

# МДК 09.01 Проектирование и разработка веб-приложений

---

## Лекция: Основы HTML

### План лекции:

1. **HTML Document Structure (Структура HTML-документа):** `html`, `head`, `body`
  2. **Текстовые элементы:** Заголовки (`h1-h6`), параграфы (`p`), выделение текста (`b`, `strong`, `i`, `em`), списки (`ol`, `ul`, `li`)
  3. **Изображения:** Тег `img`
  4. **Навигация. Гиперссылки:** Тег `a`
  5. **Семантическая структура страницы:** `div`, `nav`, `header`, `footer`, `article`, `aside`
  6. **Резюме**
- 

## 1. HTML Document Structure (`html`, `head`, `body`)

Каждый HTML-документ — это, по сути, текстовый файл, структурированный определенным образом. Его основа — теги. Теги заключаются в угловые скобки (`<` `>`) и обычно идут парами: открывающий (`<тег>`) и закрывающий (`</тег>`). Все содержимое помещается между ними.

Базовая структура выглядит так:

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Моя первая страница</title>
</head>
<body>
  <h1>Привет, мир!</h1>
  <p>Это мой первый HTML-документ.</p>
</body>
</html>
```

Давайте разберем каждый элемент:

- `<!DOCTYPE html>`: Это не тег, а декларация. Она сообщает браузеру, какая версия HTML используется в документе. `html` означает, что мы используем актуальную, пятую версию HTML (HTML5).
- `<html>`: Это корневой элемент всего документа. Все остальные теги должны находиться внутри него. Атрибут `lang="ru"` указывает язык документа (в нашем случае — русский), что важно для поисковых систем и screen readers (программ чтения с экрана).

- **<head>**: Это "голова" документа. Её содержимое не отображается напрямую на странице. Здесь размещается служебная информация:
    - **<meta charset="UTF-8">**: Определяет кодировку символов. **UTF-8** включает почти все символы всех языков мира. **Всегда используйте эту кодировку**, чтобы избежать проблем с отображением кириллицы (русских букв).
    - **<meta name="viewport" ...>**: Управляет отображением на мобильных устройствах. Эта строка говорит браузеру, что ширина страницы должна быть равна ширине экрана устройства. **Крайне важный тег для адаптивного дизайна.**
    - **<title>**: Задаёт заголовок страницы. Он отображается на вкладке браузера и используется поисковыми системами.
  - **<body>**: Это "тело" документа. Именно всё, что находится внутри этого тега, будет отображаться в окне браузера. Заголовки, текст, изображения, ссылки — всё здесь.
- 

## 2. Текстовые элементы (h1-h6, p, b и strong, i и em, ol, ul, li)

Это основные "кирпичики" для создания контента.

- **Заголовки: <h1>, <h2>, <h3>, <h4>, <h5>, <h6>** Заголовки определяют иерархию и структуру вашего текста. **<h1>** — заголовок самого верхнего уровня (главный заголовок страницы, должен быть один на странице). **<h6>** — заголовок самого низкого уровня.

```
<h1>Главный заголовок сайта (название статьи)</h1>
<h2>Подзаголовок раздела</h2>
<h3>Название пункта внутри раздела</h3>
<p>Какой-то текст...</p>
<h2>Еще один подзаголовок раздела</h2>
```

- **Параграф: <p>** Используется для обычного текста. Браузер автоматически добавляет отступы до и после каждого параграфа.

```
<p>Это первый параграф текста. Он содержит несколько предложений и
автоматически переносится на новую строку после закрывающего тега.</p>
<p>Это второй параграф. Между ним и первым будет небольшой отступ (маргин),
созданный стилями браузера по умолчанию.</p>
```

- **Выделение текста:**
  - **<b> и <strong>**: Визуально оба делают текст **жирным**. Но есть семантическое различие:
    - **<b>** — просто визуальное выделение, не придающее тексту особой важности.
    - **<strong>** — указывает на **важность** содержимого. Поисковые системы и screen readers обращают на него внимание.
  - **<i> и <em>**: Визуально оба делают текст *курсивным*.
    - **<i>** — используется для текста, который по какой-то причине отличается от обычного (например, технический термин, иностранное слово).

- `<em>` — используется для *семантического выделения*, чтобы сделать **акцент** на слове или фразе. При чтении вслух screen reader может изменить интонацию.

