



## Урок 8

# Стандарты web и вспомогательные инструменты

Загрузка проектов на сервер. Проблема «кроссбраузерности». Стандарты HTML/CSS. Будущее за стандартами HTML5 и CSS3. Грамотная, универсальная верстка. Знакомство с препроцессорами на примере LESS.

### Оглавление

[Инструменты разработчика](#)

[Ftp-клиент](#)

[Плагины](#)

[Валидатор](#)

[Оптимизация изображений](#)

[Генератор CSS-градиентов](#)

[Совместимости разных версий браузеров](#)

[Проблема «кроссбраузерности»](#)

[Условный комментарий](#)

## [Стандарты HTML/CSS](#)

[Версии HTML](#)

[Версии CSS](#)

## [Будущее за стандартами HTML5 и CSS3](#)

[Семантические элементы](#)

[Сокращенная форма атрибутов в разделе head](#)

[Атрибуты формы](#)

[Canvas](#)

[Что нам дарит CSS3](#)

[Тени вокруг блоков и текста](#)

[Несколько фоновых изображений](#)

[Прозрачность](#)

## [Препроцессоры](#)

[Что такое пропроцессоры](#)

[LESS](#)

[Переменные](#)

[Операции](#)

[Цветовые операции](#)

[Примеси \(mixins\)](#)

[Заключение](#)

## [Практика](#)

[Применение препроцессора LESS](#)

[Домашнее задание](#)

[Дополнительные материалы](#)

[Используемая литература](#)

# Инструменты разработчика

Какие инструменты могут понадобиться веб-разработчику:

- ftp-клиент (FileZilla, WinSCP, Total Commander . . .);
- Плагины (FireBug, WebDeveloper);
- Валидатор (<http://validator.w3.org/>);
- Оптимизация изображений (TinyPNG);
- Генератор CSS-градиентов (CSS3Factory, CSSMatic Gradients);
- Совместимости разных версий браузеров (Can I Use ...?);
- Тестер для IE (IETester);

## Ftp-клиент

FTP – это протокол передачи файлов. Через него файлы можно скопировать с компьютера на сервер и обратно, отредактировать.

Для использования FTP необходима специальная программа. Их существует большое количество, некоторые из них бесплатные, некоторые могут быть коммерческими. Практически все программы FTP работают таким образом: появляются 2 окна, где 1-е – ваши файлы на сервере в интернет, а 2-е – файлы на компьютере, тем самым можно легко передавать файлы на компьютер/сервер.

Предположим, программа FTP уже выбрана и установлена на компьютер, но для работы необходимо знать FTP-адрес вашего веб-сервера, имя пользователя и пароль. Получить эту информацию можно на хостинге. Это важная информация, и её нужно обязательно записать. Если вся вышеуказанная информация есть, то можно попробовать получить доступ к файлам FTP.

В различных программах действия могут различаться, но в любом случае нужно создать новое соединение с именем пользователя и паролем, что позволит сохранить настройки, тем самым, исключая необходимость ввода данных каждый раз при входе в программу.

После создания профиля и ввода всех данных, нажмите «Соединиться», после чего программа начнёт соединяться с сервером. Если все настройки верны, то через несколько секунд (в зависимости от скорости интернета) можно будет увидеть все свои файлы, находящиеся на сайте.

На веб-сервере находится набор папок и файлов. Подключаясь к серверу, программа отправляет запрос через интернет на специальный сервер и получает данные, которые запрашиваются. Существуют протоколы и с более расширенным функционалом, нежели обычный веб-серфинг.

Используемая FTP-программа даёт возможность создавать, редактировать и удалять файлы и папки на веб-сервере, а также передавать эти файлы на сервер и обратно.

## Плагины

Есть несколько способов открыть:

- Control + SHIFT + I — открывает панель элементов;

- Control + SHIFT + J или F12 — открывает консоль;
- Control + SHIFT + C — позволяет выбрать элемент с помощью мыши в самом документе и открыть его во вкладке «Elements»;
- Можно нажать на любом элементе на странице правой кнопкой мыши и выбрать «Inspect Element (Просмотр кода элемента)»;
- Можно открыть Web Inspector из меню: Иконка с ключом -> Инструменты -> Инструменты разработчика (Tools -> Developer tools).

Чтобы посмотреть список горячих клавиш, откройте любую вкладку кроме консоли и нажмите там "?"

При помощи просмотра кода элемента можно инспектировать любые сайты, чётко определять структуру сайта, вносить изменения в код и многое другое.

