

---

# HTML



## Часто используемые сокращения

- **API:** Application programming interface (Интерфейс прикладного программирования, API-интерфейс)
- **CSS3:** Cascading style sheet version 3 (Каскадная таблица стилей), версия 3
- **GUI:** Graphical user interface (Графический интерфейс пользователя)
- **HTML:** Hypertext Markup Language (Язык гипертекстовой разметки, язык HTML)
- **HTML5:** HTML, версия 5
- **SQL:** Structured Query Language (Язык структурированных запросов)
- **UI:** User interface (Интерфейс пользователя)

HTML5 — это инструмент для упорядочивания Web-контента. Он предназначен для упрощения Web-проектирования и Web-разработки за счет языка разметки, обеспечивающего стандартизированный и интуитивно понятный пользовательский интерфейс. HTML5 предоставляет разработчику средства для секционирования и структуризации Web-страниц, а также позволяет создавать обособленные компоненты, которые не только обеспечивают логическую организацию Web-сайта, но и предоставляют ему возможности синдикации. Язык HTML5 реализует подход к проектированию Web-сайтов, основанный на отображении информации, поскольку он воплощает саму суть отображения информации — разделение и маркирование информации для упрощения ее использования и понимания. Именно в этом состоит огромная семантическая и эстетическая ценность HTML5. HTML5 предоставляет дизайнерам и разработчикам всех уровней возможности для предоставления в публичный доступ буквально любого контента – от простых текстов до мультимедийно насыщенных интерактивных материалов.

браузерных Web-приложений, а также приложений для мобильных устройств. HTML5 относится к числу технологий, которые стимулируют развитие мобильных сервисов на основе облачных вычислений. Кроме того, HTML5 способствует повышению гибкости – благодаря возможности создания впечатляющих и интерактивных Web-сайтов. И, наконец, HTML5 предлагает новые теги и усовершенствования, в числе которых следующие: элегантная структура, органы управления формами, API-интерфейсы, мультимедийные функции, поддержка баз данных, существенно увеличенная скорость обработки.

Новые теги HTML5 обладают «говорящими» названиями, которые раскрывают назначение и характер использования этих элементов. В предыдущих версиях HTML использовались весьма неопределенные названия тегов. В спецификации HTML5, напротив, используются весьма описательные, интуитивно понятные названия. HTML5 предоставляет информационно-насыщенные названия, которые однозначно идентифицируют соответствующий контент. Например, широко применявшийся до настоящего времени тег `<div>` был дополнен тегами `<section>` и `<article>`. Кроме того, были добавлены теги `<video>`, `<audio>`, `<canvas>` и `<figure>`, которые обеспечивают более точное описание определенных типов контента.

HTML5 предоставляет следующие возможности.

- Теги с описательными названиями, которые точно указывают, для содержания какого контента предназначены эти теги.
- Усовершенствованные сетевые коммуникации.
- Существенно улучшенное хранение данных
- Средства Web Worker для исполнения фоновых процессов.
- Интерфейс WebSocket для установки постоянного соединения между резидентным приложением и сервером.
- Улучшенное извлечение хранящихся данных.
- Повышенная скорость сохранения и загрузки страниц
- Поддержка CSS3 при управлении пользовательским интерфейсом, что обеспечивает контентную ориентированность HTML5.
- Улучшенная обработка форм в браузере.
- API-интерфейс баз данных на основе SQL, позволяющий применять локальное хранилище на стороне клиента.
- Теги `canvas` и `video`, позволяющие добавлять графические и видеоматериалы без установки сторонних подключаемых модулей.
- Спецификация API-интерфейса Geolocation, использующая геолокационные возможности смартфонов в интересах задействования облачных сервисов и приложений для мобильных устройств.
- Усовершенствованные формы, ослабляющие потребность в загрузке кода JavaScript, что обеспечивает более эффективную связь между мобильными устройствами и серверами cloud-среды.