```
<p>Это <b>важное</b> слово выделено жирным (b).</p>
<p>А это <strong>критически важное</strong> слово выделено тегом strong.</p>

<p>Название книги — <i>Война и мир</i> выделено курсивом (i).</p>
<p>Я <em>действительно</em> хочу подчеркнуть это слово с помощью em.</p>
```

- **Списки:** Существуют нумерованные (упорядоченные) и маркированные (неупорядоченные) списки.
  - `<ul>` (**Unordered List**): Маркированный список. Каждый элемент по умолчанию отмечается маркером (точкой).
  - `<ol>` (**Ordered List**): Нумерованный список. Каждый элемент отмечается цифрой.
  - `<li>` (**List Item**): Элемент списка. Находится внутри `<ul>` или `<ol>`.

```
<h3>Мой маркированный список покупок (ul):</h3>
<ul>
  <li>Хлеб</li>
  <li>Молоко</li>
  <li>Яблоки</li>
</ul>

<h3>Мой нумерованный список дел (ol):</h3>
<ol>
  <li>Проснуться</li>
  <li>Выпить кофе</li>
  <li>Изучить HTML</li>
</ol>
```

Списки можно легко вкладывать друг в друга, создавая сложные структуры.

---

### 3. Изображения (img)

Тег `<img>` используется для вставки изображений на страницу. Это **оди́ночный (самозакрывающийся) тег**, ему не нужен закрывающий тег.

**Важнейшие атрибуты:**

- `src` (source) — **обязательный атрибут**. Указывает путь к файлу изображения. Это может быть относительный путь (`images/my-cat.jpg`) или абсолютный URL (`https://site.com/cat.jpg`).
- `alt` (alternative text) — **обязательный атрибут для доступности**. Содержит текст, который описывает изображение. Он отображается, если картинка не загрузилась, и крайне важен для незрячих пользователей и SEO.
- `title` — Необязательный атрибут. Задаёт всплывающую подсказку, которая появляется, когда пользователь наводит курсор на изображение.

- **width** и **height** — Задают ширину и высоту изображения в пикселях. Рекомендуется задавать их для резервирования места на странице во время загрузки, чтобы избежать "прыгающего" layout'a.

**Пример:**

```


```

## 4. Навигация. Hyperlinks.

Тег `<a>` (anchor — якорь) создает гиперссылку, самый важный элемент Всемирной паутины, связывающий страницы между собой.

**Ключевые атрибуты:**

- **href** (hypertext reference) — **обязательный атрибут**. Определяет URL-адрес, на который ведет ссылка. Это может быть:
  - Абсолютный URL: `href="https://www.google.com"`
  - Относительный путь к другой странице на вашем сайте: `href="about.html"`
  - Ссылка на якорь на той же странице: `href="#chapter1"`
  - Ссылка на email-адрес: `href="mailto:example@mail.com"`
- **target** — Указывает, где открыть ссылку. Чаще всего используется значение `_blank` для открытия в новой вкладке/окне браузера.
- **title** — Всплывающая подсказка при наведении.

**Примеры:**

```
<!-- Обычная ссылка на внешний сайт -->
<p>Посетите <a href="https://yandex.ru" target="_blank" title="Поиск от
Яндекса">Яндекс</a> для поиска информации.</p>

<!-- Ссылка на другую страницу вашего сайта -->
<p>Узнать больше о нас можно на <a href="about.html">странице "О компании"</a>.
</p>

<!-- Ссылка-якорь (предположим, что на странице есть элемент с id="contacts") -->
<p><a href="#contacts">Перейти к контактам</a> (вниз страницы)</p>
...
<h2 id="contacts">Наши контакты</h2>

