



Урок 1

HTML5: семантические элементы

Обзор HTML5. Новые семантические элементы HTML5. Новый способ структурирования страниц. Семантика текстового уровня

[Введение](#)

[Обзор HTML5](#)

[Структура HTML5](#)

[Новые семантические элементы HTML5](#)

[Новый способ структурирования страниц](#)

[Семантические элементы для работы со структурой страниц. Секционные элементы](#)

[Элемент <article>](#)

[Элемент <section>](#)

[<article> внутри <section>](#)

[Элемент <aside>](#)

[Элемент <footer>](#)

[Элемент <address>](#)

[Элемент <figure>](#)

[Элемент <header>](#)

[Элемент <hgroup>](#)

[Элемент <nav>](#)

[Элемент <main>](#)

[Семантика для текстового содержимого](#)

[Элемент <time>](#)

[Элемент <mark>](#)

[Элемент <bdi>](#)

[Элемент <wbr>](#)

[Применение HTML5 на практике](#)

[Материалы для домашнего изучения](#)

[Практическое задание](#)

[Дополнительные материалы](#)

[Используемая литература](#)

Введение

Рад приветствовать вас на курсе, посвященном новшествам в создании сайтов.

Данный курс рассчитан на пользователей, которые уже изучили стандарт HTML и CSS. Не будем изучать основы и синтаксис, а сразу окунемся в новый функционал и возможности HTML5 и CSS3.

На этом курсе мы научимся делать наши страницы живыми, интересными и современными. В результате у вас будет готовый проект, который вы сможете добавить в свое портфолио (конечно, если выполните программу курса).

На ваших глазах во время курса будет создан рекламный сайт организации с применением всех современных методов разработки интернет-страниц.

Обзор HTML5

Структура HTML5

Документ, написанный на HTML5, начинается с указания типа документа, после чего дается его название, а потом – содержимое.

Минимальная структура HTML5-документа:

```
<!DOCTYPE html>
<title>Первый документ на HTML5</title>
<p>Минимальная структура HTML5-документа</p>
```

Общая структура HTML5-документа:

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title></title>
    <link href="styles.css" rel="stylesheet">
    <script src="scripts.js"></script>
  </head>
  <body>
    <p>Первый документ с использованием HTML5</p>
  </body>
</html>
```

Сравним со структурой документа, написанного на HTML предыдущей версии:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
```

```

    <title>Hello HTML</title>
    <meta content="text/html; charset=Windows-1251"
http-equiv="content-type"/>
</head>
<body>
...
</body>
</html>

```

Сейчас более подробно рассмотрим структуру HTML5-документа.

<!DOCTYPE html> – указывается в первой строке каждого HTML5-документа. Это описание ясно указывает, что далее следует HTML5-содержимое.

<html> – традиционный элемент документа HTML. Он не влияет на обработку браузером остального кода страницы. В HTML5 его можно использовать или не использовать по желанию. Атрибутом HTML5 является lang="ru" – он определяет язык содержимого в создаваемом документе.

<head> и <body> – традиционные элементы разделов. Полезны для облегчения восприятия документа, т.к. они четко разделяют информацию о странице (заголовок страницы) и само содержимое (основная часть страницы).

Использование элементов <html>, <head> и <body> – вопрос стиля. Страница без этих элементов будет работать в любом браузере.

<meta charset="utf-8"> – тег, определяющий кодировку символов в документе. В HTML5 он стал короче.

<title> – обозначает заголовок документа.

<link href="styles.css" rel="stylesheet"> – указывает требуемую таблицу стилей. Атрибут type="text/css", который требовался ранее, больше не нужен.

<script src="scripts.js"> – вставка в веб-документ кода JavaScript по ссылке на внешний файл. Атрибут language="JavaScript" не обязателен, т.к., если не указан какой-либо другой язык сценариев, браузеры автоматически предполагают, что используется JavaScript.

Новые семантические элементы HTML5

До этого момента мы рассмотрели изменения в синтаксисе HTML5. Но важнее, какие элементы были добавлены, удалены или изменены в этой версии языка.