HTML5 позволяет предоставить пользователю более впечатляющие возможности: страницы, спроектированные с использованием спецификации HTML5, способны предоставлять такие же возможности, как приложения для настольных систем. Кроме того, HTML5 существенно

улучшает разработку для нескольких платформ благодаря сочетанию возможностей API-интерфейсов с повсеместностью браузера. HTML5 позволяет разработчикам предоставлять возможности современных приложений, беспрепятственно охватывающие несколько платформ.

Фактически HTML5 является синонимом непрерывных инноваций: новые теги, новые методики и общая инфраструктура разработки, базирующаяся на взаимодействии технологии HTML5 с родственными технологиями CSS3 и JavaScript. Это создает основу для функционирования приложений, ориентированных на клиентов. Помимо широкого распространения средств и методик технологии HTML5 на настольных системах, она может быть реализована в функционально насыщенных Web-браузерах для мобильных телефонов. Это растущий рынок, характерными представителями которого являются популярные и вездесущие платформы Apple iPhone, Google Android и Palm webOS.

Важнейшим аспектом мощных возможностей HTML5 является препарирование информации — или разделение контента на блоки, которое делает процесс гораздо понятнее. Высокая эффективность этого инструмента при проектировании и разработке подтверждается его усиливающимся доминированием в сфере Web-обработки.

HTML5 знаменует приход более эффективного семантического процесса на текстовом уровне и преобладание контролируемости над конструированием и использованием форм. Все эти и многие другие инновационные аспекты HTML5 обуславливают усиливающееся доминирование этой новой парадигмы. Этот нарастающий эффект до той или иной степени повлиял на многие организации (причем не только на коммерческие), в том числе на многие организации, у которых обработка информации и коммуницирование лишь недавно вошли в число основных видов деятельности.

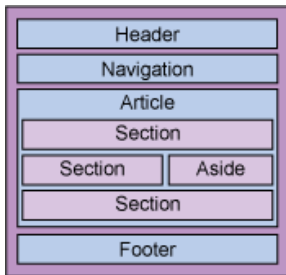
Технология HTML5 — это не волшебная лампа с джином внутри. Тем не менее технические и методические активы этой технологии уже сделали ее вторым по значимости способом решения проблем (первый способ — потерять волшебную лампу).

## Планирование страницы

Мы собираемся создать простую Web-страницу. В ходе этого процесса рассмотрим несколько новых тегов, впервые появившихся в спецификации HTML5. Для создания эффективной и рациональной Web-страницы необходимо разработать план, охватывающий все создаваемые компоненты.

Создаваемая нами страница будет иметь высокоуровневый дизайн, показанный на рисунке 1. Страница состоит из области Header, области Navigation, области Article, содержащей три раздела, области Aside и, наконец, области Footer. Эта страница предназначена для работы в браузере Google Chrome, что исключает возможный визуальный беспорядок, порождаемый стремлением к браузерной совместимости, который мог бы затруднить понимание базовой структуры. Наша цель состоит в создании страницы, которая наглядно демонстрировала бы новые теги HTML5 и показывала возможности их использования для создания хорошо оформленного кода и элегантного дизайна страницы.

## Рисунок 1. План Web-страницы Acme United



В процессе создания этой страницы я коснусь технологии CSS3, которая требуется для надлежащего отображения HTML5-страниц. Технология CSS3 необходима для придания определенного стиля HTML5-странице, для навигации по этой странице и для создания общего впечатления об этой странице. Группы свойств, описываемые в разделе CSS3 Reference на Web-сайте W3Schools.com (см. раздел [Ресурсы](#)), охватывают такие полезные теги, как фон, шрифт, выделенные области и анимация.

Однако прежде чем приступать к конструированию страницы, необходимо изучить несколько новых элементов HTML5.

