

HTML 5

HTML5 семантические элементы

Семантика - это изучение значений слов и фраз в языке. Семантические элементы = элементы со смыслом.

Что такое семантические элементы?

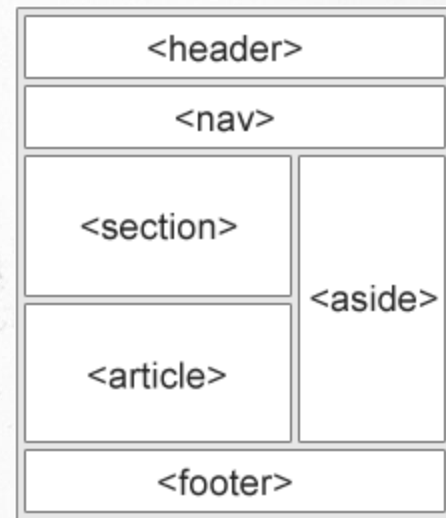
Семантический элемент четко описывает его значение как для браузера, так и для разработчика.

Примеры несемантических элементов: `<div>` и `` - ничего не говорит о его содержимом.

Примеры семантических элементов: `<form>`, `<table>` и `<article>` - четко определяет его содержание. Многие веб-сайты содержат HTML-код, например: `<div id = "nav"> <div class = "header"> <div id = "footer">`

чтобы указать навигацию, верхний и нижний колонтитулы.

HTML5 предлагает новые семантические элементы для определения различных частей веб-страницы:



HTML 5

HTML5 семантические элементы

Элемент **<section>** определяет раздел в документе.

Согласно документации W3C по HTML5: «Раздел - это тематическая группировка контента, обычно с заголовком».

Домашнюю страницу обычно можно разделить на разделы для введения, содержания и контактной информации.

Элемент **<article>** определяет независимое, автономное содержимое.

Статья должна иметь смысл сама по себе, и должна быть возможность читать ее независимо от остальной части веб-сайта.

Примеры использования элемента **<article>**:

Сообщение на форуме, Сообщение блога, Газетная статья

HTML 5

Вложение `<article>` в `<section>` или наоборот?

Элемент `<article>` определяет независимое, автономное содержимое.

Элемент `<section>` определяет раздел в документе.

Можем ли мы использовать определения, чтобы решить, как вложить эти элементы? Нет мы не можем!

Итак, в Интернете вы найдете HTML-страницы с элементами `<section>`, содержащими элементы `<article>`, и элементами `<article>`, содержащими элементы `<section>`.

Вы также найдете страницы с элементами `<section>`, содержащими элементы `<section>`, и элементами `<article>`, содержащими элементы `<article>`.

Пример для газеты: Спорт `<article>` в разделе(`<section>`) sport может содержать технический раздел(`<section>`) в каждой `<article>`.

HTML 5

Вложение `<article>` в `<section>` или наоборот?

Элемент `<header>` определяет заголовок для документа или раздела.

Элемент `<header>` должен использоваться в качестве контейнера для вводного содержимого.

Вы можете иметь несколько элементов `<header>` в одном документе.

Элемент `<footer>` определяет нижний колонтитул для документа или раздела.

Элемент `<footer>` должен содержать информацию о его содержащем элементе.

Нижний колонтитул обычно содержит автора документа, информацию об авторских правах, ссылки на условия использования, контактную информацию и т. Д.

У вас может быть несколько элементов `<footer>` в одном документе.

HTML 5

Вложение <article> в <section> или наоборот?

Элемент **<nav>** определяет набор навигационных ссылок. Обратите внимание, что НЕ все ссылки документа должны быть внутри элемента **<nav>**. Элемент **<nav>** предназначен только для основного блока навигационных ссылок.

Элемент **<aside>** определяет некоторое содержимое помимо содержимого, в котором он находится (например, боковую панель). Содержание **<aside>** должно быть связано с окружающим контентом.

HTML5 **<figure>** и **<figcaption>**. Цель **<figcaption>** - добавить визуальное объяснение к изображению.

В HTML5 изображение и подпись могут быть сгруппированы в элемент **<figure>**

```
<figure>
  
  <figcaption>Fig1. - Trulli, Puglia, Italy.</figcaption>
</figure>
```

HTML 5

Почему семантические элементы?

В HTML4 разработчики использовали свои собственные имена идентификаторов / классов для оформления элементов: header, top, bottom, footer, menu, navigation, main, container, content, article, sidebar, topnav, и т. Д.

Это сделало невозможным для поисковых систем определить правильное содержание веб-страницы.

С новыми элементами HTML5 (<header> <footer> <nav> <section> <article>) это станет проще.

Согласно W3C, Semantic Web: «Позволяет обмениваться данными и повторно использовать их в приложениях, предприятиях и сообществах».

Typical HTML4

`<div id="header">`

`<div id="menu">`

`<div id="content">`

`<div class="article">`

`<div id="footer">`

Typical HTML5

`<header>`

`<nav>`

`<section>`

`<article>`

`<footer>`

HTML 5

Разница между `<article>` `<section>` и `<div>`

В HTML4 разработчики использовали свои собственные имена идентификаторов / классов для оформления элементов: header, top, bottom, footer, menu, navigation, main, container, content, article, sidebar, topnav, и т. Д.

Это сделало невозможным для поисковых систем определить правильное содержание веб-страницы.

С новыми элементами HTML5 (`<header>` `<footer>` `<nav>` `<section>` `<article>`) это станет проще.

Согласно W3C, Semantic Web: «Позволяет обмениваться данными и повторно использовать их в приложениях, предприятиях и сообществах».

HTML 5

Руководство по стилю HTML5

Будте последовательными и планируйте верстку.

Всегда держите ваш код аккуратным, чистым и структурированным.

Используйте имена тегов в нижнем регистре

HTML5 позволяет смешивать прописные и строчные буквы в именах тегов.

```
<Section>  
  <p>This is a paragraph.</p>  
</SECTION>
```

Закрывайте все элементы HTML

В HTML5 вам не нужно закрывать все теги(например, <p>).

```
<section>  
  <p>This is a paragraph.  
  <p>This is a paragraph.  
</section>
```

Закрывайте пустые теги HTML

В HTML5 необязательно закрывать пустые теги.

```
<meta charset="utf-8">
```

HTML 5

Руководство по стилю HTML5

Используйте имена атрибутов в нижнем регистре

HTML5 позволяет смешивать прописные и строчные буквы в именах атрибутов.

```
<div CLASS="menu">
```

Используйте кавычки для значений атрибутов

HTML5 позволяет устанавливать значения атрибутов без кавычек.

```
<table class=table striped>
```

Пробелы и знаки равенства

HTML5 позволяет оставлять пробелы вокруг знаков равенства. Но код без пробелов легче читать и лучше объединять в группы.

```
<link rel = "stylesheet" href = "styles.css">
```

Избегайте длинных строк кода

При использовании редактора HTML неудобно прокручивать вправо и влево, чтобы прочитать код HTML. Старайтесь избегать строк кода длиннее 80 символов.

Не добавляйте пустые строки без причины.

Для удобства чтения добавляйте пустые строки для разделения больших или логических блоков кода.