

Лекция-3

Семантика HTML, HTML5. Атрибуты тегов. Валидация HTML.

Что такое семантика?

Семантика в контексте веб-разработки относится к использованию элементов HTML для передачи значения и смысла содержимого веб-страницы, которые можно интерпретировать не только визуально, но и с точки зрения их значения и функциональности для браузера и поисковых систем.

Семантический элемент — это элемент HTML, который несет определенное значение или смысл, помогающий браузерам и другим программным инструментам понять структуру и содержание веб-страницы без необходимости визуального представления. Использование семантических элементов не только улучшает доступность и интерпретацию содержимого для пользователя, но и повышает SEO-показатели и упрощает поддержку страниц.

SEO (Search Engine Optimization) оптимизация — это процесс улучшения видимости и ранжирования веб-сайта или веб-страницы в результатах поисковых систем (например, Google, Яндекс и других). Целью SEO оптимизации является привлечение большего количества органического трафика на сайт через улучшение позиций в поисковых запросах, связанных с содержанием сайта.

HTML5

HTML5 (HyperText Markup Language, version 5) представляет собой последнюю версию стандарта HTML, который используется для создания и структурирования веб-страниц. HTML5 был разработан с целью улучшения функциональности и возможностей веб-приложений, поддержки мультимедийного контента и обеспечения совместимости с современными браузерами.

Что дает нам новая версия HTML??

- приход семантики и новых элементов;
- формы и валидация HTML;
- встроенная поддержка мультимедиа;
- графика и визуализация (приход canvas);
- появление локального хранилища данных;
- поддержка совместимости различных устройств;
- улучшение структурности;

Категории контента HTML5.

В HTML5 категории контента — это группы элементов, которые помогают структурировать и организовывать содержимое веб-страницы согласно их функциональному назначению или семантике. HTML5 ввел новые семантические элементы, которые относятся к определенным категориям контента, что делает код более понятным, доступным для поисковых систем и вспомогательных технологий.

- Metadata content (например, `<meta>`, `<title>`)
- Flow content (почти все элементы, включая текст и другие блоки)
- Sectioning content (например, `<header>`, `<footer>`, `<article>`, `<section>`)
- Heading content (например, `<h1>` – `<h6>`)
- Phrasing content (например, текст, `<a>`, ``, ``, ``)
- Embedded content (например, ``, `<audio>`, `<video>`)
- Interactive content (например, `<a>`, `<button>`, `<input>`)

T&E. Семантические элементы HTML5

- **<header>**: Определяет верхнюю часть документа или секции.
- **<footer>**: Определяет нижнюю часть документа или секции.
- **<nav>**: Определяет блок навигации на странице.
- **<article>**: Определяет самостоятельный контент, который может быть переиспользован, например, новость в блоге.
- **<section>**: Определяет раздел документа.
- **<aside>**: Определяет дополнительный контент, который не является главным содержимым страницы.
- **<main>**: Основное содержимое документа.
- **<figure>**: Вставка независимого контента с подписью.
- **<figcaption>**: Подпись к содержимому в теге <figure>.
- **<details>**: Блок с дополнительной информацией, сворачиваемый по клику.
- **<summary>**: Заголовок для блока <details>, отображаемый при свёрнутом состоянии.
- **<time>**: Определение даты или времени.
- **<address>**: Контактные данные автора или владельца документа.
- **<mark>**: Выделение важной информации.
- **<blockquote>**: Цитата из другого источника.
- **<q>**: Короткая цитата.
- **<abbr>**: Аббревиатура или акроним.
- **<ruby>**: Аннотация для иероглифов.
- **<datalist>**: Список predetermined опций для выбора значения.