### Область Header

В рассматриваемом примере область Header содержит заголовок страницы и подзаголовок. Тег `<header>` используется с целью создания контента для области header данной страницы. Тег `<header>` может содержать открывающую информацию о теге `<section>` и о теге `<article>` в дополнение к самой Web-странице. Создаваемая нами Web-страница имеет область header, показанную на высокоуровневом представлении ее дизайна, а также область header внутри области Article и области Section. В листинге 1 показан пример разметки тега `<header>`.

### Листинг 1. Пример тега `<header>`

```
<header>
  <h1>Heading Text</h1>
  <p> Text or images can be included here</p>
  <p> Logos are frequently placed here too</p>
</header>
```

Тег `<header>` также может содержать тег `<hgroup>` (листинг 2). Тег `<hgroup>` группирует заголовки вместе, используя показанные уровни заголовков с `<h1>` по `<h6>` с главным заголовком и подзаголовком.

### Листинг 2. Пример тега `<hgroup>`

```
<header>
  <hgroup>
    <h1>Main Heading</h1>
    <h2>Sub-heading </h2>
  </hgroup>
  <p> Text or images can be included here</p>
</header>
```

## Область Navigation

Область Navigation на данной странице создается с помощью тега `<nav>`. Тег `<nav>` специфицирует область, специально предназначенную для навигации. Тег `<nav>` следует использовать для навигации по основному сайту, а не для хранения ссылок на другие области данной страницы. Область Navigation может содержать код, подобный показанному в листинге 3.

### Листинг 3. Пример тега `<nav>`

```
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About Us</a></li>
    <li><a href="#">Our Products</a></li>
    <li><a href="#">Contact Us</a></li>
  </ul>
</nav>
```

## Области Article и Section

Проектируемая нами страница имеет одну область Article, которая содержит реальный контент данной страницы. Для создания этой области мы используем тег `<article>`, определяющий контент, который может быть использован независимо от остального контента этой данной страницы. Например, если вы хотите создать RSS-поток, то сможете использовать тег `<article>` для уникальной идентификации соответствующего контента. Тег `<article>` идентифицирует контент, который при перемещении в другой контекст может стать совершенно недоступным для понимания.

Область Article в плане страницы Acme United содержит три области Section. Мы создаем эти области с помощью тега `<section>`. Тег `<section>` содержит релевантные области компонентов Web-контента. Тег `<section>`— как и тег `<article>`— способен содержать заголовки, нижние колонтитулы или любые другие компоненты, необходимые для формирования данного раздела. Тег `<section>` предназначен для группировки контента. Контент для тега `<section>` и для тега `<article>` обычно начинается с тега `<header>` и заканчивается тегом `<footer>`, между которыми находится сам контент соответствующего тега.

Тег `<section>` также способен содержать теги `<article>`, а тег `<article>` способен содержать тег `<section>`. Тег `<section>` следует использовать для группировки сходной информации, а тег `<article>` следует использовать для такой информации, как статья или блог, которая может быть перемещена в новый контекст без искажения смысла контента. Тег `<article>`, как и следует из его названия («статья»), предоставляет полный пакет информации. Тег `<section>`, напротив, содержит связанную информацию, однако эта информация не может быть помещена в другой контекст сама по себе, поскольку в этом случае ее смысл будет потерян.

В листинге 4 показан пример применения тега `<article>` и тега `<section>`.

## Листинг 4. Пример тегов <article> и <section>

```
<article>
  <section>
    Content
  </section>
  <section>
    Content
  </section>
</article>
<section>
  <article>
    Content
  </article>
  <article>
    Content
  </article>
</section>
```

## Изобразительные теги

Теги <section> и <article>, как и теги <header> и <footer>, способны содержать тег <figure>. Этот тег используется для включения изображений, диаграмм и фотографий.

Тег <figure> может содержать тег <figcaption>, который, в свою очередь, содержит подпись для иллюстрации, содержащейся в теге <figure>. Это позволяет ввести описание, которое будет теснее связывать эту иллюстрацию с контентом. В листинге 5 показан пример структуры с тегом <figure> и тегом <figcaption>.