## Валидатор

Чистый валидный код в HTML и CSS повышает скорость загрузки и шансы, что на всех без исключения устройствах и браузерах ваша верстка будет выглядеть так, как вы хотите. Поэтому любой верстке нужно проводить валидацию. Валидаторы W3C — это два сервиса, которые с незапамятных времен помогают держать код в чистоте и порядке.

Есть три варианта проверки: проверить код в файле по указанной ссылке, загрузить файл с десктопа или скопировать код прямо в форму на сайте.

HTML-валидатор производит несколько проверок Вашего кода

- Валидация синтаксиса — проверка на наличие синтаксических ошибок;
- Проверка вложенности тэгов — тэги должны быть закрыты в обратном порядке относительно их открытия;
- Валидация DTD — проверка соответствия Вашего кода указанному Document Type Definition; Она включает проверку названий тэгов, атрибутов, и «встраивания» тэгов;
- Проверка на посторонние элементы — проверка выявляет все, что есть в коде, но отсутствует в DTD.

## Оптимизация изображений

Оптимизация графики — не менее важный аспект верстки, чем чистота кода. Функция «Save for Web» в Photoshop незаменима, но с png она работает крайне плохо. Поэтому, после экспорта png-файлов из фотшопа, очень рекомендую вам оптимизировать её в TinyPNG. После обработки файл теряет 30-70% от веса, на качество это никак не влияет.

## Генератор CSS-градиентов

CSS-градиенты позволяют сделать фон из двух и более цветов, плавно переходящих из одного в другой. Они с нами уже достаточно давно и имеют довольно неплохую поддержку браузерами. Большинство современных браузеров понимает их без префиксов, для IE9 и старше есть Gradient Filter, также для IE9 можно использовать SVG.

Градиенты могут быть использованы везде, где используются картинки: в фонах, в качестве буллетов списков, они могут быть заданы в content или border-image.

## Совместимости разных версий браузеров

Свойство сайта отображаться и работать во всех наиболее популярных браузерах одинаково, когда отображение параметров страниц при использовании разных браузеров и их различных версий и

модификаций близко к исходному дизайнерскому и функциональному виду сайта. Для этого необходимо пользоваться специальными программами, чтобы сайт выглядел во всех браузерах одинаково.

Еще один инструмент, который приходится использовать ради IE - это тестировщик IETester. Он позволяет проверять, как выглядит сайт во всех версиях IE, начиная с версии 5.5.

## Проблема «кроссбраузерности»

В процессе своей работы любой веб-разработчик сталкивается с таким понятием, как кроссбраузерность. Это понятие означает, что любой сайт должен выглядеть одинаково во всех браузерах. А главным виновником этой проблемы является браузер Internet Explorer (IE). Рассмотрим пути решения проблемы кроссбраузерности:

1. Необходимо проверять, как выглядит сайт во всех популярных браузерах с некоторой периодичностью во время верстки. Тогда вы сможете оперативно устранять несоответствия.
2. Некоторые теги HTML, например, заголовки, параграфы, списки, изначально имеют определенный набор свойств и значений. Эти свойства могут определяться каждым браузером по-разному. Для того, чтобы вид HTML страницы не зависел от того, с помощью какого браузера её просматривают, используется сброс стилей CSS. Обычно, если используется сброс стилей, то их определяют в отдельном файле, который подключают на веб-страницу при помощи тега `<link />`.
3. Существует возможность воспользоваться хаками. Хак — исправление ошибки или добавление новой функции посредством использования другой недокументированной или некорректно реализованной особенности. Возникает необходимость добавлять хаки в основном для IE, но ими пользоваться не рекомендуется, т.к. это противоречит спецификации CSS. Чтобы исправить несоответствия отображения в IE, лучше воспользоваться следующим пунктом.
4. Объявить условный комментарий, который представляет собой обычный HTML-комментарий, только в квадратных скобках указывается условие, которое понимает только браузер IE, а всеми остальными браузерами игнорируется.

### Условный комментарий

```
<!--[if gte IE 6]>  
  
    <link href="css/style_ie.css" rel="stylesheet" type="text/css" />  
  
<![endif]-->
```

То, что находится между `if` и `endif` IE выполнит. В данном случае подключится файл стилей CSS. В этом файле можно писать только те стили CSS, которые будут относиться к IE. Для того, чтобы точно определять, к каким версиям IE применять стили, можно использовать дополнительные логические операторы:

- `lt` – версия меньше указанной;
- `lte` – версия меньше или равно указанной;
- `gt` – версия больше указанной;

- gte - версия больше или равно указанной.

При помощи других логических операторов (& - и, | - или, ! - не) выполняется группировка условных комментариев для различных версий IE:

- [if (IE 6) & (IE 7)] – 6-я версия И 7-я версия;
- [if (IE 6) | (IE 7)] – 6-я версия ИЛИ 7-я версия;
- [if !(IE 8)] – НЕ 8-я версия.