Добавленные элементы:

| | |
|------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Семантические элементы для работы со структурой страниц. Секционные элементы | <article>, <aside>, <figcaption>, <figure>, <footer>, <header>, <hgroup>, <nav>, <section>, <details>, <summary>, <main> |
| Семантические элементы для работы с текстом | <mark>, <time>, <wbr> (поддерживался и ранее, но теперь официально является частью языка) |
| Элементы для работы с веб-формами и интерактивности | <input> (старый элемент, но со многими новыми подтипами), <datalist>, <keygen>, |

| | |
|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <code><meter></code> , <code><progress></code> , <code><command></code> , <code><menu></code> , <code><output></code> |
| Элементы для поддержки аудио, видео и подключаемых модулей | <code><audio></code> , <code><video></code> , <code><source></code> , <code><track></code> , <code><embed></code>
(поддерживался и ранее, но теперь официально является частью языка) |
| Поддержка холста | <code><canvas></code> |

Были удалены элементы оформления, такие как `<big>`, `<center>`, ``, `<tt>` и `<strike>`.

Новый способ структурирования страниц

Новые семантические элементы HTML5 позволяют улучшить структуру веб-страницы, добавляя смысловое значение заключенному в них содержимому.

Все семантические элементы имеют отличительную особенность: их использование никак не отражается на внешнем виде страницы – в отличие, например, от элемента `<video>`, который добавляет на веб-страницу полноценный видеоплеер. Зачем тогда они нужны?

Рассмотрим некоторые причины:

- более удобное редактирование и сопровождение;
- оптимизация страницы для поисковых движков. В настоящее время поисковые роботы уже проверяют индексируемые страницы на наличие некоторых семантических элементов HTML5, чтобы собрать о них всю возможную информацию;
- поддержка будущих возможностей. В новых браузерах и инструментах редактирования веб-страниц будет использоваться весь диапазон предоставляемых семантическими элементами возможностей.

Рассмотрим назначение новых элементов более подробно.

Семантические элементы для работы со структурой страниц. Секционные элементы

Элемент `<article>`

Используется для группировки записей – публикаций, статей, записей блога, комментариев. Представляет собой независимый обособленный блок, предназначенный для многократного использования. Как правило, начинается с заголовка. Может дублироваться на других страницах сайта и содержать внутри другие элементы `<article>`, которые близки по содержанию к внешней статье. Если на странице только одна статья с заголовком и текстовым содержанием, её не нужно обертывать элементом `<article>`.

```
<article>
  <header>
    <h2>...</h2>
  </header>
  <p>...</p>
  <footer>
    Опубликовано в категории <a href="/?cat=2">Музыка</a>.
```

```
<a href="/?p=17#respond">0 комментариев</a>
</footer>
</article>
```

Элемент <section>

Представляет собой универсальный раздел документа. Группирует тематическое содержимое и обычно содержит заголовок. Один <section> не используется многократно. Не является блоком-оберткой: для этих целей уместнее использовать элемент <div>. В качестве содержимого может выступать оглавление, разделы научных публикаций, программа мероприятия. Домашняя страница сайта также может быть поделена на секции – блок вводной информации, новости и контакты.

```
<article>
  <h1>...</h1>
  <section>
    <h2>...</h2>
    <p>...</p>
  </section>
  <section>
    <h2>...</h2>
    <p>...</p>
  </section>
  <p>...</p>
</article>
```

<article> внутри <section>

Можно создавать родительские элементы <section> с вложенными элементами <article>, в которых есть один или несколько элементов <article>. Не все страницы должны быть устроены именно так, но это допустимый способ вложения элементов. Например, основная область контента страницы содержит два блока со статьями разной тематики. Можно сделать на этом акцент, поместив статьи одной тематики внутрь элемента <section>.

```
<section>
  <h1>Заметки о природе</h1>
  <article>
    <h2>...</h2>
    <p>...</p>
  </article>
  <article>
    <h2>...</h2>
    <p>...</p>
  </article>
</section>
<section>
  <h1>Исторические заметки</h1>
  <article>
    <h2>...</h2>
    <p>...</p>
  </article>
```

```
<article>
  <h2>...</h2>
  <p>...</p>
</article>
</section>
```

Элемент <aside>

Группирует содержимое, связанное с окружающим его контентом напрямую, которое можно счесть отдельным (т. е. удаление этого блока не повлияет на понимание основного содержимого). Чаще всего элемент позиционируется как боковая колонка (как в книгах) и включает в себя группу элементов: <nav>, цифровые данные, цитаты, рекламные блоки, архивные записи. Не подходит для блоков, просто размещенных сбоку.

```
<aside>
  <h2>...</h2>
  <p>...</p>
</aside>
```

```
<aside>
  <h2>...</h2>
  <p>...</p>
  <blockquote>
    <p>...<cite>...</cite>...</p>
    <p>...</p>
  </blockquote>
</aside>
```