## Листинг 5. Пример тега <figure> и тега <figcaption>

```
<figure>
  
  <figcaption>Caption for the figure</figcaption>
</figure>
```

## Медиа-теги

Теги <section> и <article> также способны содержать различный медиаконтент. HTML5 предоставляет теги, которые обеспечивают быстрое понимание содержащегося в них контента. До недавнего времени такие медийные компоненты, как музыка и видео, поддерживались только встраиваемыми средствами. HTML5 поддерживает их собственными средствами.

Тег <audio> способен содержать аудиоконтент, например, музыкальные или любые другие звуковые потоки. Тег <audio> имеет атрибуты, которые определяют, какой аудиоматериал, когда и как будет воспроизводиться. К числу этих атрибутов относятся следующие: `src`, `preload`, `control`, `loop`, `autoplay`. В примере, показанном в листинге 6, аудио начинает воспроизводиться немедленно после загрузки страницы и воспроизводится непрерывно. Пользователю предоставляются органы управления, с помощью которых он может остановить или перезапустить воспроизведение аудио.

## Листинг 6. Пример тега <audio>

```
<audio src="MyFirstMusic.ogg" controls autoplay loop>
  Your browser does not support the audio tag.
</audio>
```

Тег `<video>` позволяет транслировать видеоклипы или потоковые визуальные материалы. Он имеет все атрибуты тега `<audio>`, а также три других атрибута: `poster`, `width` и `height`. Тег `poster` позволяет указать изображение, которое должно отображаться на протяжении загрузки видео, или в ситуации, когда видео вообще не способно загрузиться.

В листинге 7 показан пример структуры тега `<video>`.

### Листинг 7. Пример тега `<video>`

```
<video src="MyFirstMovie.ogg" controls="controls">
  Your browser does not support the video tag
</video>
```

Теги `<video>` и `<audio>` способны содержать тег `<Source>`, который описывает мультимедийные ресурсы для тегов `<video>` и `<audio>`. Этот тег позволяет разработчику указать альтернативные видео- и аудиофайлы, из которых браузер сможет затем выбирать в зависимости от типа поддерживаемого медиаформата или кодека. В листинге 8 представлено два варианта выбора. Если используемый браузер не способен воспроизвести WMA-версию файла, попытайтесь использовать версию в формате MP3. В противном случае отобразите соответствующее сообщение, чтобы пользователь понимал, что данный аудиофайл недоступен.

### Листинг 8. Пример тега `<source>`

```
<audio>
  <source src="/music/good_enough.wma" type="audio/x-ms-wma">
  <source src="/music/good_enough.mp3" type="audio/mpeg">
  <p>Your browser does not support the HTML 'audio' element.</p>
</audio>
```

Тег `<embed>` описывает встроенный контент, который может быть отображен на странице, например, файлы Adobe Flash в формате SWF. Листинг 9 содержит атрибут `type`, который идентифицирует встроенный источник как Flash-файл.

### Листинг 9. Пример тега `<embed>`

```
<embed src="MyFirstVideo.swf" type="application/x-shockwave-flash" />
```

В дополнение к атрибутам `src` и `type` тег `<embed>` имеет атрибут `height` и атрибут `width`.

## Область Aside

Мы создаем область Aside в плане страницы Acme United с помощью тега `<aside>`. Этот тег предназначен для вспомогательного контента, не являющегося частью основного материала статьи, которую он дополняет. Например, в журналах теги типа `aside` часто используются для выделения выводов, сделанных в самой статье. Тег `<aside>` содержит контент, который может быть удален без ущерба для информации, излагаемой в статье, секции или странице, содержащей этот тег.

В листинге 10 показан пример применения тега `<aside>`.