T&E. Семантические элементы HTML5 - пример №1

```
1 <header>
2   <h1>Заголовок страницы</h1>
3   <nav>
4     <ul>
5       <li><a href="#">Главная</a></li>
6       <li><a href="#">О нас</a></li>
7       <li><a href="#">Контакты</a></li>
8     </ul>
9   </nav>
10 </header>
11 <section>
12   <h2>Основной контент</h2>
13   <article>
14     <h3>Статья 1</h3>
15     <p>Текст статьи 1</p>
16   </article>
17   <article>
18     <h3>Статья 2</h3>
19     <p>Текст статьи 2</p>
20   </article>
21 </section>
22 <footer>
23   <p>© 2022 Ваше название компании</p>
24 </footer>
```

Заголовок страницы

- [Главная](#)
- [О нас](#)
- [Контакты](#)

Основной контент

Статья 1

Текст статьи 1

Статья 2

Текст статьи 2

© 2022 Ваше название компании

T&E. Семантические элементы HTML5 - пример №2

```
...<body>
<header>
  <h1>Заголовок сайта </h1>
  <nav>
    ...
  </nav>
</header>
<main>
  <article>
    <h2>Новость </h2>
    <p>Описание новости... </p>
    <footer>
      <time datetime="2024-05-12">12 мая 2024 </time>
    </footer>
  </article>
  <aside>
    <h3>Дополнительная информация </h3>
    <p>Боковая панель с дополнительной информацией. </p>
  </aside>
</main>
<footer>
  <p>&copy; 2024 Компания. Все права защищены. </p>
</footer></body>
```

Заголовок сайта

- [Главная](#)
- [О нас](#)
- [Контакты](#)

Новость

Описание новости...

12 мая 2024

Дополнительная информация

Боковая панель с дополнительной информацией.

© 2024 Компания. Все права защищены.

Тег <time>

Тег <time> в HTML используется для обозначения времени или даты в тексте. Это может быть как точное время (включая часы и минуты), так и дата.

Например.

```
<p>Встреча состоится в <time datetime="2024-07-21T15:00">15:00</time>.</p>
```



Встреча состоится в 15:00.

Тег <figure> и <figcaption>

<figure> используется для группировки любого содержимого, которое должно быть связано с основным контентом документа, например, изображения, диаграммы, видео и т.д. **<figcaption>** используется внутри <figure> для добавления подписи к такому содержимому.

Например.

```
<figure>  
    
  <figcaption>Подпись к изображению</figcaption>  
</figure>
```

Тег <details> и <summary>

<details> используется для создания раскрывающегося блока с дополнительной информацией, который может быть скрыт или открыт по желанию пользователя. **<summary>** используется внутри <details> для создания заголовка или кнопки, открывающей или закрывающей блок <details>.

Например.

```
<details>  
  <summary>Дополнительная информация</summary>  
  <p>Здесь может быть дополнительный текст или содержимое</p>  
</details>
```

▼ Дополнительная информация

Здесь может быть дополнительный текст или содержимое.

Атрибуты Тегов.

Атрибут тега в HTML представляет собой дополнительную информацию, которая задается внутри открывающего тега элемента и обычно содержит пару "имя-значение". Атрибуты используются для указания дополнительных характеристик или свойств элемента, таких как идентификаторы, классы, стили, ссылки и многое др.

Атрибуты можно разделить на две ключевые группы:

- **Глобальные атрибуты;**

Глобальные атрибуты могут быть применены к большинству элементов в HTML. Они не зависят от типа элемента и могут использоваться практически везде в HTML разметке.

Примеры включают class, id, style, title, data-*, lang, dir, hidden, и tabindex.

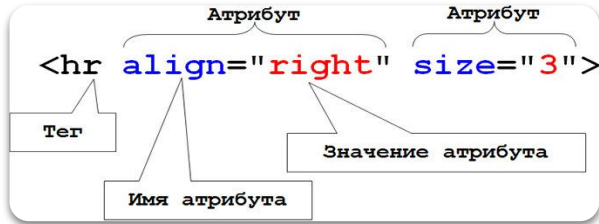
- **Специфичные атрибуты;**

Специфичные атрибуты привязаны к определенным элементам или группам элементов и обеспечивают дополнительную функциональность или семантику.

Примеры включают src, href, type, value, alr, rows, cols и другие.

Атрибуты Тегов. Структура + пример.

