

Лекция 4. Использование мультимедиа

Оглавление

Лекция 4. Использование мультимедиа	1
4.1. Подключение изображений	1
4.2. Форматы изображений для Web.....	2
4.3. Figure и figcaption	3
4.4. Изображение-ссылка	4
4.5. Изображение в ячейке таблицы.....	4
4.6. Видео	5
4.7. Аудио	7
Задание к лабораторной работе 4:.....	7

4.1. Подключение изображений

Для вывода изображений в HTML используется элемент `img`. Этот элемент представляет нам два важных атрибута:

- `src`: путь к изображению. Это может быть относительный или абсолютный путь в файловой системе или адрес в интернете
- `alt`: текстовое описание изображения. Если браузер по каким-то причинам не может отобразить изображение (например, если у атрибута `src` некорректно задан путь), то браузер показывает вместо самой картинки данное текстовое описание.

Атрибут `alt` важен также тем, что поисковые системы по текстовому описанию могут индексировать изображение.

Например, поместим в ту же папку, где у нас лежит файл `index.html`, файл изображения `image.png`. И затем отобразим его на web-странице:

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Тег img в HTML5</title>
  </head>
  <body>
    
  </body>
</html>
```

Здесь файл изображения называется `image.png`, и он находится в одной папке с web-страницей `index.html`. При этом надо учитывать, что `img` является "пустым" элементом, то есть не содержит закрывающегося тега.

Полезными являются атрибуты `width` и `height`. С их помощью задается размер рисунка:

```

```

Размеры в этих атрибутах задаются без единиц измерения.

Если задать только один из размеров, ширину или высоту, то вторую размерность браузер вычислит самостоятельно исходя из пропорций изображения.

Заданный размер не влияет на скорость загрузки. Следует указывать реальные ширину и высоту рисунка, чтобы не "рвать" макет при загрузке. Нельзя также "масштабировать" изображение указанием неверных ширины и высоты. Как правило, изображения для сайта помещают во вложенную папку сайта с названием, например, `img`. В ряде браузеров это способствует и лучшему кэшированию рисунков.

Используя стилевые особенности, в частности, отступы и обтекание, можно комбинировать изображения с текстом. Например:

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Ter img в HTML5</title>
  </head>
  <body>
    <div>
      
      <h1>Lorem Ipsum</h1>
      <b>Lorem Ipsum</b> is simply dummy text of the printing
      and typesetting industry. Lorem Ipsum has been the
      industry....
    </div>
  </body>
</html>
```

4.2. Форматы изображений для Web

Существует несколько основных форматов изображений для Web: JPEG, PNG, SVG и GIF.

Формат SVG переводится как "масштабируемая векторная графика". Качество таких изображений не меняется при изменении размеров, да и размер у них небольшой. Отлично подходит для малоцветных схем, логотипов и иконок. Чаще всего используется в случаях, когда необходимо масштабировать изображение без потерь, изменять цвет элементов изображения, анимировать части изображения.

Формат JPEG был разработан для сжатия и хранения полноцветных фотографий. Он поддерживает более 16 миллионов цветов. Но нужно помнить, что JPEG сжимает изображения с потерей качества. Поэтому если мы хотим уменьшить "вес" изображения, то придётся ухудшать его внешний вид. Главная задача при работе

с JPEG — подобрать такой уровень сжатия, чтобы и вес, и качество картинки были приемлемыми.

Таким образом, формат JPEG лучше подходит для:

- полноцветных изображений, фотографий;
- изображений с плавным переходом яркости и контраста;
- рисунков с большим количеством разноцветных деталей.

В отличие от JPEG, **формат PNG** является форматом сжатия без потерь и позволяет сохранять изображения, в которых требуется особенная чёткость. Например, скриншоты сайтов, чертежи и печатный текст. Главная особенность формата PNG — поддержка прозрачности, то есть каждому пикселю в отдельности можно задать свою степень прозрачности.

Итак, формат PNG подходит для:

- изображений с прозрачностью и полупрозрачностью;
- когда необходима повышенная точность полноцветных изображений;
- изображений с резкими переходами цветов.

PNG — это относительно новый формат, который был введён как альтернатива для формата GIF.

Формат GIF был разработан для передачи растровых изображений по интернету. GIF имеет цветовую палитру, состоящую из 256 цветов. Алгоритм GIF выбирает 256 наиболее используемых в исходном изображении цветов, а все остальные оттенки создаются путём подмешивания — подбора соседних пикселей таким образом, чтобы человеческий глаз воспринимал их как нужный цвет. По этой причине GIF не подходит для хранения полноцветных изображений и фотографий, но подходит для простейших анимаций.

Формат поддерживает прозрачность — каждый пиксель изображения может быть в двух состояниях: прозрачный или непрозрачный, полупрозрачность не поддерживается.

