
ЗАДАНИЕ 10 РЕДАКТИРОВАНИЕ СТРАНИЦЫ.

В папке **sample** представлена страница **Галерея** – коллекция миниатюр фото, по клику на которые открываются оригиналы.

Сейчас все распределение элементов на странице построено на табличной верстке и тегах `<center>`. Наша задача сделать верстку более современной и использовать для позиционирования CSS Flexbox.

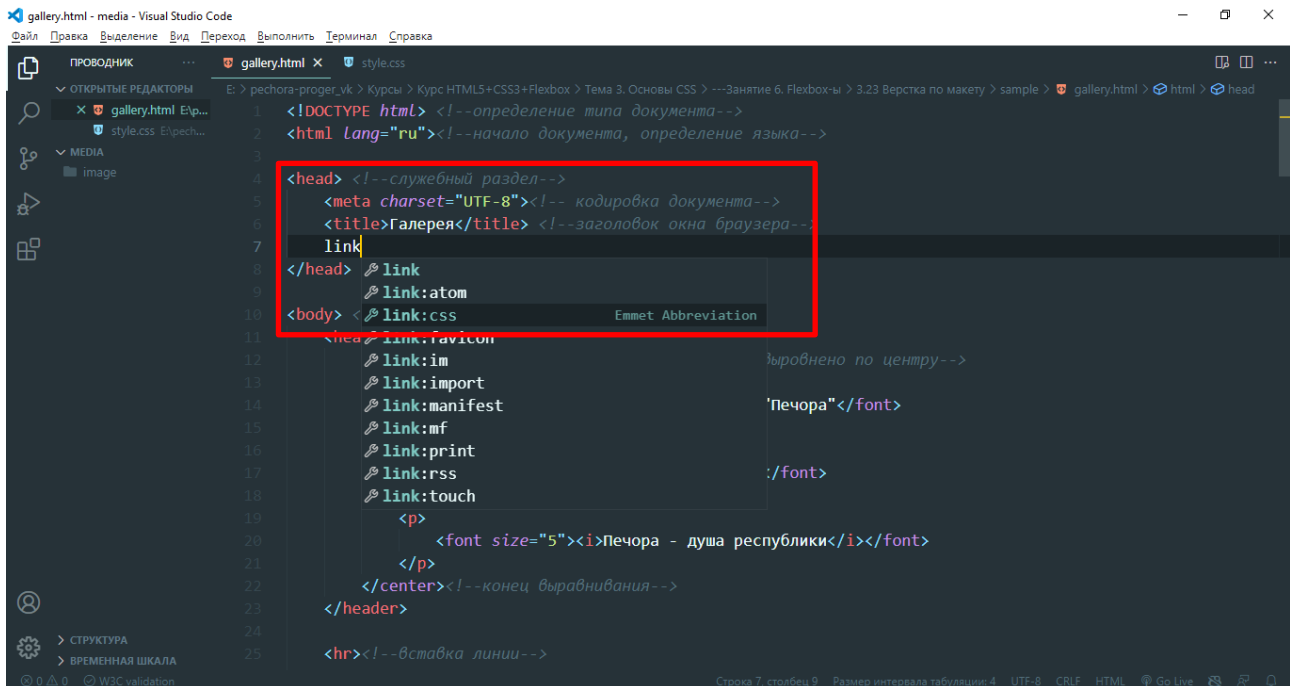
Вы можете создать новую страницу и копировать в нее необходимый контент или редактировать текущую.

Начнем с анализа структуры страницы. Она состоит из 4-х частей:

- Шапка сайта с текстом по центру блока
- Панель навигации
- Контентная часть, состоящая из трех однотипных блоков
 - Заголовков
 - 8 фото с подписями
- Подвал сайта с текстом по центру блока

1. Создаем файл **style.css** и подключаем его к нашей странице.

Можете использовать сокращение Emmet – `link:css`



- Отредактируем верстку, убрав тег **center** в шапке, подвале сайта и панели навигации. Его задачи теперь будут реализованы с помощью свойств justify-content и align-items.
- Запишем css правила для элемента **header**. Сделаем блок гибким и распределим в нем контент.