На рисунке 1 показан способ применения элемента <aside>:

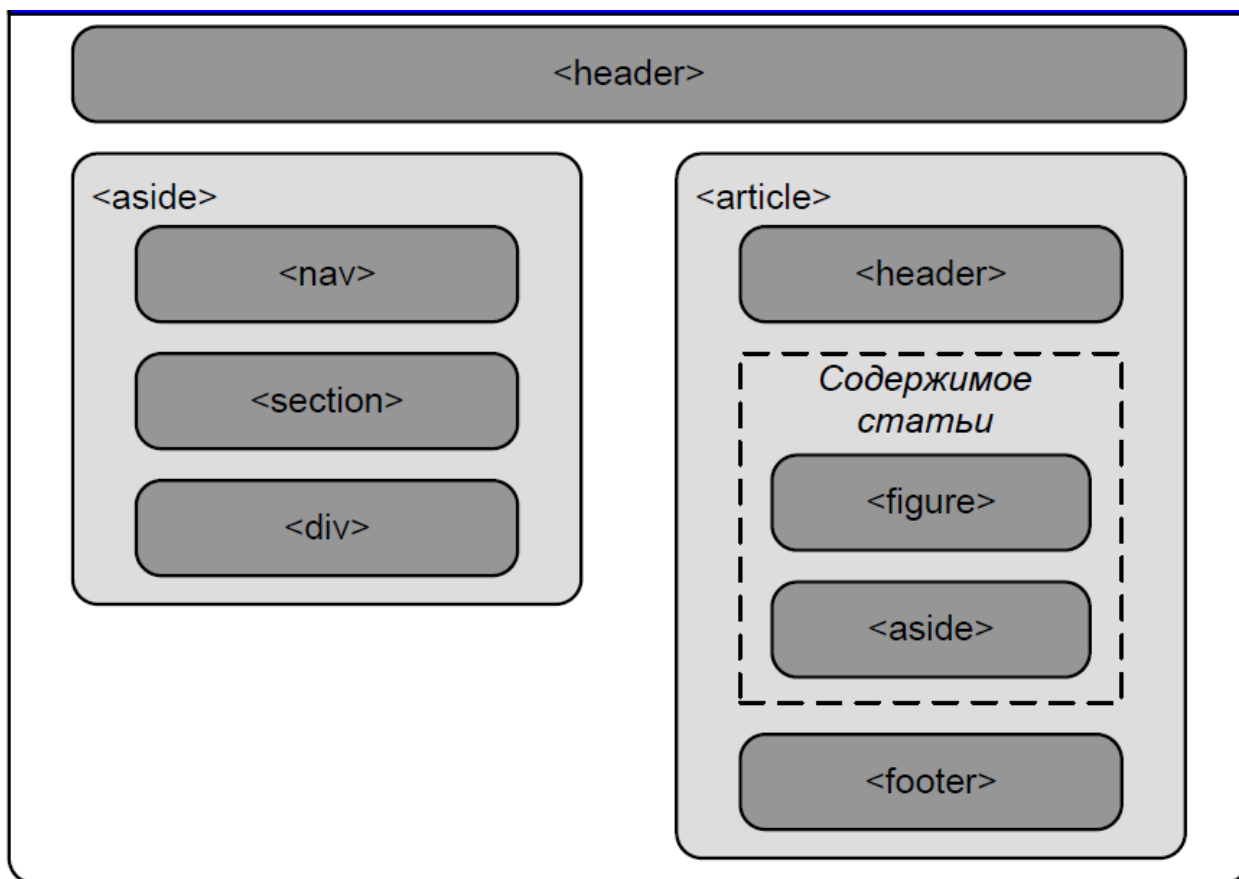


Рисунок 1. Использование элемента <aside>

Элемент <footer>

Представляет собой нижний колонтитул содержащей его секции или корневого элемента. Обычно содержит информацию об авторе статьи, данные о копирайте и т. д. Если используется как колонтитул всей страницы, содержимое дополняется сведениями об авторских правах, ссылками на условия использования, контактную информацию, ссылками на связанное содержимое и т. п.

