# Избегаем популярных ошибок в HTML5

## Не используйте **<section>** как обёртку для оформления

Одна из самых распространённых проблем, которую я часто вижу в разметке сайтов — это произвольная замена элементов **<div>** структурными элементами из HTML5, особенно замена оформительской обёртки на **<section>**. В XHTML или HTML4 я бы увидел что-нибудь такое:

<!-- Не копируйте этот код! Он неправильный! -->

<div id="wrapper">

<div id="header">

<h1>Моя супер-пупер страница</h1>

<!-- Содержимое шапки -->

</div>

<div id="main">

<!-- Содержимое страницы -->

</div>

<div id="secondary">

<!-- Вторичное содержимое -->

</div>

<div id="footer">

<!-- Содержимое подвала -->

</div>

</div>

Вместо этого я вижу следующее:

<!-- Не копируйте этот код! Он неправильный! -->

<section id="wrapper">

<header>

<h1>Моя супер-пупер страница</h1>

<!-- Содержимое шапки -->

</header>

<div id="main">

<!-- Содержимое страницы -->

</div>

<section id="secondary">

<!-- Вторичное содержимое -->

</section>

<footer>

<!-- Содержимое подвала -->

</footer>

</section>

Честно говоря, это неправильно: **<section>** — это не обёртка. Элемент **<section>** определяет смысловую секцию содержимого для создания структуры документа. Он должен содержать заголовок. Если вы ищете элемент для того чтобы обернуть в него всю страницу (в стиле HTML или XHTML), подумайте, не применить ли стили непосредственно к элементу **<body>**, как описано у Крока Кеймена. Если же вам всё ещё нужна дополнительная обёртка, используйте **<div>.**

Раз уж Доктор Майк объясняет, что **<div>** не мёртв, а вам не удаётся найти ничего более удачного, пожалуй, этот элемент будет самым подходящим для создания оформительской обёртки.

Таким образом, корректной разметкой для упомянутого выше примера с использованием HTML5 и пары ролей ARIA будет следующий код. Обратите внимание, что вам, в зависимости от дизайна, всё ещё могут понадобится экстра-элементы **<div>**.

<body>

<header>

<h1>Моя супер-пупер страница</h1>

<!-- Содержимое шапки -->

</header>

<div role="main">

<!-- Содержимое страницы -->

</div>

<aside role="complementary">

<!-- Вторичное содержимое -->

</aside>

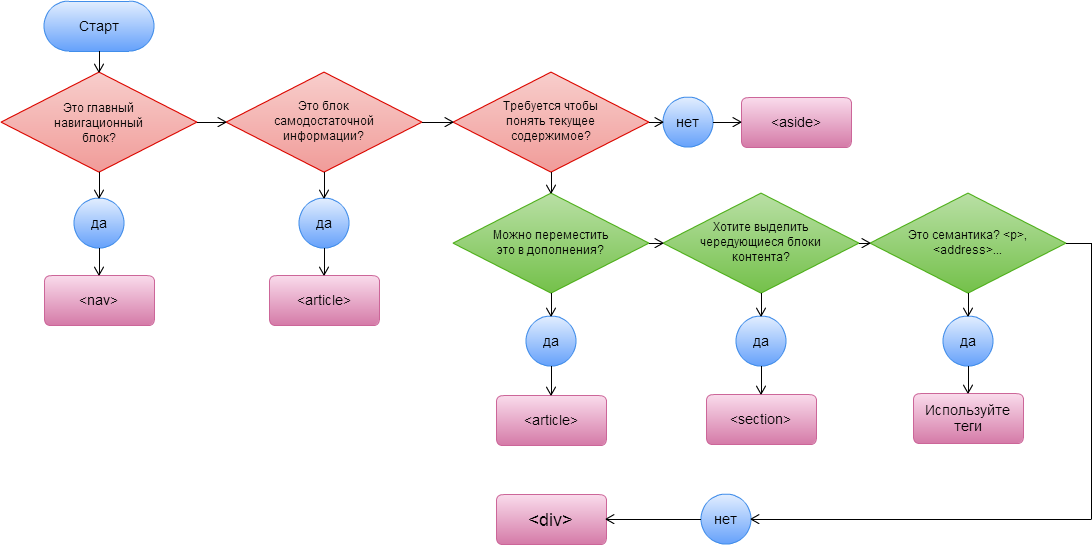
<footer>

<!-- Содержимое подвала -->

</footer>

</body>

Если вы не уверены, какие элементы использовать, я рекомендую вам обратиться к нашей пошаговой схеме выбора HTML5-элементов для разметки содержимого.