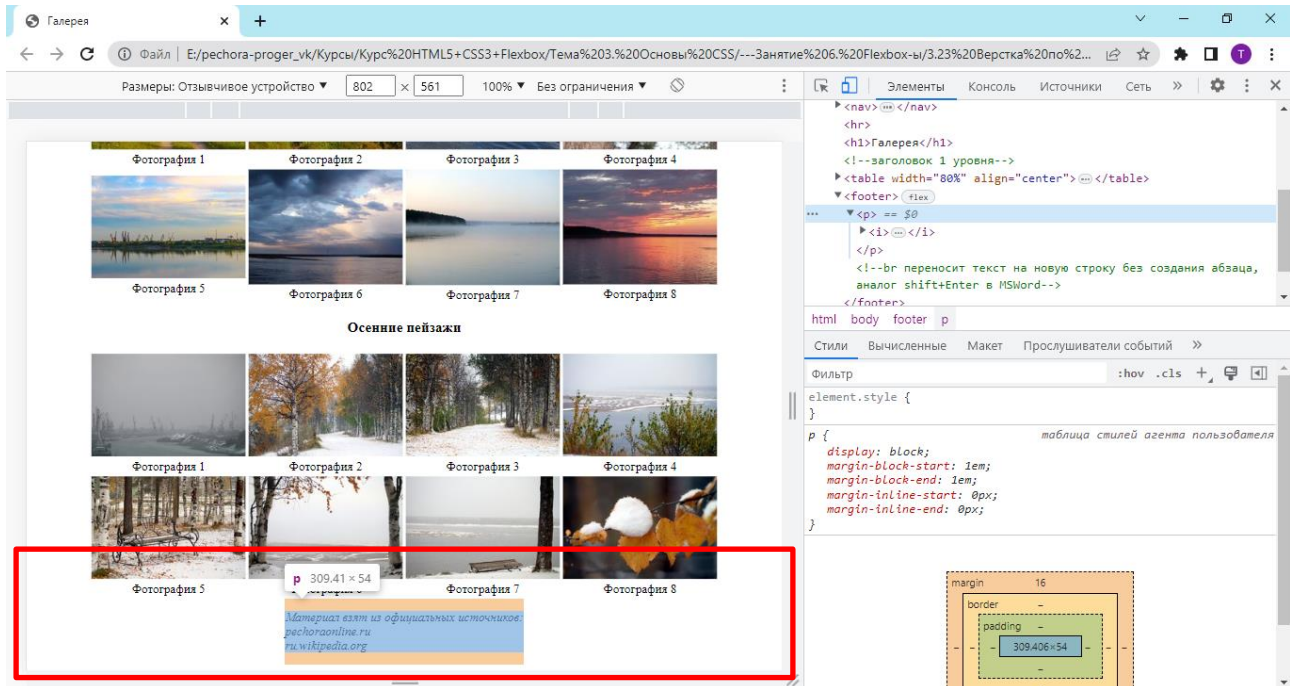
```

header {
  display: flex; /* делаем блоки гибкими */
  flex-direction: column; /* разворачиваем главную ось */
  justify-content: center; /* размещаем контент по центру по
главной оси */
  align-items: center; /* размещаем контент по центру по
второстепенной оси */
}

```

Переходим к подвалу. Внешне кажется структура блока похожа на шапку, но попробуйте добавить селектор **footer** к уже написанным стилям.

Результат нас не устраивает, обратите внимание, как теперь выровнен текст. Это последствия использования тега **
**.