## Листинг 10. Пример тега <aside>

```
<p>My family and I visited Euro Disney last year.</p>
<aside>
  <h4>Disney in France</h4>
  <p>Besides Euro Disney, there is a Disneyland in California.</p>
</aside>
```

## Область Footer

Тег <footer> содержит информацию о странице, о статье или о разделе, например, сведения об авторе или дату написания. В качестве нижнего колонтитула он может содержать информацию об авторских правах и другую важную юридическую информацию (листинг 11).

## Листинг 11. Пример тега <footer>

```
<footer>
  <p>Copyright 2011 Acme United. All rights reserved.</p>
</footer>
```

## Конструирование страницы

Теперь, когда вы познакомились с базовыми тегам, необходимыми для создания HTML5-страницы, приступим к конструированию своей страницы. Мы собираемся сконструировать Web-страницу для компании Acme United. Полностью эта страница показана на рисунке 2. Вы можете загрузить ее для последующего использования (см. раздел [Загрузка](#)).

## Рисунок 2. Web-страница Acme United



Итак, приступим к формированию страницы. Сначала следует тег <!doctype>. В спецификации HTML5 тег <!doctype> был упрощен: вам достаточно запомнить его атрибут —html. Это не только облегчает ввод этого тега, но и улучшает его защиту от ошибок. Обратите внимание, что атрибут имеет вид html, а не html5. Вне зависимости от количества версий HTML, тег <!doctype> всегда сможет иметь атрибут html.



Тег `<html>` содержит все остальные HTML-теги за исключением тега `<!doctype>`. Каждый из остальных тегов должен быть размещен между тегом `<html>` и тегом `</html>` (листинг 12).

## Листинг 12. Пример тега `<!doctype>`

```
<!doctype html>
<html lang="en">
```

После указания атрибута `html` и английского языка следует тег `<head>`, который может содержать скрипты, информацию о поддерживаемых браузерах, ссылки на таблицу стилей, метаинформацию и другие инициализационные функции. В разделе `head` можно использовать следующие теги.

- `<base>`
- `<link>`
- `<meta>`
- `<script>`
- `<style>`
- `<title>`

Тег `<title>` содержит фактический заголовок документа и является обязательным тегом раздела `<head>`. Этот заголовок можно увидеть в верхней части браузера при просмотре страницы. Тег `<link>` в листинге 13 указывает CSS3-таблицу стилей, которая будет использоваться для отображения данной HTML5-страницы. Эта таблица стилей имеет имя `main-stylesheet.css`.

## Листинг 13. Пример тега `<head>`

```
<head>
  <title>HTML5 Fundamentals Example</title>
  <link rel="stylesheet" href="main-stylesheet.css" />
</head>
```

Затем мы используем тег `<body>`, за которым следуют теги `<header>` и `<hgroup>`, описанные ранее. Область `<h1>` в данном примере содержит название вымышленной компании (Acme United), а область `<h2>` содержит подзаголовок «A Simple HTML5 Example» (простой пример на HTML5). Соответствующая разметка показана в листинге 14.

## Листинг 14. Пример тегов `<body>` и `<header>`

```
<body >
  <header>
    <hgroup>
      <h1>Acme United</h1>
      <h2>A Simple HTML5 Example</h2>
    </hgroup>
  </header>
```

Технология CSS3 используется для формирования облика страницы, как показано в листинге 15. Сначала для страницы задается шрифт, а затем детали для тела страницы. После этого задаются размеры тела, а затем проектируется структура параграфа

заголовка для тегов заголовков первого и второго уровней. Именно эти заголовки мы будем использовать для данной страницы.

## Листинг 15. Первый пример CSS3

```
* {
  font-family: Lucida Sans, Arial, Helvetica, sans-serif;
}

body {
  width: 800px;
  margin: 0em auto;
}

header h1 {
  font-size: 50px;
  margin: 0px;
  color: #006;
}

header h2 {
  font-size: 15px;
  margin: 0px;
  color: #99f;
  font-style: italic;
}
```

В листинге 16 показан тег `<nav>`, предназначенный для осуществления навигации по главному сайту.

