**[Практическое использование Emmet (Zen Coding)](http://ts-soft.ru/blog/emmet" \o "Практическое использование Emmet (Zen Coding))**

[**http://ts-soft.ru/blog/emmet**](http://ts-soft.ru/blog/emmet)

4 марта 2013 в 14:53

[Разработка](http://ts-soft.ru/blog/category/development)

Emmet (предыдущее название — Zen Coding) — это набор плагинов для различных популярных текстовых редакторов, используемых для скоростного написания HTML и CSS кода. Синтаксис Emmet довольно прост и не требует больших усилий со стороны разработчика для его изучения, в то же время использование этого инструмента позволяет ускорить процесс написания верстки в несколько раз.

**Основные операции в Emmet**

Итак, для написания HTML в Emmet используются 12 типов селекторов:

* #  — создает атрибут id
* .  — создает атрибут class
* []  — создает любые другие атрибуты, в том числе и пользовательские
* >  — делает переход на один уровень ниже
* +  — создает соседние элементы на том же уровне
* ^  — делает переход на уровень вверх
* \*  — умножает элементы
* $  — заменяется числом, каждый раз увеличивающимся на единицу
* $$  — то же самое, только двухзначное
* {}  — добавляет текстовое содержимое элементам
* ()  — группирует элементы
* :  — используется для некоторых элементов, таких как <input>, <a>, <link> и др., и задает для них атрибуты

Рассмотрим на примерах.

Для того, чтобы создать один элемент с атрибутами class и id, необходимо написать:

1. div#content.column

После набора этой строки нажимаем клавишу Tab (в Notepad++ клавиши Ctrl+Alt+Enter) и получаем результат:

1. <div id="content" class="column"></div>

В квадратных скобках можно задавать любые атрибуты тегам. Если необходимо задать их несколько, то они разделяются пробелом, например:

1. input[type="text" name="address" placeholder="some text" disabled]

Результат:

1. <input type="text" name="address" placeholder="some text" disabled="">

Для некоторых тегов есть еще более лаконичный способ задавать атрибуты, указывая их значения через двоеточие. Например, так можно задавать для элемента input его тип:

1. input:text

или еще короче:

1. input:t

Результат:

1. <input type="text" name="" id="">

Аналогично задаются типы: hidden (h), search, email, url, password (p), datetime, datetime-local, date, month, week, time, number, range, color, checkbox (c), radio (r), file (f), submit (s), image (i), reset, button (b).

Таким же образом можно задавать атрибуты тегов метаданных <link> и <meta>. Вот так можно подключить внешний css-файл:

1. link:css

Результат:

1. <link rel="stylesheet" type="text/css" href="style.css" media="all">

Остается только исправить значение href.

Файлы .js подключаются аналогично:

1. script:src

получаем:

1. <script type="text/javascript" src=""></script>

Селекторы > и + создают, соответственно, дочерние и соседние элементы, а селектор ^ позволяет подниматься в иерархии на один уровень. Также есть возможность умножать элементы с помощью знака \*. Например, следующая строка создаст список с пятью пунктами:

1. ul>li\*5

Если необходимо добавить тегам текстовое содержимое, нужно использовать фигурные скобки, например, так:

1. div#dialog>p>span{text}

Результат:

1. <div id="dialog">
2. <p>
3. <span> text </span>
4. </p>
5. </div>

В Emmet есть еще одна интересная возможность: он позволяет добавлять нумерацию, например, атрибутов множественных элементов с помощью знака $. Нумерация начинается с единицы. Это может оказаться полезным, если, например, необходимо создать  большое число элементов с названиями классов, различающимися только цифрой на конце, или выпадающий список, у пунктов которого атрибут value должен нумероваться с единицы:

1. select>option[value="$"]\*10

Результатом выполнения в данном случае будет <select>, содержащий 10 тегов <option> со значениями атрибута value от 1 до 10.

Для сложных конструкций можно применять группировку элементов, используя круглые скобки. Однако эта возможность существует не во всех редакторах, например, она не поддерживается расширением Web Essentials 2012 для Visual Studio.