## Используйте **<header>** осознанно

Нет смысла писать разметку, если этого можно не делать, так ведь? К сожалению, я часто вижу элемент **<header>** там, где они совсем не нужны.

Элемент **<header>** представляет собой группу вводного содержимого или навигационных средств и обычно содержит структурный заголовок.

### Злоупотребление **<header>**

Думаю, что вы в курсе, что **<header>** можно использовать в документе несколько раз. Но эта возможность привела к следующим ошибкам:

<!-- Не копируйте этот код! Он неправильный! -->

<article>

<header>

<h1>Мой лучший пост</h1>

</header>

<!-- Содержимое записи -->

</article>

Если ваш **<header>** содержит единственный заголовок, избавьтесь от ненужного **<header>**. Элемент **<article>** в любом случае гарантирует, что заголовок войдёт в смысловую структуру документа. И поскольку **<header>** не содержит нескольких элементов, как указано в его описании, зачем вам код, который, в общем-то, не нужен? Будьте проще:

<article>

<h1>Мой лучший пост</h1>

<!-- Содержимое записи -->

</article>

## Не оборачивайте все списки ссылок в **<nav>**

На момент написания статьи существует более 30-ти новых элементов и неудивительно, что у нас разбегаются глаза, когда дело доходит до создания осмысленной структурной разметки. Поэтому не стоит злоупортреблять всеми доступными сейчас суперсемантическими элементами. Что, к сожалению, часто происходит с элементом **<nav>**. Спецификация определяет роль **<nav>** следующим образом:

«Элемент <nav> представляет собой часть страницы, которая ссылается на другие страницы или части текущей, то есть раздел с навигационными ссылками.

Замечание: не все группы ссылок на странице должны быть обёрнуты в элемент <nav> — этот элемент, главным образом, предназначен для группировки главных навигационных блоков. В частности, в подвалах часто содержатся короткие списки ссылок на различные части сайта, вроде правил использования сервиса, домашней страницы и копирайтов. Элемента <footer> вполне достаточно для группировки такого рода ссылок; и несмотря на то, что элемент <nav> может быть использован в таких случаях, обычно в этом нет необходимости.»

Спецификация WHATWG HTML

Ключевая фраза — «главных навигационных блоков». Мы можем дискутировать весь день о значении слова «главный», но для меня это значит:

* Самая главная навигация;
* Поиск по сайту;
* Вторичная навигация (что спорно);
* Внутренняя навигация (по длинной статье, например).

И хотя здесь не может быть «правильного» или «неправильного» использования, поверхностный опрос вкупе с моей собственной интерпретацией говорят, что следующие случаи не подходят для использования **<nav>**:

* Постраничная навигация;
* Социальные ссылки, за исключением тех случаев, когда такие ссылки являются основной навигацией, к примеру, на сайтах About Me и Flavours;
* Тэги к записи в блоге;
* Категории записи;
* Навигация третьего уровня;
* Подвалы, набитые ссылками.

Если вы не уверены, стоит ли оборачивать список ссылок в <nav>, просто спросите у себя: «главная ли это навигация?» Чтобы было легче ответить на этот вопрос, обратитесь к следующим правилам:

* «Не используйте <nav>, если кажется, что в этом случае может подойти и <section> с заголовком <hx>», — Ян Хиксон, IRC.
* Добавите ли вы этот блок в список ссылок «перейти» для улучшения доступности?

Если ответ на оба эти вопроса «нет», то, скорее всего, это не **<nav>**.

## Общие ошибки с элементом **<figure>**

Ах, **<figure>**. Правильным использованием этого элемента вместе с подельником **<figcaption>** не так-то просто овладеть. Рассмотрим некоторые общие проблемы, которые я вижу при использовании **<figure>**.

### Не каждая картинка это **<figure>**

Ранее я советовал вам не писать лишний код там, где этого не требуется. Та же ошибка и здесь. Я видел сайты, где каждая картинка была обёрнута в **<figure>.** Нет никакой необходимости в добавлении экстра-разметки вокруг картинок только ради самого процесса. Вы просто делаете лишнюю работу и нисколько не улучшаете описание содержимого страницы.

Спецификация обозначает **<figure>** как «содержимое в потоке, с необязательным заглавием, самодостаточное, обычно упоминаемое в качестве смысловой единицы в основном тексте». Как раз в этом состоит вся красота элемента **<figure>**, который может быть перемещён из основного содержимого, скажем, в колонку, что никак не повлияет на основной поток документа.