## Листинг 16. Пример тега `<nav>`

```
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About Us</a></li>
    <li><a href="#">Contact Us</a></li>
  </ul>
</nav>
```

В HTML5 также имеется тег `<menu>`, который вызывает затруднения у некоторых дизайнеров и разработчиков. Это объясняется тем фактом, что под навигацией часто понимается «навигационное меню». Тег `<menu>`, который был исключен в версии HTML 4.01, а затем снова появился в HTML5, предназначен для улучшения интерактивности. Этот тег не следует использовать для основной навигации. Единственный тег, который следует использовать для основной навигации — это тег `<nav>`. Мы используем тег `<menu>` в нашем примере позднее.

Форматирование навигации осуществляется посредством CSS3. Каждое определение тега `<nav>` в листинге 17 отражает определенное состояние тегов `<ul>` и `<li>` внутри тега `<nav>`.

## Листинг 17. Второй пример CSS3

```
nav ul {
  list-style: none;
  padding: 0px;
  display: block;
```

```

clear: right;
background-color: #99f;
padding-left: 4px;
height: 24px;
}
nav ul li {
display: inline;
padding: 0px 20px 5px 10px;
height: 24px;
border-right: 1px solid #ccc;
}

nav ul li a {
color: #006;
text-decoration: none;
font-size: 13px;
font-weight: bold;
}

nav ul li a:hover {
color: #fff;
}

```

Далее следует область Article. Эта область, определяемая тегом `<article>`, содержит свой собственный тег `<header>`. Тег `<section>` внутри тега `<article>` также содержит свой собственный тег `<header>` (листинг 18).

## Листинг 18. Пример тегов `<article>` и `<section>`

```

<article>
  <header>
    <h1>
      <a href="#" title="Link to this post" rel="bookmark">Article Heading</a>
    </h1>
  </header>
  <p> Primum non nocere ad vitam Paramus . . . </p>
  <section>
    <header>
      <h1>This is the first section heading</h1>
    </header>
    <p>Scientia potentia est qua nocent docentp . . .</p>
  </section>

```

В листинге 19 показана CSS3-разметка, обеспечивающая визуализацию этого формата. Обратите внимание, что области `paragraph`, `header` и `section` определяются тегом `<article>`, внутри которого они содержатся. Заданный здесь тег `<h1>` не совпадает по формату с тегом `<h1>`, заданным для уровня страницы.

## Листинг 19. Третий пример CSS3

```

article > header h1 {
font-size: 40px;
float: left;
margin-left: 14px;
}

article > header h1 a {
color: #000090;
text-decoration: none;
}

```

```

article > section header h1 {
  font-size: 20px;
  margin-left: 25px;
}

article p {
  clear: both;
  margin-top: 0px;
  margin-left: 50px;
}

```

Второй тег `<section>` в теге `<article>` содержит ту же базовую информацию, что и первый тег `<section>`, однако на этот раз мы собираемся использовать теги `<aside>`, `<figure>`, `<menu>`, и `<mark>` (листинг 20).

Тег `<aside>` используется здесь для представления информации, которая не является частью окружающего ее потока. Тег `<figure>` содержит изображение Стоунхенджа. Данный тег `<section>` также содержит тег `<menu>`, который мы используем для создания кнопок с именами четырех Муз. При нажатии на какую-либо из этих кнопок этот тег предоставляет информацию о соответствующей Музе. Тег `<mark>` внутри тега `<p>` используется для подсвечивания слов *veni, vidi, vici*.