**Шаблоны HTML**

Emmet также предоставляет возможность создавать разметку с использованием различных шаблонов HTML, наиболее интересный из них — html:5. Такая команда даст следующий результат:

1. <!doctype html>
2. <html lang="en">
3. <head>
4. <meta charset="UTF-8">
5. <title>Document</title>
6. </head>
7. <body>
9. </body>
10. </html>

**Условные комментарии**

С помощью команд cc:ie и cc:noie можно добавить условные комментарии. Команда cc:ie выдает разметку, код внутри которой будет распознаваться только Internet Explorer, cc:noie, наборот, генерирует комментарии, содержимое которых будет доступно только всем остальным браузерам. Результат выполнения первой:

1. <!--[if IE]>
2. <![endif]-->

и второй:

1. <!--[if !IE]><!-->
2. <!--<![endif]-->

**Сокращенные конструкции**

Существует также несколько сокращенных конструкций для создания таблиц и списков: table+, ul+, ol+, dl+, select+. Результаты их выполнения будут такими:

1. <table>
2. <tr>
3. <td></td>
4. </tr>
5. </table>
7. <ul>
8. <li></li>
9. </ul>
11. <ol>
12. <li></li>
13. </ol>
15. <dl>
16. <dt></dt>
17. <dd></dd>
18. </dl>
20. <select name="" id="">
21. <option value=""></option>
22. </select>

**Оборачивание в аббревиатуру**

В случае, если часть разметки уже существует, и необходимо дополнить её, не переписывая заново, можно воспользоваться функцией оборачивания. Для этого нужно:

* установить курсор на тот тег, который должен быть обернут, либо на текст, который должен быть помещен в новый тег,
* затем открыть диалоговое окно этой команды нажатием клавиш Ctrl+Shift+G в Sublime Text 2 (здесь это будет командная строка внизу экрана) или Ctrl+Shift+Alt+Enter в Notepad++
* и написать там соответствующую аббревиатуру Emmet.

Например, у нас существует следующая разметка:

1. <div>
2. <p>Hello</p>
3. </div>

Необходимо создать внутри div новый элемент div с классом wrap и поместить в него существующий параграф с текстом. Устанавливаем курсор внутри тега <p> или даже закрывающего тега </p>, открываем диалог и пишем там:

1. div.wrap

Результат будет таким:

1. <div>
2. <div class="wrap">
3. <p>Hello</p>
4. </div>
5. </div>

Подобным образом можно оборачивать не только отдельные элементы, но и множественные, при этом даже необязательно, чтобы они были одинаковыми. Для этого нужно выделить тот набор элементов, в котором каждый элемент должен быть обернут, и в окне команды с помощью селектора \* указать, какой элемент обертки нужно размножить. Например, есть div, в котором находится несколько параграфов:

1. <div>
2. <p>About us</p>
3. <p>Catalog</p>
4. <p>Contacts</p>
5. </div>

Допустим: через некоторое время разработчику понадобилось, чтобы каждый из этих параграфов находился внутри ссылки, которая, в свою очередь, находится внутри пункта <li> списка <ul>. Выделяем все параграфы, нажимаем соответствующие клавиши и в открывшемся окне пишем:

1. ul>li\*>a

В результате получим следующий код:

1. <div>
2. <ul>
3. <li><a href=""><p>About us</p></a></li>
4. <li><a href=""><p>Catalog</p></a></li>
5. <li><a href=""><p>Contacts</p></a></li>
6. </ul>
7. </div>

**Удаление тегов**

С помощью Emmet можно упростить процесс удаления ненужных тегов. Достаточно установить курсор на тот тег, который необходимо удалить, и вызвать команду нажатием клавиш Ctrl+Shift+; (Sublime Text 2) — удалится не только открывающий тег, но и закрывающий, а также строки, на которых они находились. В Notepad++ нет горячих клавиш по умолчанию для этой команды, но их можно настроить в меню «Опции» > «Горячие клавиши». В открывшемся окне выбрать вкладку «Plugin commans» и назначить там комбинацию клавиш для команды «Remove tag».

**Работа с CSS**

Для быстрого написания  CSS кода в Emmet существует большой набор специальных сокращений свойств и их значений. Самый простой и эффективный способ использования этих сокращений — писать аббревиатуры, состоящие из первых букв свойств и их значений, например, сокращение tdn развернется как:

1. text-decoration: none;

Точно такой же результат дадут записи td-n и td:n.