Структура указания атрибута:



Пример применения:

```
<h1 id="header" class="header-cls" style="color: red;">Заголовок-1</h1>
```

```
<p class="hello">Hello, world!</p>
```

Глобальные атрибуты.

- **class**: используется для присвоения одного или нескольких классов элементу. Классы определяют стили, которые могут быть применены с помощью CSS.
- **id**: задает уникальный идентификатор элемента на странице, который может использоваться для ссылок на элемент или для стилизации с помощью CSS или для скриптов.
- **style**: позволяет применять стили CSS напрямую к элементу.
- **title**: используется для предоставления всплывающей подсказки при наведении курсора на элемент.
- **data-***: позволяют хранить пользовательские данные в элементе HTML. Они полезны для передачи данных на стороне клиента или для структурирования информации.
- **lang**: определяет язык текста внутри элемента. Это помогает поисковым системам и вспомогательным технологиям правильно интерпретировать содержимое.
- **dir**: определяет направление текста в элементе, такое как слева направо (ltr) или справа налево (rtl).
- **hidden**: скрывает элемент отображения на странице. Он полезен для управления видимостью элементов динамически с помощью JavaScript.
- **tabindex**: определяет порядок навигации между интерактивными элементами с клавиатуры. Он позволяет управлять фокусировкой элементов при использовании клавиши Tab.

Специфичные атрибуты.

- **src:** указывает источник ресурса, такого как изображение, аудио или видео файл, который должен быть загружен и отображен.
- **href:** определяет целевой URL ссылки для элемента `<a>`, который указывает, куда перейти при клике на ссылку.
- **alt:** задает альтернативный текст для элементов, таких как изображения, который отображается в случае, если изображение недоступно или его не удалось загрузить.
- **type:** определяет тип данных элемента формы `<input>`, например, текст, пароль, email и другие.
- **value:** устанавливает начальное значение для элементов формы, таких как `<input>` или `<button>`.
- **checked:** используется с элементами формы, такими как `<input type="checkbox">` или `<input type="radio">`, чтобы установить их начальное состояние выбора.
- **placeholder:** определяет подсказывающий текст, который отображается в элементе формы, пока пользователь не введет свои данные.
- **rows и cols:** используются с элементом `<textarea>` для определения количества строк и столбцов текстового поля соответственно.
- **span:** используется в контексте таблиц для определения числа столбцов, которые ячейка должна занимать.
- **target:** задает окно или фрейм, в котором открывается ссылка, указанная в элементе `<a>`.
- **disabled:** отключает интерактивность элемента, делая его недоступным для взаимодействия пользователем.
- **autocomplete:** определяет, должен ли браузер предложить автозаполнение для элементов формы.
- **maxlength:** указывает максимальное количество символов, которые пользователь может ввести в элемент формы.
- **min и max:** используются с элементами формы, например `<input type="number">`, для задания минимального и максимального допустимого значения.
- **rel:** Определяет отношения между текущим и связанным документом, например, для стилей CSS или скриптов.
- и другие...

Пример с атрибутами id и class

```
<body>
  <div id="header" class="container">
    <h1 id="main-heading">Добро пожаловать на наш сайт</h1>
    <p>Здесь вы найдете много интересного контента</p>
  </div>

  <div id="content" class="container">
    <h2>О нас</h2>
    <p>Наша компания занимается созданием качественных веб-сайтов</p>
  </div>

  <div id="footer" class="container">
    <p>&copy; 2024 Все права защищены</p>
  </div>
</body>
```

Пример с атрибутом alt + width/height

```
<body>
  <div>
    <p>..</p>
  </div>
  <h1>Пример изображения с атрибутами</h1>
  
</body>
```


Пример с атрибутом href

```
<body>  
  <h1>Пример гиперссылки с атрибутом href</h1>  
  <a href="https://www.example.com">Перейти на сайт примера</a>  
</body>
```

