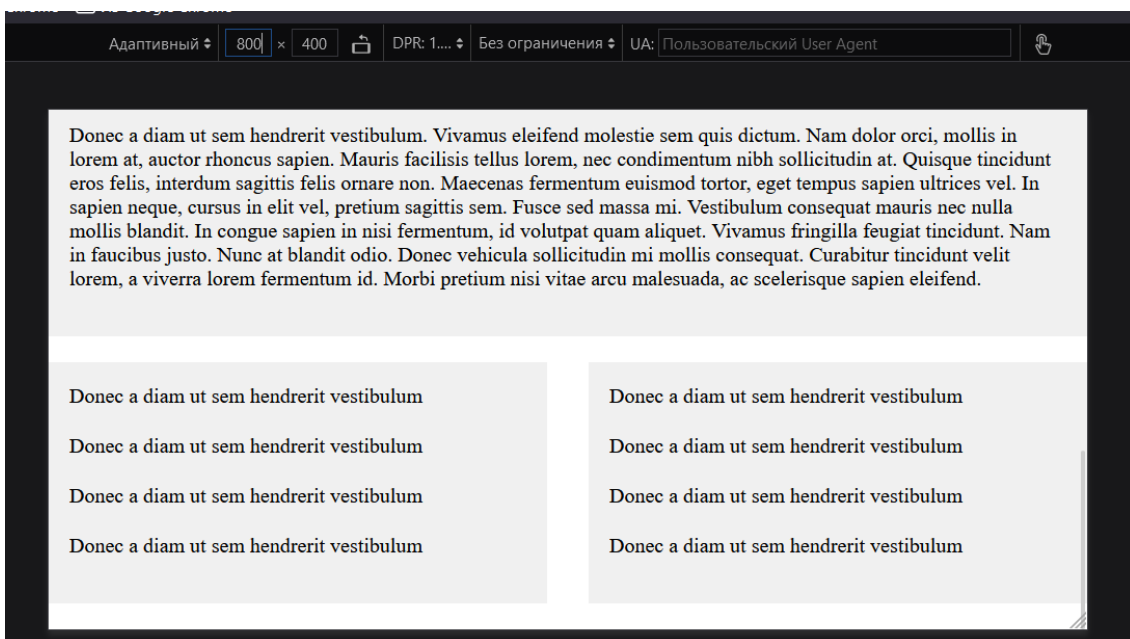


Теперь настроим боковую колонку внутри медиазапроса:

- Для блока aside
 - указываем ширину в 100%
 - добавляем внешний отступ сверху в 20 пикселей
 - реализуем колонки при помощи display: flex.
 - Выравниваем колонки при помощи свойства justify-content со значением space-between. Это позволит приклеить колонки по краям нашего блока aside.
 - Также сразу добавляем свойство для дальнейшего переноса колонок и для и
- Для каждой колонки указываем ширину и меняем внутренний отступ

```
30 @media screen and (max-width: 800px) {
31   .content {
32     width: 100%;
33   }
34   aside {
35     width: 100%;
36     margin-top: 20px;
37     display: flex;
38     justify-content: space-between;
39     flex-wrap: wrap;
40   }
41   .side-block {
42     width: 44%;
43     padding: 2%;
44   }
45 }
```

Результат:

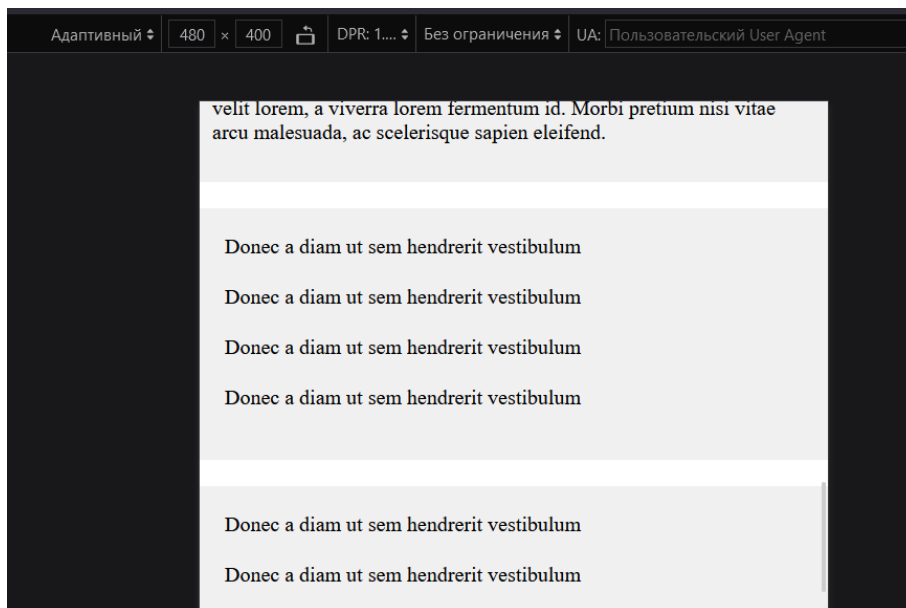


3. Мобильный макет

Для мобильного макета потребуется написать новый медиазапрос и в нём всего два свойства к блоку side-block

```
46 @media screen and (max-width: 480px) {  
47     .side-block {  
48         width: 96%;  
49         padding: 4%;  
50     }  
51 }
```

Проверяем



Проверяем при другой ширине