В одном веб-документе может быть несколько элементов <footer>. Как каждая страница, так и каждая статья может иметь свой элемент <footer>. Более того, <footer> можно поместить в элемент <blockquote>, чтобы указать источник цитирования.

```
<footer>
  <address>...</address>
  <small>©2014...</small>
</footer>
```

Элемент <address>

Используется для определения контактной информации автора/владельца документа или статьи. Для обозначения автора документа тег размещают внутри элемента <body>, для отображения автора статьи – внутри тега <article>. В браузере обычно отображается курсивом.

Элемент <figure>

Рисунок в HTML5 – не то же самое, что изображение. Его следует рассматривать как рисунок в книге – изображение, отдельное от текста, на которое в тексте делаются ссылки. Рисунки размещаются как плавающие объекты, т. е. их вставляют в первое удобное место в тексте, вместо того чтобы закреплять возле конкретного слова или элемента. Часто рисунки снабжены подписями, которые плавают вместе с ними.

Для контейнера рисунка можно использовать элемент <figure>. А текст подписи к рисунку помещается в элемент <figcaption> внутри элемента <figure>:

```
<figure class="FloatFigure">
  
  <figcaption>Данный текст отображается под рисунком</figcaption>
</figure>
```

Элемент <header>

Группирует вводные и навигационные элементы, не является обязательным. Может содержать заголовки, оборачивать содержание раздела страницы, форму поиска или логотипы, группировать основной заголовок или группу заголовков, которые, в свою очередь, могут быть обернуты элементом <hgroup>. В HTML5-документе может содержаться одновременно несколько элементов <header>, и они могут располагаться в любой части страницы.

```
<header>
  <hgroup>
    <h1>...</h1>
    <h2>...</h2>
  </hgroup>
</header>
```

Элемент <header> нельзя помещать внутри элементов <footer>, <address> или другого элемента <header>.

Элемент <hgroup>

Используется для группировки элементов <h1> – <h6> в случае, когда заголовок имеет сложную структуру, например, если есть уточняющие подзаголовки, альтернативные заголовки и т. п.

```
<hgroup>
  <h1>...</h1>
  <h2>...</h2>
</hgroup>
```

Элемент <nav>

Предназначен для создания блока навигации веб-страницы или всего веб-сайта, при этом не обязательно должен находиться внутри <header>. На странице может быть несколько элементов <nav>. Не заменяет теги или , он просто их обрамляет. Не все группы ссылок на странице

должны быть обернуты `<nav>`: этот элемент предназначен в первую очередь для разделов, которые состоят из главных навигационных блоков.

```
<nav>
  <ul>
    <li><a>...</a></li>
    <li><a>...</a></li>
    <li><a>...</a></li>
  </ul>
</nav>
```

В качестве элементов панели навигации можно использовать не только элементы списков:

```
<nav>
  <p><a href="">...</a></p>
  <p><a href="">...</a></p>
</nav>
```

Также можно добавлять заголовки внутрь элемента:

```
<nav>
  <h2>...</h2>
  <ul>
    <li><a>...</a></li>
    <li><a>...</a></li>
    <li><a>...</a></li>
  </ul>
</nav>
```

Элемент `<main>`

Используется для основного содержимого документа. Содержимое должно быть уникальным и не включать типовые блоки, такие как шапка, подвал, навигацию, боковые панели, формы поиска и т. д.

Семантика для текстового содержимого

Элемент `<time>`

Определяет время (24 часа) или дату по григорианскому календарю с возможным указанием времени и смещения часового пояса. Текст, заключенный в данный тег, не имеет стилового оформления браузером. Для тега доступен атрибут `datetime`, в качестве содержимого которого указывается то, что будет видеть пользователь на экране своего компьютера:

```
<time datetime="2014-09-25"> 25 Сентября 2014</time>
```

Чтобы дата могла считываться автоматически, она должна быть в формате YYYY-MM-DD. Время, которое также может указываться, задается в формате HH:MM с добавлением разделяющего префикса T (time):

```
<time datetime="2014-09-25T12:00"> 25 Сентября 2014</time>
```

Элемент <mark>

Текст, помещенный внутри тега <mark>, выделяется по умолчанию желтым цветом (цвет фона и цвет шрифта в выделенном блоке можно изменить, задав определенные CSS-стили). С помощью данного тега можно отмечать важное содержимое, а также ключевые слова.