Если это исключительно оформительская картинка, никаким образом не упомянутая в основном документе, то это точно не **<figure>**. Есть и другие варианты использования, но просто спросите себя: «Нужна ли эта картинка для лучшего понимания контекста?» Если нет, то это вероятно не **<figure>,** а, возможно, **<aside>**. Если да, спросите себя: «Можно ли переместить эту картинку в примечания к тексту?» Если ответ на оба вопроса «да», то это, вероятнее всего, **<figure>.**

### Ваш логотип — это не **<figure>**

Плавно переходим к следующей проблеме, вышеупомянутые правила применимы и к ней. Вот пара регулярно встречающихся примеров:

<!-- Не копируйте этот код! Он неправильный! -->

<header>

<h1>

<figure>

<img src="logo.png" alt="Моя компания">

</figure>

Название компании

</h1>

</header>

<!-- Не копируйте этот код! Он неправильный! -->

<header>

<figure>

<img src="logo.png" alt="Моя компания">

</figure>

</header>

Добавить здесь нечего: это совсем неправильно. Мы можем спорить до посинения насчёт того, должно ли лого находиться внутри **<h1>**, но мы здесь не за этим. Настоящая проблема — в неправильном употреблении **<figure>.** Этот элемент должен использоваться, только если он упоминается в документе или контексте общего структурного элемента. Будет честным признать, что ваш логотип вряд ли будет упомянут подобным образом. Просто не делайте так. Всё, что вам нужно, это:

<header>

<h1>Название компании</h1>

<!-- и всё остальное здесь -->

</header>

### Элемент <figure> — это не только картинки

Другое распространённое заблуждение насчёт **<figure>** — что он может быть использован только для картинок. Это не так. Элемент **<figure>** может быть видео, аудио, графиком (на SVG, к примеру), цитатой, таблицей, блоком кода, фрагментом текста или любой комбинацией этих и многих других элементов. Не ограничивайте использование **<figure>** только картинками. Наша работа, как энтузиастов от веб-стандартов, заключается в том, чтобы максимально точно описывать содержимое при помощи разметки.

## Не используйте ненужные атрибуты type

Дело в том, что в HTML5 не нужно добавлять атрибут type для элементов **<script>** и **<style>**. Если эти элементы автоматически добавляются вашей CMS, то избавиться от них может быть непросто, но если вы пишете код руками или полностью контролируете шаблоны, то нет никакого смысла писать избыточные атрибуты. Поскольку все браузеры ожидают, что скриптами будет JavaScript, а стилями CSS, вам это совсем не нужно:

<!-- Не копируйте этот код! Он неправильный! -->

<link type="text/css" rel="stylesheet" href="styles.css">

<script type="text/javascript" src="scripts"></script>

Вместо этого просто напишите:

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

<script src="scripts"></script>

Кроме того, вы можете уменьшить объём кода, который пишете для задания кодировки документа и других вещей. Глава про семантику в «Погружении в HTML5» Марка Пилгрима объясняет всё.

## Неправильное использование атрибутов форм

Вы, должно быть, знаете, что HTML5 предложил набор новых атрибутов для элементов форм. Мы расскажем о них подробнее в ближайшие месяцы, а пока я покажу вам некоторые вещи, которые делать не стоит.

### Одиночные атрибуты

Некоторые новые атрибуты для элементов форм являются одиночными, и только одно их присутствие в разметке обеспечивает смену поведения. Эти атрибуты включают:

* autofocus
* autocomplete
* required

Надо признаться, я нечасто встречал подобное, но если в качестве примера взять атрибут required, то попадалось следующее:

<!-- Не копируйте этот код! Он неправильный! -->

<input type="email" name="email" required="true">

<!-- Ещё один плохой пример -->

<input type="email" name="email" required="1">

В конечном счёте это никому не вредит. HTML-парсер видит атрибут required в разметке, поэтому требуемая функциональность будет включена. Но что, если поставить код вверх ногами и написать required="false"?

<!-- Не копируйте этот код! Он неправильный! -->

<input type="email" name="email" required="false">

Парсер по-прежнему увидит атрибут required и применит поведение, несмотря на то, что вы пытались сказать ему не делать этого. Явно не то, что вам было нужно.

Существует три правильных способа задания одиночных атрибутов, второй и третий из которых нужны только если вы пишете XHTML:

* required
* required=""
* required="required"

Если применить эту запись к нашему примеру (в HTML), получится следующее:

<input type="email" name="email" required>

# Резюме

Было бы просто невозможным перечислить здесь все странные приёмы и образцы разметки, с которыми мне довелось столкнуться, но эти — одни из самых часто встречавшихся.

Перевод оригинальной статьи «[Avoiding common HTML5 mistakes](http://html5doctor.com/avoiding-common-html5-mistakes/)» [Ричарда Кларка](http://richclarkdesign.com/) (Richard Clark), опубликованной на сайте [HTML5Doctor.com](http://html5doctor.com/).