<!-- Ссылка для отправки email -->
<p><a href="mailto:support@my-site.com">Напишите нам</a></p>
```

## 5. Структура страницы (div, nav, header, footer, article, aside)

Раньше для создания структуры (шапка, меню, основной контент, боковая панель, подвал) использовался практически один тег `<div>` (контейнер, блочный элемент без собственного смысла). С появлением HTML5 были добавлены **семантические теги**. Они придают структуре смысл, что лучше для SEO, доступности и читаемости кода.

- **`<div>`**: Универсальный блочный контейнер. Используется для группировки элементов в чисто стилистических целях, когда не подходит ни один семантический тег. `</div>` — это "коробка" без смысла.
- **`<header>`**: Задаёт "шапку" сайта или раздела. Обычно содержит логотип, навигацию, заголовок.
- **`<nav>`** (navigation): Предназначен **только** для основного меню навигации по сайту. Не всякую группу ссылок нужно оборачивать в `<nav>`, только основную.
- **`<footer>`**: Задаёт "подвал" сайта или раздела. Обычно содержит информацию об авторских правах, контакты, второстепенную навигацию.
- **`<article>`**: Представляет самостоятельную, независимую часть content'a: пост в блоге, новость, статью, комментарий.
- **`<aside>`**: Представляет содержимое, косвенно связанное с основным контентом (сайдбар, блок с рекламой, цитата, glossary).

#### Пример структуры всей страницы с семантическими тегами:

```
<body>
  <!-- Шапка сайта -->
  <header>
    <h1>Название моего сайта</h1>
    <!-- Главное меню -->
    <nav>
      <ul>
        <li><a href="index.html">Главная</a></li>
        <li><a href="about.html">О нас</a></li>
        <li><a href="contact.html">Контакты</a></li>
      </ul>
    </nav>
  </header>

  <!-- Основное содержание страницы -->
  <main>
    <!-- Независимая статья -->
    <article>
      <h2>Заголовок статьи</h2>
      <p>Текст статьи...</p>
    </article>

    <!-- Боковая панель -->
    <aside>
      <h3>Интересные ссылки</h3>
      <ul>
        <li><a href="#">Ссылка 1</a></li>
        <li><a href="#">Ссылка 2</a></li>
      </ul>
    </aside>
  </main>
```

```
<!-- Подвал сайта -->
<footer>
  <p>&copy; 2023 Мой сайт. Все права защищены.</p>
</footer>
</body>
```

Использование этих тегов делает код понятным не только для браузера, но и для вас, и для других разработчиков.

---

## 6. Резюме

Сегодня мы заложили фундамент всего веб-программирования. Давайте кратко повторим самое главное:

1. **HTML** — это язык разметки, который определяет **структуру** и **содержание** веб-страницы, но не её внешний вид (это задача CSS).
2. Любой документ начинается с декларации `<!DOCTYPE html>` и имеет базовую структуру: `<html>` - `>` (`<head>` + `<body>`).
3. В `<head>` помещается служебная информация (кодировка, заголовок, мета-теги), а в `<body>` — весь видимый контент.
4. Мы изучили основные текстовые теги: заголовки (`h1-h6`), параграфы (`p`), выделение (`b`, `strong`, `i`, `em`) и списки (`ul/ol/li`).
5. Для вставки изображений используется **одиночный** тег `<img>` с **обязательными** атрибутами `src` и `alt`.
6. Гиперссылки создаются с помощью тега `<a>` и атрибута `href`. Они связывают страницы в единую сеть.
7. Для построения логической структуры страницы следует использовать **семантические теги** HTML5 (`header`, `nav`, `main`, `article`, `aside`, `footer`), а не только универсальные `<div>`.

**Ваш следующий шаг:** открыть редактор кода (например, VS Code), создать файл `index.html` и набрать вручную все примеры из этой лекции. Посмотрите, как они выглядят в браузере. Поэкспериментируйте, измените атрибуты, добавьте свои теги. Практика — единственный способ освоить HTML.