Элемент <bdi>

Отделяет фрагмент текста, который должен быть изолирован от остального текста для двунаправленного форматирования текста. Используется для текстов, написанных одновременно на языках, читающихся слева направо и справа налево.

Элемент <wbr>

Одиночный тег, показывает браузеру место, где можно добавить разрыв длинной строки в случае необходимости.

Применение HTML5 на практике

Форматирование демонстрационной страницы, которая показана на рисунке 2, задано во внешней таблице стилей.



Рисунок 2. Демонстрационная страница

В традиционной HTML-странице форматирование происходит при помощи таблицы стилей и контейнеров `<div>` и ``. Элемент `` позволяет форматировать отрывки текста внутри другого элемента. А элемент `<div>` – целые блоки содержимого, а ещё он устанавливает общую структуру страницы (рисунок 3).

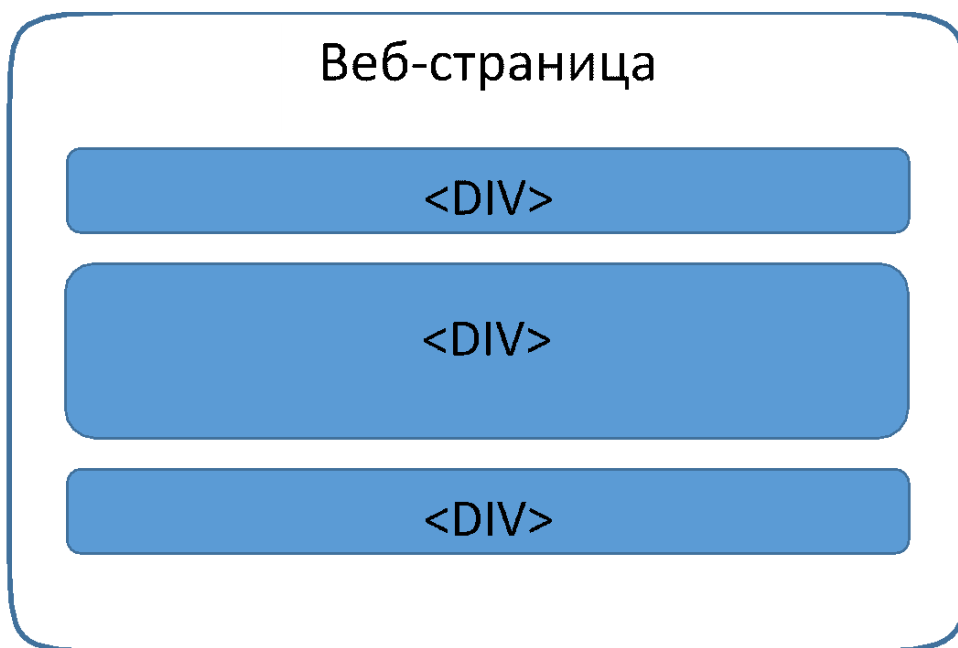


Рисунок 3. Общая структура страницы

Содержимое стандартного HTML-файла приведено ниже:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Cache-Control" content="no-cache">
<meta charset="utf-8">
<link rel="stylesheet" href="css/Base.css">
<title>Первая страница</title>
</head>
<body>
<div id="header">
Блок вывода шапки!!!
</div>
<div id="block">
  <div id="nav">
    <a href="?url=index">Главная</a>
    <a href="?url=dop">Дополнительная информация</a>
    <a href="?url=cont">Контакты</a>
  </div>
</div>
<div id="content">
  <p>Блок вывода контента!!! </p>
  <h1>Урок 1. HTML5: начало</h1>
  <div class="floatfigure">
    
    <p>HTML5 & CSS3</p>
  </div>
  <p>Обзор HTML5, новые семантические элементы HTML5, новый способ
структурирования страниц. Семантика текстового уровня. <br>
```