В последнее время GIF-изображения становятся всё менее используемыми и заменяются на другие, более оптимальные форматы.

Таким образом, формат GIF подходит если:

- нужна простейшая анимация.

4.3. Figure и figcaption

Тег `<figure>` подходит для любого иллюстративного или демонстрационного материала, которым можно дополнить содержание документа: схемы, графики, примеры кода, таблицы и так далее. При удалении такого материала основное содержание не должно пострадать, иначе это не дополнительный материал и тег `<figure>` не подходит для его разметки.

Для разъясняющего комментария к такому иллюстративному материалу существует тег `<figcaption>`, который размещается первым или последним элементом внутри `<figure>`. Например:

```
<figure>
  <p></p>
  <figcaption>Подпись к содержимому</figcaption>
</figure>
```

4.4. Изображение-ссылка

Ссылки можно делать не только с помощью текста, но и с помощью изображений. Для этого нужно обернуть тег `` в тег `<a>`. Например:

```
<a href="http://keksby.ru">
  
</a>
```

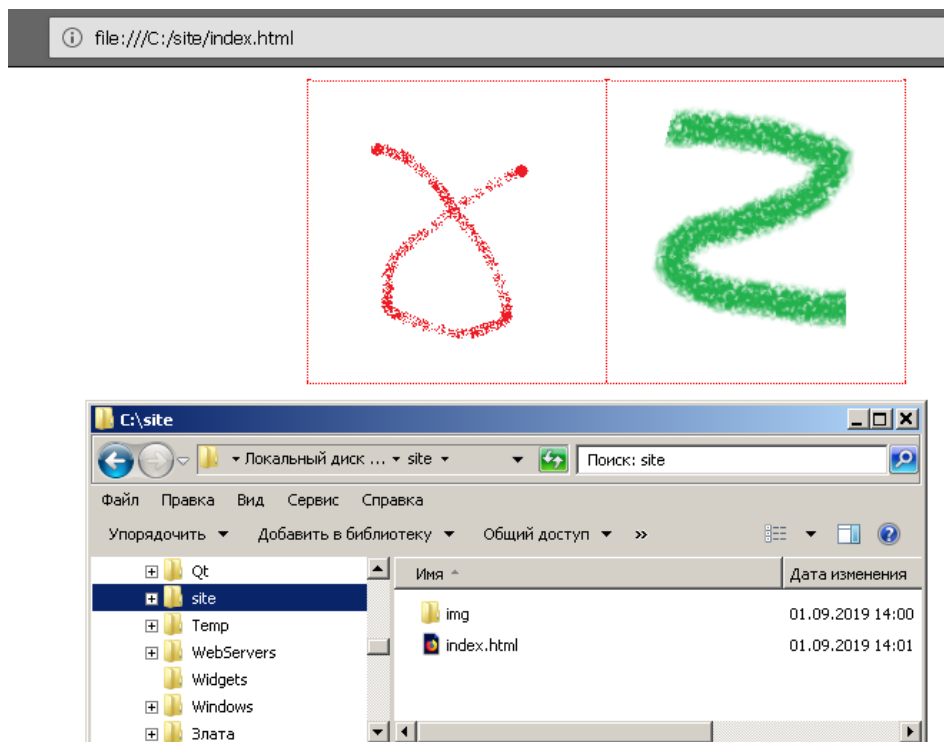
4.5. Изображение в ячейке таблицы

Нередки случаи, когда изображения размещаются в табличных шаблонах, например:

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <title>Таблица с рисунками</title>
  <style>
    .mytab {
      margin: auto;
      padding: 4px;
      border: 1px dotted red;
      border-collapse: collapse;
    }
    .mytd {
      text-align: center;
      vertical-align: top;
      width: 50%;
    }
  </style>
</head>
<body>

<table class="mytab">
  <tr>
    <td class="mytab mytd">
      
    </td>
    <td class="mytab mytd">
      
    </td>
  </tr>
</table>
</body></html>
```

Вот результат выполнения этого кода в браузере, показана также открытая папка с файлами примера.



Здесь в ячейки таблиц вставлено 2 картинки с указанием их пиксельных размеров и альтернативного текста. Рисунки загружаются из вложенной в папку сайта папки `img`.

4.6. Видео

Для воспроизведения видео в HTML5 используется элемент `video`. Чтобы настроить данный элемент, мы можем использовать следующие его атрибуты:

- `src`: источник видео, это может быть какой-либо видеофайл;
- `width`: ширина элемента;
- `height`: высота элемента;
- `controls`: добавляет элементы управления воспроизведением;
- `autoplay`: устанавливает автовоспроизведение;
- `loop`: задает повторение видео;
- `muted`: отключает звук по умолчанию

Хотя мы можем установить ширину и высоту, но они не окажут никакого влияния на aspectное отношения ширины и высоты самого видео. Например, если видео имеет формат 375×240, то, к примеру, при настройках `width="375" height="280"` видео будет центрироваться на 280-пиксельном пространстве в HTML. Что позволяет избавить видео от искажений, которые были бы при растягивании.

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Видео в HTML5</title>
  </head>
<body>
  <video src="cats.mp4" width="400" height="300"
    controls >
```

```
</video>
</body>
</html>
```