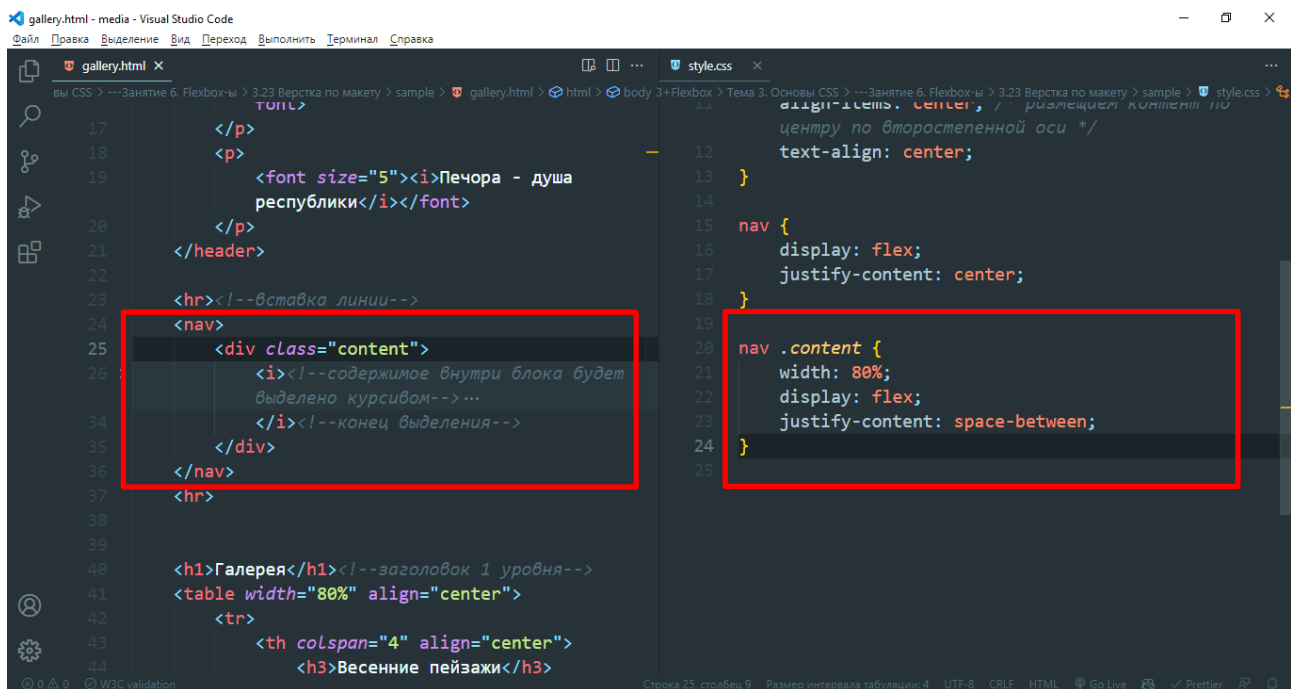


4. Чтобы исправить это, нам придется забежать вперед и добавить к подвалу еще одно стилевое правило: `text-align: center;` Оно обратится непосредственно к текстовому содержимому и выровняет его по центру.
5. Следующий шаг – меню.

```
nav {
  display: flex;
  justify-content: center;
}
```

Но давайте немного поэкспериментируем.

6. Добавим в меню дополнительную гибкую обертку шириной 80 % (70%, 60% на ваш выбор) и изменим для нее распределение элементов на `space-between` или `space-around` или `space-between`.



```
gallery.html
17 </p>
18 <p>
19 <font size="5"><i>Печора - душа
20 республики</i></font>
21 </p>
22 </header>
23 <hr><!--вставка линии-->
24 <nav>
25 <div class="content">
26 <i><!--содержимое внутри блока будет
34 выделено курсивом--> ...
35 </i><!--конец выделения-->
36 </div>
37 </nav>
38 <hr>
39
40 <h1>Галерея</h1><!--заголовок 1 уровня-->
41 <table width="80%" align="center">
42 <tr>
43 <th colspan="4" align="center">
44 <h3>Весенние пейзажи</h3>
45 </th>
46 </tr>
47 </table>
48 </body>
49 </html>
```