```

</div>
<div id="footer">Блок вывода нижнего колонтитула!!!
<p>HTML5 & CSS3. Профессиональная веб-разработка. Уровень 2.</p>
<p>Все права защищены. Использование материалов сайта только со ссылкой на
первоисточник.</p>
</div>
</body>
</html>

```

Недостаток элемента `<div>` в том, что он не предоставляет никакой информации о странице. Встретив в разметке элемент `<div>`, вы (или браузер, средство разработки, поисковый робот и т. п.) понимаете, что нашли отдельный блок страницы, но не знаете назначение этого блока.

Чтобы исправить эту ситуацию, в HTML5 некоторые элементы `<div>` можно заменить более описательными семантическими элементами. Эти семантические элементы действуют таким же образом, как и `<div>`-элемент: они группируют часть разметки в блок, но не выполняют никаких собственных операций над содержимым блока; они также предоставляют «стилевую зацепку», позволяющую отформатировать блок. Но кроме этого, они также добавляют в страницу семантический смысл.

Далее рассмотрим разметку страницы, показанной на рисунке 2, но с применением новшеств HTML5. Упрощенное представление HTML5-страницы с применением новых семантических элементов будет выглядеть, как представлено на рисунке 4:

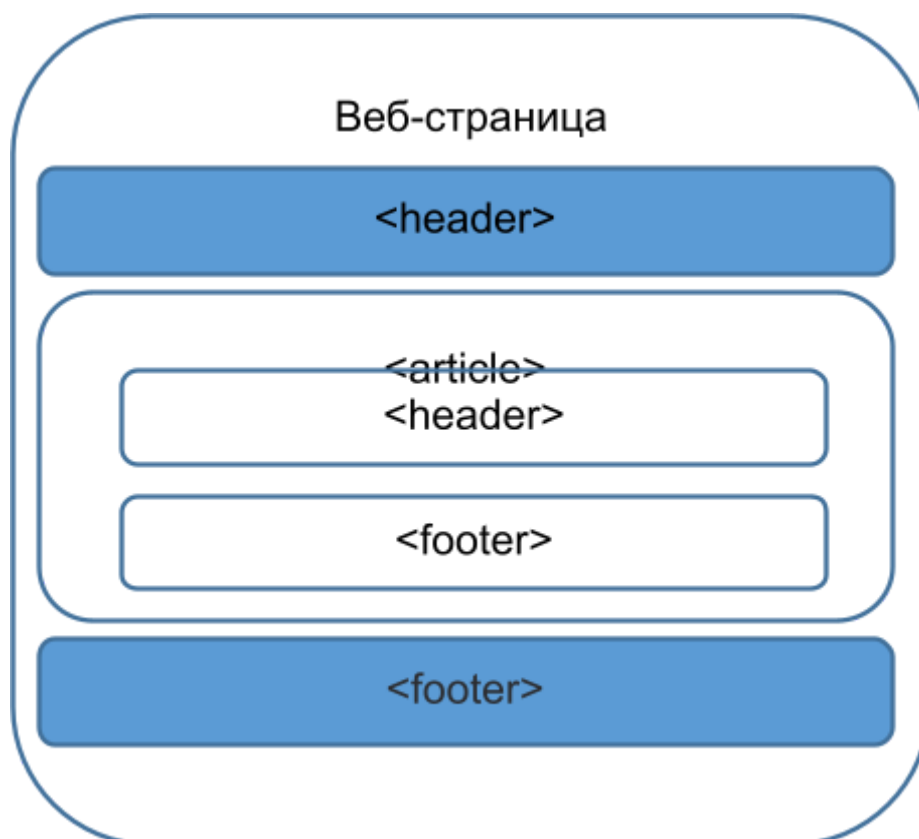


Рисунок 4. Представление HTML-страницы с применением новых семантических элементов

Вместо применяемых ранее элементов `<div>` используются соответствующие контенту элементы HTML5.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Cache-Control" content="no-cache">
<meta charset="utf-8">
<link rel="stylesheet" href="css/Base.css">
<title>Первая страница курса HTML5&CSS</title>
</head>
<body>
<header>
  <div class="floatfigure">
    
  </div>
  <h1>HTML5 & CSS</h1>
</header>

<nav>
  <a href="?url=index">Главная</a>
  <a href="?url=1erer">Обучение</a>
  <a href="?url=dop">Дополнительная информация</a>
  <a href="?url=cont">Контакты</a>
</nav>
<div id="content">
<article>
  <header>
    <small>    <time datetime="2016-06-16">Опубликовано 16 июня 2016</time>
  </small>
  <h1>Урок 1. HTML5: начало</h1>
  </header>
  <figure class="floatfigure">
    
    <figcaption>HTML5 & CSS</figcaption>
  </figure>
  <p>Обзор HTML5, новые семантические элементы HTML5, новый способ
структурирования страниц. Семантика текстового уровня.<br>
</p>
</article>
</div>
<footer>
Блок вывода нижнего колонтитула!!!
<p>HTML5&CSS. Профессиональная веб-разработка. Уровень 2.</p>
<p>Все права защищены. Использование материалов сайта только со ссылкой на
первоисточник.</p>
</footer>
</body>
</html>
```