Если нужно указать несколько значений для свойства, это можно сделать через знак дефиса:

1. m10-20

Результат:

1. margin: 10px 20px;

Подобным образом задаются и отрицательные значения:

1. m-10--20

Результат:

1. margin: -10px -20px;

Если после числовых значений свойств не указывать единицы измерения, то по умолчанию автоматически выставляются пиксели, за исключением случая, когда значения нецелые — тогда выставляются em. Также существует несколько сокращений для единиц измерения: вместо % можно писать p, вместо em — e, а вместо ex — x.  
Например:

1. w100p

Результат:

1. width: 100%;

Если добавить в конец аббревиатуры восклицательный знак, то к свойству будет добавлено значение !important, например:

1. p20!

даст результат:

1. padding: 20px !important;

Однако наиболее интересная возможность для работы с CSS, которую предоставляет Emmet — это создание свойств с вендорными префиксами. Для ряда таких свойств (border-radius, transform и т. д.) автоматически создаются копии со всеми необходимыми вендорными префиксами. Так, например, аббревиатура brs5 будет расширена следующим образом:

1. -webkit-border-radius: 5px;
2. -moz-border-radius: 5px;
3. border-radius: 5px;

Также Emmet значительно облегчает процесс написания стилей для элементов с градиентом. Для этого используется специальная аббревиатура lg (или linear-gradient), после которой в круглых скобках указываются параметры градиента:

1. lg(left, #ddd, #ccc)

Результат:

1. background-image: -webkit-gradient(linear, 0 0, 100% 0, from(#ddd), to(#ccc));
2. background-image: -webkit-linear-gradient(left, #ddd, #ccc);
3. background-image: -moz-linear-gradient(left, #ddd, #ccc);
4. background-image: -o-linear-gradient(left, #ddd, #ccc);
5. background-image: linear-gradient(left, #ddd, #ccc);

Для редактирования значений в уже написанных свойствах с вендорными префиксами можно воспользоваться командой обновления значения (горячие клавиши в Sublime Text 2 — Ctrl+Shift+R). Сначала нужно исправить значение в одной из строк, затем вызвать команду, и данное значение обновится во всех копиях этой строки.

**Математические вычисления**

Emmet позволяет выполнять несложные арифметические вычисления прямо во время написания HTML или CSS кода, не используя какие-либо сторонние калькуляторы. Нередко возникают ситуации, когда необходимо вычислить значения отступов, ширины элементов и других значений; данная особенность освобождает разработчика от необходимости переключаться между окнами редактора и калькулятора. Поддерживаются операции сложения, вычитания, умножения и деления. В Sublime Text 2 горячие клавиши для этой операции — Ctrl+Shift+Y, в Notepad++ по умолчанию горячих клавиш для этой команды нет, необходимо задать их самому (меню «Опции» > «Горячие клавиши»).

**В каких случаях стоит использовать Emmet**

В перечисленных выше примерах может быть не вполне очевидна разница между написанием кода с использованием Emmet и без него. Однако его преимущества становятся очевидны, когда возникает необходимость написания большого количества однотипного кода с многократным повторением одних и тех же элементов. Как правило, даже если прибегать к методу копирования и вставки повторяющихся участков, процесс занимает довольно длительное время, вдобавок есть возможность запутаться.

Например, необходимо сделать таблицу из пяти строк и пяти столбцов, в первой колонке должна быть нумерация строк (от 1 до 5). Такая разметка создается при помощи одной небольшой строки:

1. table>tr\*5>td{$}+td\*4

Результат (37 строк кода):

1. <table>
2. <tr>
3. <td>1</td>
4. <td></td>
5. <td></td>
6. <td></td>
7. <td></td>
8. </tr>
9. <tr>
10. <td>2</td>
11. <td></td>
12. <td></td>
13. <td></td>
14. <td></td>
15. </tr>
16. <tr>
17. <td>3</td>
18. <td></td>
19. <td></td>
20. <td></td>
21. <td></td>
22. </tr>
23. <tr>
24. <td>4</td>
25. <td></td>
26. <td></td>
27. <td></td>
28. <td></td>
29. </tr>
30. <tr>
31. <td>5</td>
32. <td></td>
33. <td></td>
34. <td></td>
35. <td></td>
36. </tr>
37. </table>

Многие плагины, используемые разработчиками, требуют определенной разметки тех элементов, к которым они подключаются. Зная заранее о том, какую структуру необходимо создать, гораздо удобнее сделать это сразу, одной командой. Так будет выглядеть создание разметки виджета «Accordion» популярной библиотеки jQuery UI:

1. div#accordion>(h3+div>p)\*5

Результат:

1. <div id="accordion">
2. <h3></h3>
3. <div>
4. <p></p>
5. </div>
6. <h3></h3>
7. <div>
8. <p></p>
9. </div>
10. <h3></h3>
11. <div>
12. <p></p>
13. </div>
14. <h3></h3>
15. <div>
16. <p></p>
17. </div>
18. <h3></h3>
19. <div>
20. <p></p>
21. </div>
22. </div>

Или виджета «Tabs»:

1. div#tabs>(ul>li\*5>a[href="#tabs-$"])+div#tabs-$\*5>p

Результат:

1. <div id="tabs">
2. <ul>
3. <li><a href="#tabs-1"></a></li>
4. <li><a href="#tabs-2"></a></li>
5. <li><a href="#tabs-3"></a></li>
6. <li><a href="#tabs-4"></a></li>
7. <li><a href="#tabs-5"></a></li>
8. </ul>
9. <div id="tabs-1">
10. <p></p>
11. </div>
12. <div id="tabs-2">
13. <p></p>
14. </div>
15. <div id="tabs-3">
16. <p></p>
17. </div>
18. <div id="tabs-4">
19. <p></p>
20. </div>
21. <div id="tabs-5">
22. <p></p>
23. </div>
24. </div>

**Фильтрация**

В Emmet есть возможность изменять генерируемый код при помощи различных фильтров. Фильтр включается в конце аббревиатуры, после вертикальной черты. Существуют следующие фильтры:

* haml — генерирует результат как HAML-код, включен по умолчанию, если выбран синтаксис HAML.

Без фильтра аббревиатура

1. #content>ul.nav>li\*2

развернется следующим образом:

1. <div id="content">
2. <ul class="nav">
3. <li></li>
4. <li></li>
5. </ul>
6. </div>

Если добавить фильтр |haml, получим разметку в виде HAML-кода:

1. #content
2. %ul.nav
3. %li
4. %li

* html — генерирует HTML-код, включен по умолчанию везде, кроме HAML-файлов
* e — заменяет символы <, > и & соответственно на последовательности &lt;, &gt; и &amp

Добавим в предыдущий пример фильтр |e:

1. #content>ul.nav>li\*2|e

Результат:

1. &lt;div id="content"&gt;
2. &lt;ul class="nav"&gt;
3. &lt;li&gt;&lt;/li&gt;
4. &lt;li&gt;&lt;/li&gt;
5. &lt;/ul&gt;
6. &lt;/div&gt;

* c — добавляет комментарии в код
* s — выводит всю разметку в одну строку
* t — работает только при оборачивании тегов; обрезает маркеры оборачиваемых пунктов списка.

Пусть нужно каждый из пунктов следующего списка обернуть в ссылку:

1. <div>
2. 1. About us
3. 2. Catalog
4. 3. Contacts
5. </div>

Список нумерованный, и, если бы количество пунктов было в нем гораздо большим, то процесс удаления номеров занял бы значительное время. Вместо этого можно применить команду оборачивания тегов, описанную выше, добавив в конце фильтр |t:

1. a\*|t

Результат:

1. <div>
2. <a href="">About us</a>
3. <a href="">Catalog</a>
4. <a href="">Contacts</a>
5. </div>

**Настройка**

Поскольку в плагинах Emmet предусмотрена поддержка расширений с помощью json- и js-файлов, существует несколько возможностей пользовательской настройки, которые позволяют разработчикам полностью адаптировать этот инструмент под свои нужды:

1. Настройка горячих клавиш команд

Каждой команде Emmet можно назначить любое удобное сочетание горячих клавиш. В Sublime Text 2 для этого нужно открыть в папке плагина файл Default(Windows).sublime-keymap (для Windows), найти нужную команду и назначить для нее другое сочетание клавиш. Для Notepad++, как было сказано ранее, настройка горячих клавиш осуществляется в меню «Опции» > «Горячие клавиши» на вкладке «Plugin commands».

2. Возможность расширения позволяет добавлять свои команды и фильтры

Можно в файле Emmet.sublime-settings определить параметр extensions\_path, и Emmet будет подгружать все .js-файлы автоматически при запуске Sublime Text 2.

3. Добавление и редактирование сниппетов

Все расширения аббревиатур и сниппеты Emmet можно редактировать в файле snippets.json, там же можно добавлять новые. Это может быть полезным, если разработчику не нужны некоторые атрибуты тегов, которые создаются по умолчанию, или, наоборот, всегда нужны те, которые не создаются.