## Стандарты HTML/CSS

### Версии HTML

Стандарт HTML зародился в конце 80-х - начале 90-х годов. Но только в 1995 году был принят стандарт HTML версии 2.0. Затем стандарт активно развивался до 2000 года, и остановилось развитие на версии 4.01 Strict. Как раз в это время начал развиваться стандарт XHTML, который основывается на языке представления данных XML, и предъявляет к документам более строгие правила оформления. Существуют три версии стандарта XHTML – это версии 1.0, 1.1 и 2.0. Но версия 2.0 не достигла статуса рекомендации, и к 2010 деятельность по разработке стандарта XHTML прекращена, и группа разработчиков переведена на разработку стандарта HTML 5, который разрабатывается в настоящее время.

### Версии CSS

CSS версии 1 была принята в 1996 году, а в 1998 году она была расширена версией 2. Рабочий проект CSS версии 2.1 появился в 2006 году, а был принят организацией w3c в 2011 году. В нём были исправлены ошибки версии 2 и изменены некоторые части спецификации. В настоящее время разрабатывается CSS версии 3, которая имеет множество нововведений и дополнений.

## Будущее за стандартами HTML5 и CSS3

За один урок невозможно рассказать о нововведениях и преимуществах новых стандартов HTML5 и CSS3, поэтому я вам расскажу о HTML5 и CSS3 то, что вы сможете применять уже сегодня. Для начала ответим на следующий вопрос, можно ли применять эти стандарты сегодня, ведь они ещё только разрабатываются. Можно, ведь HTML5 и CSS3 уже поддерживаются всеми современными браузерами, включая IE, начиная с 9-й версии. Но существует проблема: 9-ю версию IE возможно установить только на ОС Windows Vista или Windows 7. Компания Microsoft отказалась от поддержки Windows XP. Но в настоящее время у многих пользователей установлен Windows XP, и пройдет еще немалое количество времени, пока XP не исчезнет с компьютеров пользователей. И поэтому, таким пользователям невозможно обновить IE дальше 8 версии. Новые теги HTML 5 можно включить в документ при помощи JavaScript, а в случае с CSS 3 пользователь, просматривающий сайт через IE меньше 9 версии, не увидит эффектов применения стилей CSS 3.

При помощи HTML 5 можно заменить основные элементы в структуре макета специальными тегами и тем самым представить более четко структуру сайта.

## Семантические элементы

`<section>` - является способом разделения страницы или статьи на тематические разделы. Не используется многократно, обычно содержит заголовки.

`<header>` - образует содержимое верхней (шапки) части страницы или её секции. Объединяет вводную информацию и навигационные элементы, может располагаться в любой части страницы. В HTML-документе может содержаться одновременно несколько элементов `<header>`. Не является обязательным.

`<footer>` - формирует содержимое нижней части (подвала) страницы или её секции. Предназначен для размещения информации о веб-ресурсе, например, сведения об авторских правах, ссылки на условия использования, контактную информацию, ссылки на связанное содержимое и т.п. В одном веб-документе может быть несколько элементов `<footer>`.

`<aside>` - группирует содержимое, связанное с окружающим его контентом напрямую, но которое можно считать отдельным. Чаще всего элемент позиционируется как боковая колонка.

`<hgroup>` - используется для группировки элементов `<h1>` - `<h6>` в случае, когда заголовок имеет сложную структуру, например, имеет уточняющие подзаголовки, альтернативные заголовки.

`<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>`.

`<article>` - используется для группировки записей: публикаций, статей. Представляет собой независимый, обособленный блок, предназначенный для многократного использования.

`<main>` - группирует основной контент веб-страницы. Содержимое элемента должно быть уникальным на странице и не должно отображаться где-либо еще на сайте.

`<section>` - Является способом разделения страницы или статьи на тематические разделы.

Возможность вставки аудио и видео без использования Flash. Это стало возможным благодаря новым тегам, добавленным в HTML5.

## Сокращенная форма атрибутов в разделе head

- `<!DOCTYPE html>`
- `<meta charset="UTF-8">`
- `<script src="script.js"></script>`
- `<link rel="stylesheet" href="styles.css">`

## Атрибуты формы

Появились богатые возможности по работе с формами, масса новых типов полей. Например, e-mail, tel, url. Благодаря этим типам Вам больше не нужно писать код на JavaScript для проверки формы. Теперь с помощью HTML5 форма автоматически проверяется на корректность заполнения, сразу выдавая все сообщения об ошибках пользователю. Причём всё это сделано очень красиво и аккуратно, впрочем, через CSS можно будет изменить внешний вид всего этого. Также появился элемент `datalist`, позволяющий создать список вариантов, которые пользователь сможет выбрать при

наборе текста. Такой функционал очень часто используется при наборе поискового запроса, когда вводя первые символы, сразу появляются возможные варианты. Никогда такая сложная функциональность не создавалась так легко.