Материалы для домашнего изучения

Изучить понятие «Микроформат»:

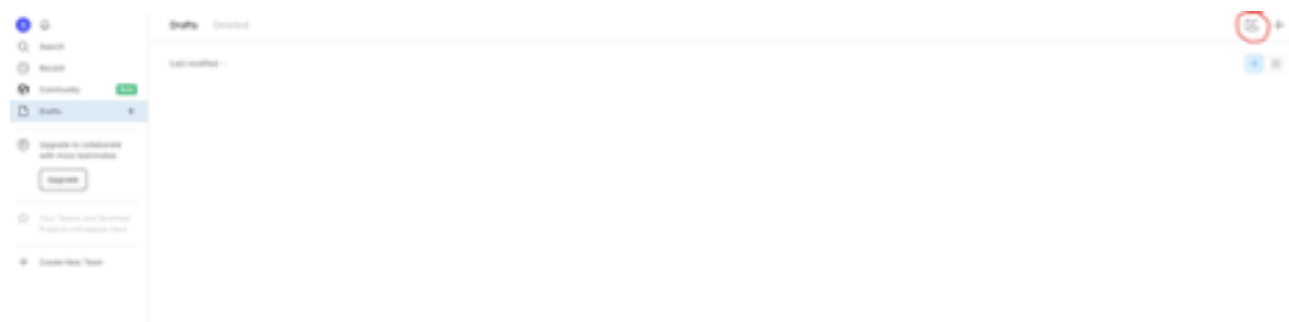
1. Обозначение контактной информации (микроформат hCard).
2. Обозначение событий (микроформат hCalendar).

Инструкция по работе с Figma

Чтобы загрузить файл .fig, нужно зайти либо в приложение figma, либо на веб-интерфейс <https://www.figma.com/>. Далее нужно попасть в личный кабинет с проектами. Если вы пользуетесь figma в первый раз, то нужно зарегистрироваться.



Далее нажимаем кнопку «Sign up», вводим почту и пароль и нажимаем «Create account». После создания, заходим в кабинет через кнопку «Log in»



В кабинете нажимаем на иконку стрелочки(Import) и выбираем наш .fig файл (ссылка на него указана в практическом задании). Далее открываем проект.



Нажимаем кнопку Inspect и теперь можно смотреть свойства элементов и их отступы относительно других объектов.

Практическое задание

Основываясь на изученном материале, создать HTML5-документ следующего содержания:

1. [Макет страницы](#) (настоящий верстальщик должен иметь в портфолио несколько сверстанных макетов, поэтому будем их создавать). Не самый сложный, в то же время есть на чем руку набить, четко видна семантическая структура.
2. Добавить необходимые теги HTML5, которые изучили на уроке.
3. Домашнее задание рассчитано на 2 урока, не стоит переживать, если не успели все сделать.

Дополнительные материалы

1. <https://dcrazed.com/category/graphics/psd/>.
2. <http://psd-html-css.ru/shablony/besplatnye-psd-makety> – можно посмотреть по тематикам.
3. <https://all-psd.ru/site-templates/> – один из самых распространенных, но макеты немного устарели.
4. <http://freepsd.ws/>.
5. <http://beloweb.ru/shablonyi/25-besplatnyih-html-shablonov-v-stile-flat.html> – самые простые, но в тот же момент чтобы быть в курсе новых тенденций (рекомендую).

Можно выбрать любой другой макет.

Используемая литература

1. <http://www.wisdomweb.ru/>
2. <http://html5book.ru/>

3. Гоше Х. HTML5. Для профессионалов. СПб.: Питер, 2013.
4. Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. СПб.: Питер, 2012.
5. Макфарланд Д. Большая книга CSS3. СПб.: Питер, 2014.