Пример с атрибутом type/value/placeholder

```
<body>
  <h1>Форма:</h1>
  <form action="/submit-form" method="post">
    <label for="username">Имя пользователя:</label>
    <input type="text" id="username" name="username" placeholder="Введите ваше имя" required>
    <br><br>
    <label for="email">Электронная почта:</label>
    <input type="email" id="email" name="email" placeholder="example@example.com" required>
    <br><br>
    <label for="password">Пароль:</label>
    <input type="password" id="password" name="password" minlength="8" placeholder="Введите пароль" required>
    <br><br>
    <input type="submit" value="Отправить">
  </form>
</body>
```

Пример с атрибутом min/max

```
<body>  
  <h1>Пример числового ввода с атрибутами min/max</h1>  
  <label for="quantity">Количество товара:</label>  
  <input type="number" id="quantity" name="quantity" min="1" max="10" value="1">  
  
  <p>Выберите количество товара от 1 до 10</p>  
</body>
```

Валидация HTML

Валидация HTML - это процесс проверки HTML-кода на соответствие официальным стандартам и спецификациям языка, установленным W3C (World Wide Web Consortium). Валидация помогает разработчикам убедиться, что их HTML-код написан корректно и соответствует стандартам, что важно для обеспечения правильного отображения и интерпретации веб-страниц всеми браузерами и устройствами.

Зачем валидировать HTML?

1. **Соответствие стандартам:** Валидация помогает убедиться, что ваш HTML-код соответствует стандартам, что улучшает доступность и переносимость вашего контента.
2. **Предотвращение ошибок:** Валидация помогает обнаружить и исправить синтаксические ошибки и недочеты в коде, которые могут повлиять на работу веб-страницы.
3. **Улучшение SEO:** Валидный HTML может способствовать улучшению показателей SEO, так как правильный и чистый код облегчает поисковым системам понимание структуры и содержания страницы.


Проведение валидации.

- **Использование онлайн-сервисов:** Существуют различные онлайн-инструменты, такие как W3C Markup Validation Service (<https://validator.w3.org/>), которые позволяют загружать HTML-код или указывать URL страницы для валидации.
- **Интеграция в IDE и редакторы кода:** Многие современные интегрированные среды разработки (IDE) и текстовые редакторы предлагают встроенные инструменты для валидации HTML, которые могут автоматически проверять ваш код на лету.
- **Локальные инструменты:** Вы можете использовать локальные программы или скрипты для валидации HTML, что полезно при автоматизации процессов сборки и развертывания веб-приложений.

При валидации HTML проверяется:

- Синтаксис HTML-кода на соответствие стандартам.
- Наличие всех обязательных элементов и атрибутов в соответствии с выбранной DOCTYPE.
- Верное использование атрибутов, правильное вложение элементов и другие аспекты, касающиеся структуры HTML-документа.

W3C

 Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI Validate by File Upload Validate by Direct Input

Validate by URI

Validate a document online:

Address:

▼ More Options

Character Encoding (detect automatically) ☐ Only if missing

Document Type (detect automatically) ☐ Only if missing

☒ List Messages Sequentially ☐ Group Error Messages by Type

☐ Show Source ☐ Clean up Markup with HTML-Tidy

☐ Show Outline ☐ Validate error pages ☐ Verbose Output

Check

This validator checks the [markup validity](#) of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as [RSS/Atom feeds](#) or [CSS stylesheets](#), [MobileOK content](#), or to [find broken links](#), there are [other validators and tools](#) available. As an alternative you can also try our [non-DTD-based validator](#).



The W3C validators rely on community support for hosting and development.

[Donate](#) and help us build better tools for a better web.

[Home](#) [About...](#) [News](#) [Docs](#) [Help & FAQ](#) [Feedback](#) [Contribute](#)



COPYRIGHT © 2024 WORLD WIDE WEB CONSORTIUM. W3C® LIABILITY, TRADEMARK AND PERMISSIVE LICENSE RULES APPLY.



W3C (World Wide Web Consortium) -

это международная организация, устанавливающая стандарты для веб-технологий, таких как HTML, CSS, SVG и другие. Она разрабатывает и поддерживает спецификации, которые определяют, как должны взаимодействовать браузеры и другие веб-технологии с веб-документами.

Кнопка “check” - для проверки.