Упрощённая реализация drag and drop. Теперь реализовать подобный механизм стало намного проще. Пример использования drag and drop – перетащили мышкой картинку товара в корзину, и он автоматически туда добавился. Раньше это была достаточно сложная задача, теперь же она делается в течение нескольких минут.

## Canvas

Позволяет рисовать прямо на Web-странице через JavaScript, генерировать картинки без перезагрузки страницы. Раньше без использования PHP это было сделать невозможно.

## Что нам дарит CSS3

Закругленные уголки у блоков задаются при помощи свойства border-radius.

Можно задавать скругления как всем 4 сторонам вместе, так и каждой в отдельности, тогда отсчёт у уголков производится от левого верхнего угла и идёт по часовой стрелке. Также возможно задать значения скругления через слеш, тогда первое значение будет задавать скругление по горизонтали, а второе - по вертикали. Таким образом из блока можно сделать эллипс.

## Тени вокруг блоков и текста

Для блоков:

```
box-shadow: 2px 3px 4px #ccc;
```

Для текста:

```
text-shadow: 2px 3px 4px #ccc;
```

- 2px – смещение по оси X;
- 3px – смещение по оси Y;
- 4px – радиус размытия;
- #ccc – цвет тени;

## Несколько фоновых изображений

Для того, чтобы задать несколько картинок для одного элемента в качестве фонового изображения, их необходимо перечислить через запятую для свойства background:

```
background: url(img/HTML5.png) top center no-repeat,  
            url(img/css3.png) top right no-repeat;
```

## Прозрачность

При помощи свойства opacity можно задавать элементу полупрозрачность. Значение указывается в интервале от 0 до 1, где 0 - элемента не будет видно, но при этом он не исчезнет из потока документа, как в случае со свойством display: none. А при значении 1 элемент будет четко виден:

```
opacity: 0.1 .... 1;
```

Появилась возможность управлять элементами на странице. Если задать для многострочного поля ввода <textarea> свойство resize, то тогда пользователь получает возможность растягивать данное поле. При значении horizontal можно растянуть его по горизонтали, vertical – по вертикали, а при значении both – и по горизонтали и по вертикали.

## Препроцессоры

### Что такое препроцессоры

CSS-препроцессоры — это «программистский» подход к CSS. Они позволяют использовать при написании стилей свойственные языкам программирования приёмы и конструкции: переменные, вложенность, наследуемость, циклы, функции и математические операции. Синтаксис препроцессоров похож на обычный CSS. Код, написанный на языке препроцессора, не используется прямо в браузере, а преобразуется в чистый CSS-код с помощью специальных библиотек.

### LESS

- Динамический язык стилей;
- Продукт с открытым исходным кодом;
- Может работать на стороне клиента или на стороне сервера под управлением Node.js или Rhino.

## Переменные

Синтаксис переменных:

```
@название_переменной: значение_переменной;
```

Создав переменную один раз, можно использовать её в любом месте кода.

Например:

```
@color_red: #f00;
background-color: @color_red;
color: @color_red;
border-color: @color_red;
```

Во всех местах, где указана переменная, LESS заменит строку `@color_red` на `#f00`. Теперь, если понадобится изменить цвет, не нужно искать все его объявления в файле, а достаточно просто изменить значение переменной в одном месте.

Область видимости переменных:

- Переменные можно объявлять как «снаружи» правил, так и «внутри». В случае «внутреннего» объявления переменная будет доступна только внутри правила, в котором она объявлена:
- Если переменная объявлена и «внутри» правила, и «снаружи» — LESS применит «внутреннее» значение.
- Таким образом можно «переопределять» глобальные переменные в локальном контексте.
- Важно отметить, вопреки тому, что вы только что прочитали, переменные в LESS больше похожи на константы. Это значит, что они, в отличие от переменных, могут быть определены только один раз.

## Операции

Можно использовать операции умножения, деления, сложения и вычитания.

```
@width: 100px;
.class_1 {
  width: @width;
}
.class_2 {
  width: @width / 3;
}
.class_3 {
  height: 100px + 20px;
}
```

## Цветовые операции

На практике есть немало случаев, когда мы начинаем с базового цвета и нуждаемся в слегка затемненном или осветленном его варианте.

```
@color-button: #ccc;
.button {
  width: 150px;
  height: 75px;
  background:@color-button;
  border:5px solid @color-button - #222;
}
```

Этот код создает кнопку с немного затемненной рамкой. Это частая ситуация и определение лишь одного цвета – большая помощь.