## Листинг 20. Пример тегов `<article>` и `<section>`

```

<section>
  <header>
    <h1>Second section with mark, aside, menu & figure</h1>
  </header>
  <p class="next-to-aside"> . . . <mark>veni, vidi, vici</mark>. Mater . . .</p>
  <aside>
    <p>This is an aside that has multiple lines. . . .</p>
  </aside>
  <menu label="File">
    <button type="button" onClick="JavaScript:alert('Clio . . .')">Clio</button>
    <button type="button" onClick="JavaScript:alert('Thalia . . .')">Thalia</button>
    <button type="button" onClick="JavaScript:alert
      ('Urania . . .')">Urania</button>
    <button type="button" onClick="JavaScript:alert
      ('Calliope . . .')">Calliope</button>
  </menu>
  <figure>
    <figcaption>Figure 1. Stonehenge</figcaption>
  </figure>
</section>

```

CSS3-разметка для данного раздела содержит новое определение тега `<p>`, который имеет уменьшенную ширину по сравнению с заданной нами шириной страницы. Это изменение позволяет боковой панели (aside) смещаться вправо без перекрытия текста. Соответствующая разметка показана в листинге 21.

## Листинг 21. Четвертый пример CSS3

```

article p.next-to-aside {
  width: 500px;
}

article > section figure {
  margin-left: 180px;
}

```

```

margin-bottom: 30px;
}

article > section > menu {
margin-left: 120px;
}

aside p {
position: relative;
left: 0px;
top: -100px;
z-index: 1;
width: 200px;
float: right;
font-style: italic;
color: #99f;
}

```

## Теги раздела Video

Заключительным компонентом тега `<article>` является раздел `video`. В данном примере (листинг 22) видео представляет собой видеофайл в формате `ogg`, который автоматически начинает воспроизводиться при загрузке страницы и воспроизводится в бесконечном цикле. Пользователю предоставляются органы управления для приостановки и возобновления воспроизведения. Во многих современных реализациях видеофайлы в формате `ogg` имеют расширение `ogv` («v» означает видео). Тег `<audio>` функционирует аналогичным образом.

### Листинг 22. Пример тегов `<article>` и `<section>`

```

<section>
  <header>
    <h1>This is a video section</h1>
  </header>
  <p><video src="http://people.xiph.org/~maikmerten/demos/BigBuckBunny.ogv"
    controls autoplay loop>
    <div class="no-html5-video"><p>This video will work in
      Mozilla Firefox or Google Chrome only. </p>
    </div>
  </video></p>
</section>
</article>

```

В листинге 23 приведены CSS3-определения для раздела `video`.

### Листинг 23. Пятый пример CSS3

```

article > section video {
height: 200px;
margin-left: 180px;
}

article > section div.no-html5-video{
height: 20px;
text-align: center;
color: #000090;
font-size: 13px;
font-style: italic;
font-weight: bold ;
background-color: #99f;
}

```

В листинге 24 показан нижний колонтитул и завершающая часть страницы.

### Листинг 24. Пример тега <footer>

```
<footer>
  <p>Copyright: 2011 Acme United. All rights reserved.</p>
</footer>
</body>
</html>
```

В листинге 25 показан CSS3-код для нижнего колонтитула.

### Листинг 25. Пятый пример CSS3

```
footer p {
  text-align: center;
  font-size: 12px;
  color: #888;
  margin-top: 24px;
}
```

## Заключение

Итак, мы создали нашу Web-страницу. На этом завершается первая статья в данной серии из четырех частей. Цель этой статьи состояла в том, чтобы представить новый режим работы HTML5. HTML5 — это не просто модернизация HTML4, — это новый способ цифрового коммуницирования. Благодаря функциональности технологий CSS3 и JavaScript, спецификация HTML5 вплотную приблизилась к тому, чтобы предоставить разработчику все возможности в одном пакете. Если вы усвоите все необходимое из обширного массива имеющейся в открытом доступе информации по HTML5, то сможете присоединиться к растущему сообществу Web-дизайнеров и Web-разработчиков, компетентных в мультимедийных аспектах HTML5. В следующей части данной серии мы рассмотрим, как писать код для HTML5-форм и как структурировать эти формы.

---