4. Настройка команд

Чтобы скорректировать результаты выполнения некоторых команд, нужно создать в папке расширений файл preferences.json, в котором указать значения опций, полный список и описание которых перечислено в документации: <http://docs.emmet.io/customization/preferences/>.

Например, встроенный генератор градиента, о котором шла речь в разделе «Работа с CSS», не поддерживает градиенты для Internet Explorer. Это можно исправить, сделав небольшую настройку в preferences.json:

1. {
2. "css.gradient.prefixes": "webkit, moz, o, ms"
3. }

К трём префиксам для градиента, которые были установлены по умолчанию, мы добавили еще один — «ms». Результат работы генератора будет выглядеть таким образом:

1. background-image: -webkit-gradient(linear, 0 0, 0 100%, from("555"), to("666"));
2. background-image: -webkit-linear-gradient("555", "666");
3. background-image: -moz-linear-gradient("555", "666");
4. background-image: -o-linear-gradient("555", "666");
5. background-image: -ms-linear-gradient("555", "666");
6. background-image: linear-gradient("555", "666")

Теперь код сгенерирует градиент в IE 10, но в браузерах ниже IE 9 он работать не будет. Для того, чтобы задать градиент для старых версий IE, нужно задать такое свойство:

1. filter: progid:DXImageTransform.Microsoft.gradient(GradientType=0,startColorstr='#555555', endColorstr='#666666');

Эту проблему в последней версии Emmet можно решить только написанием сниппета для CSS:

1. "lg:ie": "filter: progid:DXImageTransform.Microsoft.gradient(GradientType=0,startColorstr='${1}', endColorstr='${2}');"

5. Настройка синтаксиса

Каким образом будет выглядеть разметка, сгенерированная Emmet, зависит от того, какой doctype был указан на странице. Всего у Emmet существует несколько профайлов для форматирования синтаксиса:

* html — профайл по умолчанию, внутри одиночных тегов не ставится слеш, например: <br>;
* xhtml — такой же, но одиночные теги закрываются слешем: <br />;
* xml — профайл по умолчанию, если в редакторе определен синтаксис XML или XSL; каждый тег находится на новой строке с соответствующим отступом, все пустые теги закрываются слешем: <br/>.
* line — выводит результат без отступов и в одну строку.

Если данный профайл необходимо каким-либо образом изменить, или применить к одному синтаксису профайл для другого, то для этого нужно в папке расширений создать файл syntaxProfiles.json и произвести там соответствующую настройку.

Например, если нужно, чтобы в html-разметке все одиночные теги были закрыты слешем, можно назначить для html профайл для форматирования xhtml следующим образом:

1. {
2. "html": "xhtml"
3. }

Или создать собственный профайл (список доступных свойств в документации: <http://docs.emmet.io/customization/syntax-profiles/>):

1. {
2. "html": {
3. "self\_closing\_tag": true
4. }
5. }

**Установка Emmet в Notepad++ на Windows**

Пока Emmet недоступен для установки через менеджер плагинов, поэтому необходимо установить его вручную, выполнив следующие шаги:

1. Установить через менеджер плагинов (Плагины > Plugin Manager > Show Plugin Manager) Python Script
2. Скачать архив <https://github.com/emmetio/npp/raw/master/emmet-npp.zip> и распаковать его
3. Скопировать файл EmmetNPP.dll в C:Program FilesNotepad++plugins
4. Скопировать папку EmmetNPP в %USERPROFILE%AppDataRoamingNotepad++pluginsconfig
5. Запустить Notepad++

**Установка Emmet в Sublime Text 2**

Устанавливаем Package Control — менеджер пакетов (плагинов):<http://wbond.net/sublime_packages/package_control>  
Открываем командную панель нажатием Сtrl+Shift+P и находим Package Control: Install Package.

Нажимаем Enter, и теперь мы должны увидеть список пакетов, доступных для установки. Находим пакет Emmet, нажимаем Enter, чтобы установить, и перезапускаем редактор.

**Установка Zen Coding в Visual Studio 2012**

Так как для Visual Studio 2012 пока не выпущен Emmet v1.0, можно установить только более старую версию — Zen Coding. Поддержку Zen Coding в Visual Studio 2012 осуществляет [расширение Web Essentials](http://visualstudiogallery.msdn.microsoft.com/07d54d12-7133-4e15-becb-6f451ea3bea6).

7

3

2