Есть гораздо больше возможностей для работы с цветом; LESS позволяет манипулировать ими на канальном уровне.

- осветлять lighten;
- затемнять darken;
- насыщать desaturate;
- обесцвечивать saturate;
- вращать цвета spin.

## Примеси (mixins)

Примеси в LESS избавят вас от набора излишнего кода. Вам когда-нибудь приходилось создавать закругленную рамку, в которой только верхние углы скругленны?

```
.border_top {
  -webkit-border-top-left-radius: 6px;
  -webkit-border-top-right-radius: 6px;
  -moz-border-radius-topleft: 6px;
  -moz-border-radius-topright: 6px;
  border-top-left-radius: 6px;
  border-top-right-radius: 6px;
}
.block {
  background: #333;
  .border_top;
}
```

Благодаря такому синтаксису мы можем использовать любой элемент в качестве примеси.

## Заключение

Препроцессоры очень удобное изобретение, которым я советую пользоваться всем разработчикам, Это сокращение кода, возможности создания большого функционала сайтов на простом CSS.

Использовать препроцессоры стало проще, чем раньше, причём, намного проще. Для этого нужно лишь установить программу, которая будет следить за файлами, предназначенными для препроцессора, и при их изменении будет компилировать содержимое этих файлов в чистый CSS-код. Для более продвинутых пользователей есть специальные сборщики проектов.

Не думайте, что если вы используете программу для препроцессоров, а не сборщик проектов, то вы недостаточно круты. На самом деле, такие сборщики предлагают полный контроль и расширенные настройки, а не делают из вас джедая.

Разумеется, как и в любой другой области, всегда есть конкуренция, и на рынке препроцессоров есть несколько препроцессоров. Какой лучше выбрать, какой синтаксис лучше подходит, это выбирать вам! Они очень похожи, с минимальными отличиями.

## Практика

### Применение препроцессора LESS

```
<header></header>
<main></main>
<footer>
  <div class="block_1"></div>
  <div class="block_2"></div>
  <div class="block_3"></div>
</footer>

@width: 1400px;
@height: 300px;
* {
  margin:0;
  padding:0;
}
header {
  width:@width;
  height:200px;
  background-color: bisque;
  margin:0 auto;
}
main {
  width:@width;
  height:@height * 2;
  background-color: cadetblue;
  margin:0 auto;
}
footer {
  width:@width;
  height:@height;
  background-color: blueviolet;
  margin:0 auto;
  .block_1, .block_2, .block_3 {
    width:@width / 3;
    float: left;
    height:@height;
  }
  .block_1 {
    background-color: aqua;
  }
  .block_2 {
    background-color: #3074ff;
  }
  .block_3 {
    background-color: #42ffba;
  }
}
```

# Домашнее задание

## Финальный проект

1. Завершить работу над своим магазином
2. Разместить сайт в интернете
3. Проверить проект на валидность и кроссбраузерность.

# Дополнительные материалы

1. [Немного о кроссбраузерности](#)
2. [Кроссбраузерность веб-сайта](#)
3. [Новые атрибуты html5](#)
4. [Использование html5](#)
5. [HTML5 для начинающих](#)
6. [Примеры CSS3](#)
7. [Новое в CSS3](#)
8. [CSS Препроцессоры](#)
9. [Начинаем писать CSS](#)

# Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81%D0%B1%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80%D0%BD%D0%BE%D1%81%D1%82%D1%8C>
2. <http://htmlbook.ru/html5>
3. [https://ru.wikipedia.org/wiki/CSS#.D0.A3.D1.80.D0.BE.D0.B2.D0.B5.D0.BD.D1.8C\\_3\\_.28CSS3.29](https://ru.wikipedia.org/wiki/CSS#.D0.A3.D1.80.D0.BE.D0.B2.D0.B5.D0.BD.D1.8C_3_.28CSS3.29)
4. <http://htmlbook.ru/css3>
5. <http://html5book.ru/html5-semantic-elements/>
6. [https://mrmlnc.gitbooks.io/less-guidebook-for-beginners/content/chapter\\_1/css-reprocessors.html](https://mrmlnc.gitbooks.io/less-guidebook-for-beginners/content/chapter_1/css-reprocessors.html)
7. <http://cmsart.ru/joomla/joomla-articles/joomla-1-7/chto-takoe-ftp-i-dlya-chego-on-nuzhen>