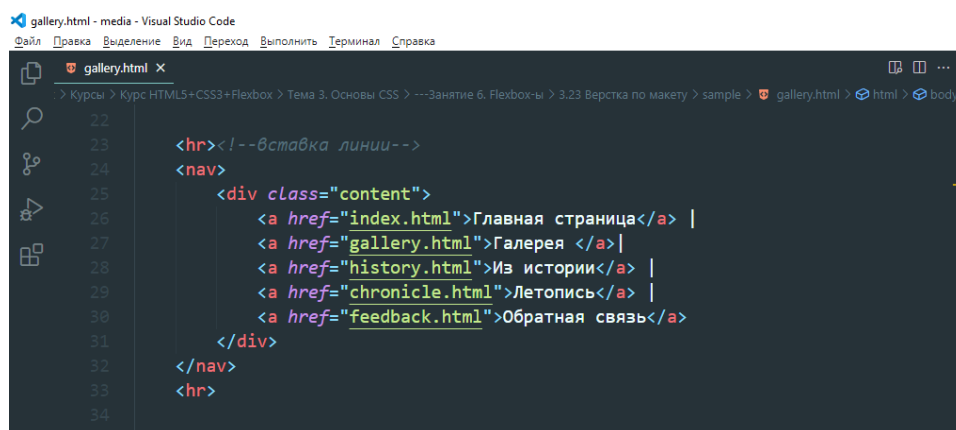
```
style.css
11 align-items: center; /* выравнивание контента по
12 центру по второстепенной оси */
13 text-align: center;
14 }
15 nav {
16 display: flex;
17 justify-content: center;
18 }
19
20 nav .content {
21 width: 80%;
22 display: flex;
23 justify-content: space-between;
24 }
```

Посмотрите на страницу в браузере. Как считаете, почему так произошло? Ведь свойства выставлены правильно...

Подсказка: DOM-дерево

Исправим расположение, отредактировав верстку.

- Откажемся от курсивного оформления и настройки шрифта. Ничего страшного в этом нет, такие настройки выносятся в стили. Работе с текстом будет посвящен отдельный урок.



```
gallery.html
22 <hr><!--вставка линии-->
23 <nav>
24 <div class="content">
25 <a href="index.html">Главная страница</a> |
26 <a href="gallery.html">Галерея </a> |
27 <a href="history.html">Из истории</a> |
28 <a href="chronicle.html">Летопись</a> |
29 <a href="feedback.html">Обратная связь</a>
30 </div>
31 </nav>
32 <hr>
33
34
```

Переходим к самому объемному блоку.

- Проанализируем нашу таблицу и выстроим для нее замену из блоков. Продублируем содержимое контейнера по количеству разделов фото

```

<h1>Галерея</h1><!--заголовок 1 уровня-->

<div class="container"> <!-- таблица-->
  <div class="caption"></div> <!-- первая строка-->
  <div class="photos"> <!-- 2-3 строки-->
    <div class="photo"></div> <!--ячейки-->
    <div class="photo"></div>
    <div class="photo"></div>
    <div class="photo"></div>
    <div class="photo"></div>
    <div class="photo"></div>
    <div class="photo"></div>
  </div>
</div>

```

9. Скопируйте контент из таблицы в полученную структуру.

gallery.html - media - Visual Studio Code

Файл Правка Выделение Вид Переход Выполнить Терминал Справка

```

35
36 <h1>Галерея</h1><!--заголовок 1 уровня-->
37
38 <div class="container"> <!-- таблица-->
39   <div class="caption"> <!-- первая строка-->
40     <h3>Весенние пейзажи</h3>
41   </div>
42   <div class="photos"> <!-- 2-3 строки-->
43     <div class="photo">
44       <a href="media/image/gallery_image/pechora_springtime/1.jpg" target="_blank">
45         
46       </a>
47       <br>
48       Фотография 1
49     </div> <!--ячейки-->
50     <div class="photo">...</div>
51     <div class="photo">...</div>
52     <div class="photo">...</div>
53     <div class="photo">...</div>
54     <div class="photo">...</div>
55     <div class="photo">...</div>
56     <div class="photo">...</div>
57   </div>
58 </div>
59

```

Итак, стили для наших блоков записываем поочередно.

Помним, что гибкость блоков не наследуется.

10. Настроим отображение контейнера.

```
.container {  
  width: 80%;  
  margin: 0 auto;  
  display: flex;  
  flex-direction: column;  
}
```

11. Следующий шаг – обертка для всех фото. Здесь важно выровнять элементы по ширине, разрешить им переходить на следующую строку.

```
.photos {  
  display: flex;  
  flex-wrap: wrap;  
  justify-content: space-between;  
}
```

12. Теперь для каждого блока с фото зададим ширину, отступы и распределение элементов.

```
.photo {  
  width: 24%;  
  margin-bottom: 2%;  
  display: flex;  
  flex-wrap: wrap;  
  justify-content: center;  
  align-items: end;  
}
```

На этом работа над страницей завершена.