Если применить атрибуты `autoplay` и `loop`:

```
<video src="cats.mp4" width="400" height="300" controls
autoplay loop ></video>
```

то видео будет автоматически проигрываться бесконечное число раз.

Если при воспроизведении надо отключить звук, то мы можем воспользоваться атрибутом `muted`:

```
<video src="cats.mp4" width="400" height="300" controls
muted></video>
```

Еще один атрибут - `preload` призван управлять загрузкой видео. Он принимает следующие значения:

- `auto`: видео и связанные с ним метаданные будут загружаться до того, как видео начнет воспроизводиться;
- `none`: видео не будет загружаться в фоне, пока пользователь не нажмет на кнопку начала проигрывания;
- `metadata`: в фоне до воспроизведения будут загружаться только метаданные (данные о формате, длительности и т.д), само видео не загружается

```
<video src="cats.mp4" width="400" height="300"
controls preload="auto"></video>
```

Атрибут `poster` позволяет установить изображение, которое будет отображаться до запуска видео. Этому атрибуту в качестве значения передается путь к изображению:

```
<video src="cats.mp4" width="400" height="300"
controls poster="mycat.jpg"></video>
```

Главной проблемой при использовании элемента `video` является **поддержка** различными браузерами определенных **форматов**. С помощью вложенных элементов `source` можно задать несколько источников видео, один из которых будет использоваться:

```
<video width="400" height="300" controls>
  <source src="cats.mp4" type="video/mp4">
  <source src="cats.webm" type="video/webm">
  <source src="cats.ogv" type="video/ogg">
</video>
```

Элемент `source` использует два атрибута для установки источника видео:

`src`: путь к видеофайлу

`type`: тип видео (MIME-тип)

Если браузер не поддерживает первый тип видео, то он пытается загрузить второй видеофайл. Если же и тип второго видеофайла не поддерживается, то браузер обращается к третьему видеофайлу.

Популярные видеохостинги часто предоставляют **код для встраивания** размещённых на них видео на сайт, обычно этого код построен на элемента `iframe` (<http://htmlbook.ru/html/iframe>), например:

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/AqYsbttfgaQ"
frameborder="0" allow="accelerometer; autoplay; encrypted-
```

```
media; gyroscope; picture-in-picture"
allowfullscreen></iframe>
```

Этот тег можно обычными средствами встроить в макет своей страницы.

4.7. Аудио

Для воспроизведения звука без видео в HTML5 применяется элемент `audio`. Он во многом похож на элемент `video`. Для настройки элемента `audio` мы можем использовать следующие его атрибуты:

- `src`: путь к аудиофайлу;
- `controls`: добавляет элементы управления воспроизведением;
- `autoplay`: устанавливает автовоспроизведение;
- `loop`: задает повторение аудиофайла;
- `muted`: отключает звук по умолчанию;
- `preload`: устанавливает режим загрузки файла

Действие всех этих атрибутов будет аналогично их действию в элементе `video`.

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Аудио в HTML5</title>
  </head>
  <body>
    <audio src="mobile_phone.mp3" controls></audio>
  </body>
</html>
```

В зависимости от браузера внешний вид элементов управления может отличаться.

Ключевым моментом для работы с аудио является поддержка браузером тех или иных форматов. На данный момент подавляющее большинство браузеров поддерживают `mp3`. Однако если у нас есть неуверенность, что наше аудио в определенном формате будет поддерживаться браузером пользователя, то мы можем использовать вложенный элемент `source` и указать аудио в иных форматах:

```
<audio width="400" height="300" controls>
  <source src="mobile_phone.mp3" type="audio/mpeg">
  <source src="mobile_phone.m4a" type="audio/aac">
  <source src="mobile_phone.ogg" type="audio/ogg">
</audio>
```

Как и в случае с элементом `video`, здесь у элемента `source` устанавливается атрибут `src` с ссылкой на файл и атрибут `type` - тип файла.

Задание к лабораторной работе 4:

1. Добавьте на сайт страницу-фотогалерею, выполненную на табличном шаблоне. При щелчке по уменьшенному изображению в фотогалерее открывается новое окно (вкладка) с полноразмерным изображением.
2. Добавьте на сайт страницу с аудио и видео.